



**FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE**

Dynamic Texture Modeling

by

Radek Richtr

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Dissertation degree study programme: Informatics
Department of Theoretical Computer Science

Prague, May 2018

Supervisor:

Prof. Ing. Michal Haindl, DrSc. FIAPR
Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Department of Pattern Recognition
Institute of Information Theory and Automation
Czech Academy of Sciences
Pod Vodárenskou Věží 4
182 08 Prague 8
Czech Republic

Copyright © 2018 Radek Richtr

Abstract and contributions

This dissertation thesis deals with a novel dynamic texture model. The proposed toroidal model can represent a vast amount of possible dynamic textures realisation. The proposed model is based on triple toroid-shaped tiles and capable of texture analysis, synthesis, enlargement, and editing. The crucial toroidal property allows synthesizing infinitely large output simultaneously in a spatial and temporal domain. Furthermore, the analysis and synthesis parts are entirely separated.

This dissertation thesis presents several techniques to enhance the visual quality of results and their potential amount. Particular attention is therefore paid to a dynamic texture inpainting problem, especially inpainting with a dynamic background which is generally recognized as a challenging problem.

Afterwards, the dynamic texture similarity the problem is addressed and a novel Fourier transformation based criterion is presented. Subsequently, with proposed criterion utilising and psycho-physical test, the results of stated inpainting method and synthesizing result are validated.

Finally, a model capability, particularly its editing abilities, to extend the existing textural BTF model to the dynamic domain is discussed and furthermore illustrated. Consequently, the novel DBTF model for representing a dynamics object is introduced.

Keywords:

Dynamic texture, temporal texture, video texture, dynamic texture synthesis, dynamic texture editing, inpainting, error concealment, textural similarity, textural models, toroidal tile.

In particular, the main contributions of the dissertation thesis are as follows:

A) DYNAMIC TEXTURE SYNTHESIS MODEL

Creating a sufficiently general and descriptive model that offers a more in-depth insight into dynamic in the dynamic texture. The model is capable of synthesizing infinitely large output simultaneously in *spatial dimensions* and *temporal dimension* (see Chapter 3).

B) DYNAMIC TEXTURE EDITING APPROACH

The novel toroidal model capable of the dynamic textures editing is presented. The method allows to suppress visual disturbances, edit texture property like dynamics and color tone or *create the mix-of-DTs*. Due to *temporal editing method*, the *interactive dynamic texture* that consists of states with varying dynamics can be created (see Chapter 4).

C) INPAINTING AND ERROR CONCEALMENT

The proposed dynamic texture inpainting approach yields to handle challenging object removal case where the *object to remove is placed on the dynamic background*. The proposed method focuses on the perceptual quality in term of human perception and produces visually high-quality results (see Chapter 5).

D) DYNAMIC TEXTURE PERCEPTUAL SIMILARITY

The novel multidimensional spatiotemporal frequency criterion regarding human perception and a different way of perceiving spatial and temporal dimension is developed and validated through series of psycho-physical tests (see Chapter 6).

E) DYNAMIC BTF

The Bidirectional Texture Function is currently the best visual texture representation of various textured materials which can be simultaneously modeled and acquired. The possible *combination of BTF and DT model* is addressed in this dissertation thesis. The stated combination allows developing novel medium for dynamic material under varying observing and illumination conditions (see Chapter 7).

Acknowledgements

First of all, I would like to express here my great gratitude to my dissertation thesis supervisor, Prof. Michal Haindl. He has been a constant source of encouragement and insight during my research and helped me with numerous problems, professional advancements and especially proofreading.

Special thanks go to the staff of the Department of Pattern Recognition of the Institute of Information Theory and Automation, who maintained a pleasant and flexible environment for my research. I would like to express special thanks to the department management for providing most of the funding for my research. Further thanks belong to the staff around the SAGElab for providing psycho-physical tests background.

My research has also been partially supported by the Czech Science Foundation as project No. 14-10911S, by the Czech Science Foundation as project No. 102/08/0593, and SGS SGS15/117/OHK3/1T/18 and SGS14/102/OHK3/1T/18.

Finally, my greatest thanks go to my family members, for their infinite patience and care.

Dedication

Rosebud. Always.

Contents

List of Acronyms	xv
Notations	xvii
1 Introduction	1
1.1 What is a Texture?	3
1.2 What is a Dynamic Texture?	7
1.3 Complex Textures Models	9
1.3.1 Bidirectional Surface Scattering Reflectance Distribution Function .	10
1.3.2 Bidirectional Texture Function	10
1.3.3 Texture Models with Fixed View and Illumination Directions	10
1.4 Motivation	11
1.5 Problem Statement	12
1.6 Goals of the Dissertation Thesis	13
1.7 Structure of the Dissertation Thesis	14
2 Background and State-of-the-Art	15
2.1 Theoretical Background	15
2.1.1 Dynamic Texture	16
2.1.2 Tile and Patches	18
2.1.3 Toroidal Mapping	20
2.2 Previous Results and Related Work	21
3 Toroid-Shaped Patch Based Dynamic Texture Model	27
3.1 Method Principles Overview	28
3.2 Toroidal-Shaped Tile Approach	30
3.2.1 Double Toroid-Shaped Tile	30
3.2.2 Optimal Cut Search	35

3.3	Dynamic Toroid-Shaped Tile	38
3.3.1	Animated Double Toroid-Shaped Tile	38
3.3.2	Dynamic Double Toroid-Shaped Animated Tile	40
3.4	Triple Toroid-Shaped Tile	46
3.4.1	Temporal Cut	47
3.4.2	Fitting plane	51
3.5	Synthesis	54
4	Dynamic Texture Editing	55
4.1	Dynamic Texture Editing Overview	56
4.2	Method Overview	57
4.3	Mix of Dynamic Textures	61
4.3.1	Transition patches	61
4.3.2	Multiple shaped tiles	66
4.3.3	Temporal Dimension Editing	68
5	Dynamic Texture Inpainting	71
5.1	Problem definition	72
5.2	Toroidal-Patch Model Adaptation	73
5.3	Dynamic Texture Inpainting	77
5.4	Automated DT inpainting	83
6	Dynamic Texture Similarity Criterion	85
6.1	Frequency characteristics	85
6.2	General Similarity Model	86
6.3	Criterion model	89
6.4	Psycho-Physical Test	94
7	Dynamic BTF	97
7.1	Bidirectional Texture Function	98
7.2	Dynamic Bidirectional Texture Function	98
8	Dynamic Texture Database and Criterion Validation	103
8.1	Dynamic Texture Database Subset	103
8.2	Spatio-Temporal Fourier Transformation Criterion Validation	106
8.2.1	Subclasses criterion validation	107
9	Main Results	111
9.1	Dynamic Texture Model	111
9.2	Dynamic Texture Editing	117
9.3	Dynamic Texture Inpainting	121
9.3.1	Time complexity	123
9.4	Psycho-physical Tests	132

9.5	Dynamic Texture Similarity Criterion Results	136
9.6	DBTF	140
10	Conclusions	143
10.1	Summary	143
10.2	Contributions of the Dissertation Thesis	144
10.3	Future Work	146
	Bibliography	147
	Reviewed Publications of the Author Relevant to the Thesis	161
	Remaining Publications of the Author Relevant to the Thesis	163
A	Inpaintig Method Comparison	165
B	Psycho-physical tests values	167
C	Similarity criterion values	169

List of Figures

1.1	Real and apparent textures	4
1.2	IRISES dynamic texture enlargement example	6
1.3	Dynamic texture in volumetric view	7
1.4	Various water-type dynamic texture examples	9
2.1	Rectangular tile	20
3.1	Synthesized dynamic texture teaser	27
3.2	Comparison of standard DTs synthesis approaches	29
3.3	Double toroid-shaped tile	30
3.4	Tile overlap	30
3.5	Spatial cuts	33
3.6	An optimal cut	35
3.7	Animated tile	38
3.8	Restrictive search	41
3.9	Toroidal boundary location for the restrictive spatiotemporal cut searching	42
3.10	Multiple rectangular tiles	43
3.11	Fourier transformation block illustration	44
3.13	Temporal cuts	47
3.12	Similarity matrix	47
3.14	Set-of-cuts illustration	48
3.15	Temporal cut plane	51
3.16	More temporal jumps	52
3.17	More temporal jumps, two possible arrangements	53
3.18	DT synthesis overall flowchart	54
4.1	Edited dynamic texture teaser	55
4.2	Data patches A and B with the same tile shape.	57
4.3	Simple mix-of-DTs arrangement	58
4.4	Two DT types combining	59

4.5	Direct mix-of-DTs with color toning	60
4.6	The transition patches illustration.	61
4.7	Transition texture patches illustration	62
4.8	Transition patches principle scheme	64
4.9	Multiple shaped tiles	66
4.10	Multiple tiles shore arrangement	67
4.11	THUJA in various rain condition with probabilities of change of states.	68
4.12	Arrangement with temporal shift	70
5.1	Several inpainted dynamic textures	71
5.2	Inpainting principle illustration	74
5.3	Inpainting by synthesis	75
5.4	The overall flowchart of the presented DT inpainting method. The straight blue arrows indicate the procedure of the present method.	78
5.5	Randomly placed patched with globally optimal shape	79
5.6	Typical inpainting problems	81
6.1	The similarity criterion computation flowchart	89
6.2	Harmonics example	89
6.3	Raw frequencies example	90
6.4	Lost frequencies example	91
6.5	False frequencies example	92
6.6	Informed psycho-physical tests arrangement	95
7.1	DBTF	97
7.2	DBTF scheme	101
8.1	<i>Class</i> CALM WATER	104
8.2	<i>Class</i> FLOWER	104
8.3	Representative selected frames from several GRASS	105
8.5	Visualization of similarity ratios between four grass DTs	109
9.1	Synthesised FIREA examples	111
9.2	Original of DTs FIREA, FIREB and FIREC	112
9.3	Synthetized texture of FIREB	112
9.4	Synthetized texture of an artificial dynamic FIREC texture	112
9.5	Synthetized texture of dynamic WATERA texture	113
9.6	Synthetized texture of dynamic WATERB texture	114
9.7	Synthetized texture of dynamic WATERC texture	114
9.8	Different number of CREEK patches used	116
9.9	Mix-of-DTs FENCE and SHRUBS	117
9.10	Mix-of-DT GRASS	118
9.11	Mix-of-DTs, TREES IN THE RAIN (multiple tiles)	119
9.12	Interactive texture THUJA in the rain	120

LIST OF FIGURES

9.13	Temporaly shifted texture	120
9.14	Detail of a WALK synthesis, method comparison	125
9.15	the comparison between our and the Newson method, part A	127
9.16	The comparison between our and the Newson method, part B	128
9.17	Informed psycho-physical tests arrangement	133
9.18	Overall comprehensive results of psycho-physical tests	134
9.19	Overall results of psycho-physical tests	135
9.20	Representative DTs subset for psycho-physical tests	138
9.21	Several frames from grass DBTF	140
9.22	Several frames from grass DBTF, detail	141
9.23	Several frames from composed DBTF	142

List of Tables

1.1	Basic texture model overview	7
8.1	DynTex database classes	103
9.1	Demonstration of different enlargements combination	115
9.2	Synthesized water texture examples	116
9.3	Measured textures (left column) and inpainted results, part A	121
9.4	Measured textures (left column) and inpainted results, part B	122
9.5	TIME COMPLEXITY COMPARISON.	124
9.6	Changes in FTs between original and inpainted DTs frames, part A	130
9.7	Changes in FTs between original and inpainted DTs frames, part B	131
9.8	The average difference of spatiotemporal frequencies criterion	136
9.9	A spatial Fourier transformation criterion values difference comparison	137
9.10	Overall comparison of frequency criterion values as the difference between proposed method and alternative methods	137
A.1	Comparisons with some state-of-the-art video and DT inpainting algorithms	166
B.1	Results of psycho-physical test	168
C.1	Presented dynamic texture similarity criterion values	170
C.2	STSIM-I and STSIM-II criteria values	171
C.3	STSIM-I and STSIM-II comparison to psychophysical test	172

List of Acronyms

Miscellaneous Computer Graphics Abbreviations

G^0	G continuity of 0th order
G^1	G continuity of 1st order
$G^{\{x,y\}}$	G continuity of x th or y th order
C^0	C continuity of 0th order
C^1	C continuity of 1st order
$C^{\{x,y\}}$	C continuity of x th or y th order

Miscellaneous Methods Abbreviations

\mathcal{Y}	Multispectral dynamic texture
\mathcal{T}	Tile, 2D (spatial) or 3D (spatiotemporal) toroidal boundary shape
\mathcal{P}	Patch, 3D (spatial) or 4D \mathcal{Y} subspace
\mathcal{H}	Hole, 4D \mathcal{Y} subspace
\mathcal{S}	Arrangement (usually created from \mathcal{P} -lattice)

Miscellaneous Abbreviations

BSSRDF	Bidirectional Subsurface Scattering Reflectance Distribution Function
BRDF	Bidirectional reflectance distribution function
DBTF	Dynamic Bidirectional Texture function
BTF	Bidirectional Texture function
MRF	Markov random field
SVD	Singular value decomposition
FT	Fourier Transformation
OF	Optical Flow
DT	Dynamic Texture
2D	two-dimensional
3D	three-dimensional
4D	four-dimensional

Notations

2D Tile

\mathcal{Y}	multispectral dynamic texture space
N	number of rows
M	number of columns
Q	common parametric curve
$Q^{v+}(u)$	parametric curve, vertical dimension of u-th pixel
$Q^{h+}(u)$	parametric curve, horizontal dimension of u-th pixel
$Q^{v-}(u)$	vertically shifted parametric curve, vertical dimension of u-th pixel
$Q^{h-}(u)$	vertically shifted parametric curve, horizontal dimension of u-th pixel
$Q(u, v)$	convex surface
$Q^v(u, v)$	convex surface, vertical line
$Q^h(u, v)$	convex surface, horizontal line
h	horizontal size of horizontal overlap
v	vertical size of vertical overlap
I_{ld}	rectangular tile (left bottom subspace), $M \times N$
I_T	whole rectangular tile $M + h - 1 \times N + v - 1$
τ	torus surface with appropriate poloidal s and toroidal t coordinates and mapping functions $x(s, t)$, $y(s, t)$ and $z(s, t)$, $s, t \in [0, 2\pi)$

Various operators and functions

$\lceil n \rceil$	ceil function
$\lceil \zeta \rceil$	size of an ζ (number of pixels)
$\lceil \zeta \rceil_v$	vertical size of ζ area
$\lceil \zeta \rceil_h$	horizontal size of ζ area
\bullet	all possible values
$\mathcal{Y}_{[\bullet, \bullet, \bullet, 42, \bullet]}$	3D multispectral 42-th frame from DT \mathcal{Y}

Overlap error

\mathcal{Y}_r	pixel r from space \mathcal{Y}
I_h	horizontal rectangular overlap $N - h \times v$
I_v	vertical rectangular overlap $h \times M - v$
I_H	horizontal rectangular overlap $N \times v$
I_V	vertical rectangular overlap $h \times M$
I_d	corner diagonal overlap $h \times v$
$\check{\epsilon}$	total oblong overlap error (horizontal, and vertical)
$\check{\epsilon}^h$	horizontal square overlap error array
$\check{\epsilon}_r^h$	particular error of pixel r from $\check{\epsilon}^h$
$\check{\epsilon}^v$	vertical square overlap error array
$\check{\epsilon}$	total corner overlap error
$\check{\epsilon}^h$	horizontal error of corner area
$\check{\epsilon}^v$	vertical error of corner area
$\check{\epsilon}^d$	diagonal error of corner area
$\check{\epsilon}^x$	complex corner areas error array
ζ	$I_v \cup I_h \cup I_d$ overlap region shape for cut finding
ζ^*	optimal overlap region with minimal overlap error ϵ^ζ
ζ_r^*	selected arranged optimal overlap region with minimal overlap error
$\check{\epsilon}^{\circ H}$	total error for complete horizontal overlap
$\check{\epsilon}^{\circ V}$	total error for complete vertical overlap

Optimal cut

ϵ_r^H	horizontal error lattice for cut search
ϵ_r^V	vertical error lattice for cut search
C	cut, series of indices on appropriate lattice
C_h	horizontal cut with length n_V
C_v	vertical cut with length n_H
Δ	neighbourhood function
Δ_r	neighbourhood of r
Δ^8	8-neighbourhood
Δ^4	4-neighbourhood
Δ	monotonic subspace of Δ in main dimension meaning
$\overset{m}{\Delta}^p$	p -th order hierarchical neighbourhood
$\tilde{\epsilon}$	cut line error
$\tilde{\epsilon}(C)$	cut line C error
$\tilde{\epsilon}^H(C_h)$	weighted horizontal cut line C_h error
$\tilde{\epsilon}^V(C_v)$	weighted vertical cut line C_v error
$C_v^{\{s\}}$	vertical cut with start/endpoint s
s^*	optimal start/end point for cut line
$C_v^{\{s^*\}}$	optimal vertical cut with start/endpoint s
C_v^*	optimal vertical cut with the same start/endpoint
ϵ	error arised by dynamic programming rules
ϵ^{h+}	horizontal composed error
ϵ^{v+}	vertical composed error
ϵ_r^{h+}	horizontal composed error for pixel r
ϵ_r^{v+}	vertical composed error for pixel r

Spatiotemporal tile and patch

Φ	an optical flow
$\Phi_{\mathcal{Y}}$	average optical flow from an whole \mathcal{Y} DT
r	multi-index with four of five components i.e. $r = [r_1, r_2, r_3, r_4, r_5]$ r_1 : row, r_2 : column, r_3 : spectral, r_4 : frame r_5 : 3-dimensional optical flow vector
$\epsilon(p, q)$	error between pixels p and q (ultiindices) with appropriate error function X and/or Z
I_{rt}	3D rectangular lattice $(0, \dots, N) \times (0, \dots, M) \times (0, \dots, T)$
$\hat{\hat{C}}$	2D cut plane
\hat{C}^{h*}	horizontal-temporal cut with dimension of $(0, \dots, T)$
\hat{C}^{v*}	vertical-temporal cut with dimension of $(0, \dots, T)$
$\hat{\hat{C}}^{h*}$	horizontal-temporal cut plane with size of $(0, \dots, N) \times (0, \dots, T)$
$\hat{\hat{C}}^{v*}$	vertical-temporal cut plane with size of $(0, \dots, M) \times (0, \dots, T)$
\mathcal{P}^A	animated patch
\mathcal{P}_i^A	i -th 2D patch from whole animated patch
\mathcal{T}^A	animated tile
$\hat{\psi}$	spatiotemporal error composed by dynamic programming rules
$\hat{\psi}_r^{v*}$	spatiotemporal error of temporal-vertical cut composed by dynamic programming rules
$\hat{\psi}_r^{h*}$	spatiotemporal error of temporal-horozintal cut composed by dynamic programming rules
δ_r^v	r vertical neighbourhood
δ_r^{tv}	r spatiotemporal vertical neighbourhood
\mathcal{E}	Animated tile error area
p	radius of \mathcal{E} in appropriate t -th frame, scalar parameter directly driven by $\Phi_{\mathcal{Y}_{[r_4=t]}}$
\mathcal{F}	Fourier (discrete, continous)
Γ	frame-to-frame similarity
I_{to}	temporal overlap index set $(0, \dots, M) \times (1, \dots, M) \times (0, \dots, T - u - t)$
Ψ^{T*}	temporal cut error (cost)
$\hat{\hat{C}}^{T+}$	temporal cut plane
$\tilde{\epsilon}()$	spatiotemporal set-of-cuts plane error
$\tilde{\epsilon}(\hat{\hat{C}}_h^*)$	spatiotemporal set-of-cuts plane error for horizontaly oriented cut plane
$\tilde{\epsilon}(\hat{\hat{C}})$	spatiotemporal cut line error for horizontal cut lines
$T\hat{\hat{C}}_h^*$	horizontaly oriented set-of-cuts structure
$T\hat{\hat{C}}_v^*$	verticaly oriented set-of-cuts structure
$\hat{\psi}^{th}$	temporal-horizontal error array
$\hat{\psi}_r^{Th*}$	optimal temporal-horizontal error value builded on vertical structure
$\hat{\psi}_r^{Tv*}$	optimal temporal-vertical error value builded on horizontal structure
$\hat{\psi}_r^{th*}$	temporal-horizontal error value for pixel r

DT editing

\mathcal{Y}_t	multispectral transition dynamic texture
\mathcal{P}_t	transition patch
\mathcal{T}_t	transition tile
δ^*	offset (i.e. \mathcal{P}_t in \mathcal{Y} or shift between \mathcal{P}_i and \mathcal{P}_j)
ϵ^*_{th}	potential error for one arrangement location and appropriate temporal-horizontal cut plane
ϵ^*_{tv}	potential error for one arrangement location and appropriate temporal-vertical cut plane
ϵ^*_{hv}	potential error for one arrangement location and appropriate vertical-horizontal cut plane
$\hat{\epsilon}_{th}$	real cut plane error for one arrangement location and appropriate temporal-horizontal cut plane
$\hat{\epsilon}_{tv}$	real cut plane error for one arrangement location and appropriate temporal-vertical cut plane
$\hat{\epsilon}_{hv}$	real cut plane error for one arrangement location and appropriate horizontal-vertical cut plane
$\delta_{\mathcal{P}}$	location (shift from $[0, 0, 0]$) of \mathcal{P} in the appropriate \mathcal{Y}
Λ	total temporal shift
λ	gradual temporal step

DT inpainting

\mathcal{E}_{Λ}	gradual time changes area
ζ^*	vertical overlap size
\mathcal{H}^{ζ}	\mathcal{H} neighbourhood area
$I_{\mathcal{H}}$	\mathcal{H} area index set with
I_{ζ}	\mathcal{H} neighbourhood area index set
I_{ζ}^-	left-bottom-front quadrant of I_{ζ}
$\bar{\phi}(\mathcal{Y})$	average optical flow vector
$\hat{\epsilon}_B$	border patch total error
ϵ^{ϕ}	optical flow error
\mathcal{S}_{inp}	\mathcal{P} -lattice inpainting arrangement
Υ^*	dissimilarity measure of an area to whole DT
Υ	or dissimilarity measure of between two areas.
v	dissimilarity measure of multispectral pixel

DT similarity criterion

CSF	contrast sensitivity function
ξ	spatial frequencies vector, harmonics
$\hat{\xi}$	unit vector ξ
$\hat{\Xi}$	set of all unit vectors $\hat{\xi}$
ξ^t	ξ projection to t axis
$\hat{\Xi}^t$	subset of all ξ^t
$\tilde{\xi}$	crucial spatial frequency of given \mathcal{Y}
$\hat{\Xi}$	set of all $\tilde{\xi}$ from given \mathcal{Y}
$\xi_{Y\delta}^{\zeta, \bullet}$	crucial spatial frequency temporal harmonics set
w_f	multiplicant constant for $\hat{\Xi}$ size
Θ	dynamic texture similarity function between two dynamic textures
θ	dynamic texture similarity function between two harmonics set
M_{x_3}	magnitude of human perception scaling function
L	measure between two harmonics set
L^σ	raw measure between two harmonics set
L^{lost}	lost frequency measure between two harmonics set
L^{false}	false frequency measure between two harmonics set
L^α	weighted measure between two harmonics set with parameters α_l for lost and α_f for false frequencies
$\dot{\Psi}$	overall inter-class average ratio
$\ddot{\Psi}$	total mean ratio
${}^{inter}_{\mathcal{Y}}\Theta_{AY_s}^L$	intra-class average similarity criterion
${}^{iter}_{\mathcal{Y}}\Theta_{AY_s}^L$	inter-class average similarity criterion

DBTF

- ς spectral band, r_3
- ω_i illumination direction, $\omega_i = [\theta_i, \phi_i]$
- ω_v viewing direction, $\omega_v = [\theta_v, \phi_v]$
- \mathcal{U} hidden multidimensional dynamic generation process
with general κ parameter
- ω general angular speed
- ϵ_ω angular speed error

Graph Theory

- G (oriented) graph
- V set of all vertices
- E set of all edges
- $w(V)$ weight function

Introduction

The use of visual textures allows the concentration of considerable amount of information within a single medium. While for designers a texture adds material feeling to design and allows the improve simple geometry, for computer scientist textures provide a considerable amount of information as a key component to image analysis, human perception understanding, scene segmentation and texture synthesis.

In many cases, the static images are not enough to obtain the sufficient amount of information to scene content analysis and recognition. Multiple images of a scene from a different viewpoint or at different time instants can add a desired semantic description. For example, the movement vector of some object in the scene, the weather and wind direction determining from water surface or forest images, detection of movement behind scene object and many other. For many processes analysation, the entire sequences of images (textures) are crucial or even necessary.

This necessity is due to the fact that the crucial information is not detectable by the mere scene geometry in a one-time interval or by the scene radiometry but in the scene and its evolution over time, or its dynamics only. Even though this dissertation thesis is devoted to analysis, modeling, synthesis and editing of images of scenes that exhibit varying statistics in the scene geometry and radiometry through time series which is closely related to analysis, modeling, synthesis and editing of still texture the problem definition and topics overview will by often presented in still texture domain first. Video sequences that have certain homogeneous visual and dynamic properties will continue to be called as dynamic textures from now on.

Unfortunately, neither static nor dynamic rigorous mathematical texture definition exists. Dynamic textures (DTs) can be vaguely defined as spatially repetitive motion patterns exhibiting homogeneous properties in the temporal well as spatial domain. Dynamic textures examples might be smoke, haze, fire or liquids, also waving trees or straws or some moving mechanical objects.

Visual texture modelling is the crucial part for any computer-based visualisation application because whatever size is the measured texture (or generally any data), it is always insufficient and requires its enlargement to cover the required visualised object's surface

area. The typical example in still texture domain is the recent most advanced visual surface material representation in the form of the bidirectional texture function (BTF). Dynamic texture synthesis editing aims to create a visual texture which is perceptually similar, ideally visually indiscernible, to the target dynamic texture and has the required spatial and temporal extent.

Texture inpainting and error concealment (although it can also be seen as texture editing), unlike classical texture synthesis, does not enlarge the dimension of the resulting texture and instead removes inconsistent or unwanted objects in the texture. The original motivation for image inpainting was to reconstruct damaged images or videos frame by frame due to the ageing of stored media or noise.

From the original task that consists of error area and artifact restoration, methods continuously generalize to the problem of editing and object removal. This task has been intensively studied in the area of image restoration, and many useful results have been reached. However the subject of the video (or dynamic texture) inpainting is far less extensively studied, probably because of the complexity of high dimensional data processing and due to obstacles caused by the need for homogeneity of the temporal dimension. Since more methods can generate the desired results, it is necessary to compare the results between each other to obtain the optimum results. Method results comparing can be achieved either by psycho-physical tests or via a suitable validated similarity or quality metric. Emphasize that the term (video) inpainting in the context of this thesis does not just mean to recover missing or unwanted data but to synthesize it with regard to the whole dynamic texture. Therefore, in the interest of the quality, borderline patch error and homogeneity of the texture as a whole the minimal residual part of the input texture could be sacrificed to a certain extent.

As the term texture is crucial for this thesis topics, the term definition and usage is first briefly discussed in the Section 1.1, followed by a discussion of a more specific term dynamic texture in Section 1.2. The general complex texture models are briefly described in Section 1.3. The dissertation thesis motivation (Sec. 1.4), problem statement (Sec. 1.5), goals (Sec. 1.6) and structure are presented in the order given.

1.1 What is a Texture?

The world around us is full of textures and materials which are visually demonstrated as visual textures. The surface of any visible object is, in some scale, textured as well as the apparent visual surface (like a dense forest, or a crowd of people, thick bushes, . . .) because also more object can be in some scale perceived as a texture. A spectrum of visual textures is extremely wide from natural surfaces of a wood plant, skin, artificial surface like metal, car paint, linoleum, or complex visual surfaces like grass, hair, dense forest or crowd of people. In both real and apparent surfaces the texture refers to surface characteristics and appearance in terms of color, arrangement, roughness, glossy, coarse or fine, reflectivity, etc.

Texture can be obtained as a measurement from the real world by photographing or measuring with a specific photometric device. Also, texture can be painted or rendered from virtual scene artificially by computer graphics tools.

Here is proper to say that there is one very specific sort of textures which has a very tight connection to human perception - sound textures. Although the sound texture synthesis is interesting topics (see, e.g. [92, 91]) and even sound texture can be present in some natural dynamics visual texture (i.e. the sound of water in waterfall scene, birds song in the dynamic texture of the forest) this dissertation thesis is strictly focused to visual dynamic texture without any sound texture track.

The basic division between tactile textures and visual textures is represented by a primary sense that the given texture is perceived. Tactile texture directly refers to a tangible human feel of real surfaces. Visual textures refer to human visual perception. The impression of visual texture is more than the impression of tactile textures influenced by the environment - illumination direction, intensity and spectral characteristics, optical conditions of environment, but also spectral properties of the surrounding surface as well as the observer's own (i.e., the properties of the eye). This dissertation thesis is focused on visual textures, so the term texture is therefore used as phrase visual texture although this will be highlighted in some specific situations.

After the insight into some dictionaries, various definitions of the term texture can be found. The terms are related to vast characteristics and research or art areas as painting, taste, music, etc. More relevant definitions are related to structural arrangement or fabrics. The various definitions of textures, for example, consist of:

English oxford living dictionary:

”The feel, appearance, or consistency of a surface or a substance.”

Merriam-Webster:

”The disposition or manner of union of the particles of a body or substance; the visual or tactile surface characteristics and appearance of something.”



Figure 1.1: *Real and apparent textures*: Natural texture examples with various characteristics. On the right real physical surface (tactile) textures, on left apparent surface textures.

Dictionary.com:

”The characteristic physical structure given to a material, an object, etc., by the size, shape, arrangement, and proportions of its parts.”

Art Vocabulary Terms:

”The tactile quality of the surface of a work of art.”

American Heritage Collegiate Dictionary:

”Something composed of closely interwoven elements; the structure formed by the threads of a fabric.”

The vast spectrum of characteristics as a representation of the original 16th-dimensional function lead to many purpose-oriented models like Bidirectional Texture Function, Spatially Varying Bidirectional Reflectance Distribution Function, Bidirectional Surface Scattering Reflectance Distribution Function, etc. Depending on the model, textures are represented by two or more dimensional images or sets of images.

Visual texture

Although textures are remarkably often used in computer graphics and science neither visual nor tactile rigorous mathematical texture definition exists, and term texture is often considered to be very vague and obscure. The reason is probably the extremely wide

spectrum of properties and close relation to particular human perception. Even worse, in many cases, the key features are clearly contradictory or almost mutually exclusive.

The range of these properties covers regularity versus stochasticity, uniformity versus distortion, sharpness versus noisiness, locality versus global properties. Moreover, these properties often fluctuate with the distinctive texture scale and view angle.

From a large number of approaches and researchers, let's name the most basic ones which differ fundamentally from each other and represent basic views and approaches. The different texture definitions usually consequently lead to strictly different computational approaches to texture synthesis and analysis. Francos, Meiri and Porat[50] defines a texture vaguely as a structure which is made of a large ensemble of elements that resemble each other *very-much*, with some kind of an *order* in their locations, so that there is no one element which attracts the viewer's eye in any unique way. Although the definition is intuitively correct, it is in many cases inadequate, even inaccurate in terms of visual perception and visual quality. The robust self-similarity criterion with no irregularity to "attract viewer's eye" leads to material-like or poor perception (see. i.e. marble texture in Figure 1.1).

Haralick[60, 59] defines texture as an "organised area phenomenon". As a typical structural approach, this leads to texture decomposition into "primitives" with some specific spatial distributions. The structural decomposition is strongly driven by the human visual experience of visual textures. The spatial structure of primitive objects (i.e. apples, leaves, or scales on in image 1.1) is seen as a realisation of a stochastic field interestingly satisfy many of different texture definition in a similar way as a primitive itself. The primitives can be realised not in a stochastic way but some higher order placement (i.e. wool, knot pattern, tile floor). With both principle (stochastic and regular) structure approach can cover a wide variety of visual textures.

In apparent contradiction to structural approaches is stochastic one, i.e. based on Markov Random Field[22]. Cross and Jain[22] consider that texture is "A stochastic, possibly periodic, two-dimensional image field." This approach is popular probably due to the possibility of stochastic field mathematical notation and therefore a more explicit link with multidimensional mathematical processes, unlike structural approach. For example, texture definition as a stochastic process that generates the texture is possible with the stochastic approach.

Without limitation to generality and a particular model, the consensus across the community says that the key textural property is *spatial and temporal homogeneity*. From the structural point of view satisfying the need for homogeneity require a repetition of texture primitives and therefore some degree of self-resemblance. Even if the exact primitives can vary in color, illuminance or shape the basics, visual appearance is similar. Typical examples can be fractal[127] or building block, tiled floor, or kilts and tartans. The statistical approach defines homogeneity as *statistical stationarity*. Therefore the self-similarity is required at the level of the statistical values of the signals. The degree of homogeneity can serve to classify the texture into several basic groups. For example, Zhou[145] classifies the texture into the homogeneous, weakly-homogeneous, and inhomogeneous classes.

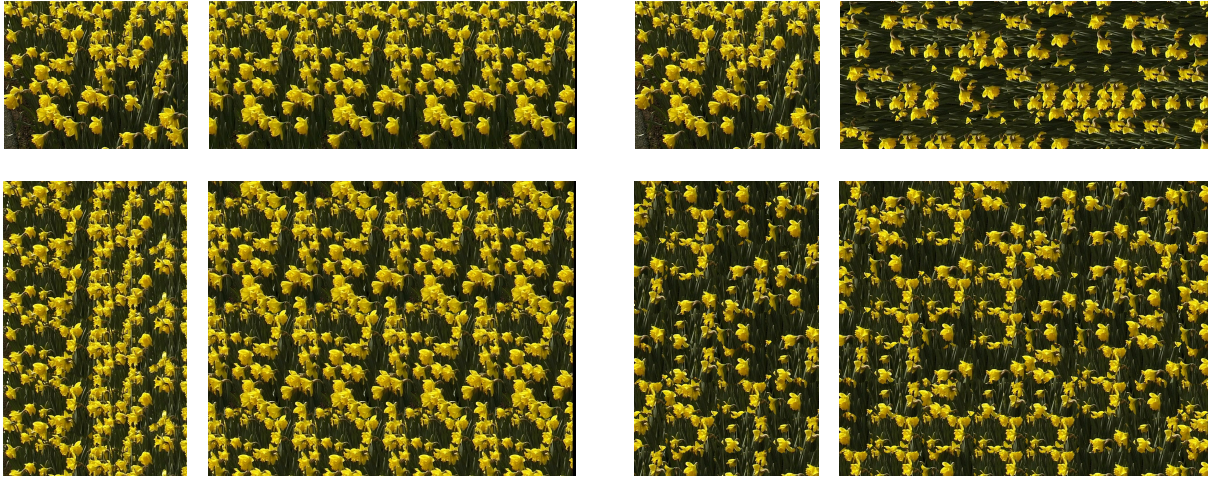


Figure 1.2: IRISES *dynamic texture enlargement example*. Left: Verbatim repetition, right: more patches. The original dynamic textures are the small pictures on the top left. Upper row and left columns illustrates one dimension enlargement while big square represent both spatial dimension enlargement.

Texture Synthesis

The using of texture is a quintessential part of almost any computer-based visualisation application. Because of the desired object which needs to be textured can has any size the appropriate amount of texture is needed. Although in some exceptional cases the measured amount of data may be sufficient, it may easily prove inadequate if the size of the object is changed. What's more, sufficiently precise measurement can be resource-intensive, time-consuming or even impossible, i.e., reconstruction and preservation of incomplete museum artifacts. The typical example of a precise but time-consuming medium in still texture domain is the recent most advanced visual surface material representation in the form of the bidirectional texture function.

Texture modelling and synthesis aims to create a visual texture which is perceptually similar, ideally visually indiscernible, to the target original texture and has the required spatial extent. The synthesized or enlarged texture should have similar, or the same perceptual characteristics as stated for example De Bonet[25] but should also avoid to verbatim repetitions[145] of some pattern like tileable textures (see Figure 1.2).

Texture enlargement and synthesis are closely related to texture model approach. The structural approach, therefore, leads directly to *intelligent sampling* - picking some representative patches of appropriate shape from given input texture sample of limited size and consequently using them to generate the desired size texture. Statistical approach based synthesis is used in some defined *mathematical models*, often with relation to a particular physical phenomenon. The generated texture is created according to given analytical function. Of course, the hybrid methods, i.e. as propose Nealen[95] are possible, usually trying to pick the advantages of both approaches.



Figure 1.3: *Dynamic texture in volumetric view*: First 100 frames visualized. The side view (from right to left) and top view (from top to bottom) are visualized of left and bottom, respectively.

1.2 What is a Dynamic Texture?

At first, it is good to say that dynamic textures are sometimes called as temporal textures in some literature, e.g. articles as by Szummer[124], or video texture, e.g. articles by Schödel[113]. Both terms share a large part of the characteristics but due to the excessive generality of both terms which may in principle cover an unnecessary amount of data the term dynamic texture[30] will be exclusively used in this dissertation thesis. Remind, that the similar areas as i.e. motion textures[99] are not part of this thesis topic, as its do not model the time-varying spectral information but the motion measurement itself. Similarly

Table 1.1: *Basic texture model overview.*

Variables	spatial	spectral	view	illumination	Temporal
Sound texture	0	0	0	0	1
Still grayscale texture	2	0	0	0	0
Still multispectral texture	2	1	0	0	0
Solid texture	3	0	0	0	0
Multispectral procedural texture	3	1	0	0	0
Dynamic texture	2	1	0	1	1
BTF	2	1	2	2	0
DBTF	2	1	2	2	1
BSSRDF	4	1	2	2	0

to other types of visual textures mentioned above, there is neither exact definition of the dynamic texture [29, 57]. The Dynamic texture can be characterised by some spatially invariant statistics like other visual textures [146], but additionally, temporal invariance through particular images is also required. Consequently, the spatial and temporal homogeneity is required. It is important to say that the spatial domain homogeneity and the temporal domain homogeneity have fundamentally different characteristics. This can readily be seen by a simple experiment: Imagine some dynamic texture, e.g. the irises on Fig. 1.2. For human perception the content of the dynamic texture is clear and distinct - yellow flowers of irises moving in the wind. If the spatial axes will be switched, the human recognized content should be preserved. Moreover, the DT's dynamic still remains intact. When any spatial axis (for example, horizontal dimension) will be changed with the temporal axis (or, geometrically said, the whole data block will rotate around Y axis by 90 degrees) the resulting dynamic texture will be for human perception not the video of irises, but a video with oscillating yellow / green lines. Moreover, the dynamics of the resulting dynamic texture will drastically change. The experiment can be well viewed on volumetric view of a dynamic texture IRISES on Figure 1.3.

In many typical cases, the static images are not enough to obtain the sufficient amount of information to analyse. The recognition capabilities can be strictly limited as an image is only one-time interval realisation. For example speed of some object, or its movement direction needs the set of images and thus set of more time interval observation for an effective estimation. Multiple images of a scene from a different viewpoint or at different time instants can add a semantic description which. For many processes analysis, the entire sequences of observations are crucial or even necessary. This necessity is due to the fact that the crucial information is not recognizable by the mere scene geometry in the one-time interval or by the scene radiometry, but also in scene dynamics.

The fields of application for dynamic texture consist of video restoration and recognition, animation, computer games, virtual reality, dynamic object modeling, video rendering and many other. The basic advantage of dynamic textures allows, due to temporal high-order characteristics, bypassing difficult modeling of some complex spatial structures and their dynamics in particular. Moreover, using dynamic textures enables to keep photorealistic material with plausible and real dynamics.

From interminable types of dynamic textures, the examples could be grass, trees in the wind, flowers, smoke, foliage, snowing, insect swarm, waving flags, human crowd, sea waves, haze, fire, and many others. The determination of a dynamic texture by classes are difficult problem itself with tight connections to human perception. An immense space of possible dynamic texture realisation with a different color, illumination of structure can be seen on Figure 1.3 as a small subset of WATER type DTs examples.



Figure 1.4: *Various WATER-type dynamic texture examples, selected frames.*

1.3 Complex Textures Models

A real material surface reflectance is a complex physical phenomenon which depends on many physical conditions. The general reflectance function[57] can be expressed in form of 16-dimensions function:

$$GRF(\varsigma_i, x_i, y_i, z_i, t_i, \theta_i, \phi_i, \varsigma_v, x_v, y_v, z_v, t_v, \theta_v, \phi_v, \theta_t, \phi_t) , \quad (1.1)$$

where ς_i is incident light spectral value. The triplet x_i, y_i and z_i denote the illuminating surface location in time t_i . $\omega_i = [\theta, \phi_i]$ is illumination spherical incidence angles observed at time t_v . Surface location is denoted by x_v, y_v and z_v with spherical reflectance angles $\omega_v = [\theta_v, \phi_v]$ and appropriate spectrum ς_v . $\omega_t = [\theta_t, \phi_t]$ are the corresponding transmittance angles. Note that $\omega = [\theta, \phi]$ are the elevation and azimuthal angles, respectively.

Clearly, GMF is an extremely complex function to measure or even model. Thus more suitable model for visual texture representing is needed. The wide spectrum of simplifying assumptions from general GMF leads to particular specific model as the existence of a subsurface scattering, texture isotropy condition, independence on simultaneous rotation of illumination and viewing azimuthal angles around the surface normal and many others. From complex GRF taxonomy, the following model has the closest relation to this dissertation thesis topics.

1.3.1 Bidirectional Surface Scattering Reflectance Distribution Function

The Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF) is a 9-dimensional model proposed by Nicodemus[99] as a visual texture reflectance representation:

$$BSSRDF(\varsigma, x_i, y_i, \theta_i, \phi_i, x_v, y_v, \theta_v, \phi_v). \quad (1.2)$$

where parameters are a GRF (1.1) model generalisation based on specular (reflected light) and diffuse components that are determined by the light scattering in surface microstructure. Bidirectional Surface Scattering Reflectance Distribution Function is currently the most general yet, with some simplifying assumptions, suitable GRF submodel. Note that no satisfactory BSSRDF dataset was measured yet and only approximate visualization methods exist, but still, BSSRDF is probably the best GRF approximation which is possible that will be measured in the near future.

1.3.2 Bidirectional Texture Function

Bidirectional Texture Function (BTF) is 7-dimensional reflectance model and current visual texture state-of-the-art. The model is defined as a BSSRDF subspace:

$$BTF(\varsigma, x, y, \theta_i, \phi_i, \theta_v, \phi_v) \quad (1.3)$$

and was (in monospectral variant) first proposed by Dana[23]. The BTF is currently the best visual texture representation of various textured materials which can be simultaneously modeled and acquired even for nonplanar and opaque surfaces. The process of BTF measuring is time consumable and complex, but the results are currently the most high-end and physically correct surface materials appearance modeling. Above all, the adaptability of the model to the illumination conditions is extremely wide.

BTF as current state-of-the-art is used in top computer graphic applications, photorealistic material modeling, filming, virtual reality modeling, museums preservation, visual scene analysis, recognition of complex real-world, biometric and medical applications and many others.

1.3.3 Texture Models with Fixed View and Illumination Directions

Multispectral visual texture The well known and widely used model for GRF is a simple multispectral visual texture:

$$VT(\varsigma, x, y) \quad (1.4)$$

where both viewing and illumination direction if fixed. Emphasize that contrary to frequently misunderstanding the function is 3-dimensional with two spatial and one spectral dimension. The VT is usual and mostly utilized form of a texture, or generally, any image representation.

Monospectral Texture Recall, that monospectral texture obviously miss the ς parameter:

$$GT(x, y) . \quad (1.5)$$

Procedural Texture The model similar to (1.4) is already mentioned material procedural (grayscale) texture in monospectral:

$$PGT(x, y, z) , \quad (1.6)$$

or multispectral color variant:

$$PT(\varsigma, x, y, z) . \quad (1.7)$$

Dynamic texture Finally, the dynamic texture model is defined as a multispectral texture (1.4) with dynamics and so in the most general and pure form as

$$DT(\varsigma, x, y, t) . \quad (1.8)$$

1.4 Motivation

Texture synthesis In the computer graphic and more specifically in pattern recognition, the study of dynamics textures dates three decades back. The pioneering works perform dynamic texture classification based on temporal features extracted by segments of dynamic textures. By many years the definitions of this medium type vary from temporal texture, video texture, spatiotemporal textures, dynamic texture and many other. The medium was more or less precisely described in many different ways [A.5] based on statistical or structural properties. While the structural approaches typically exhibit high-quality results, the statistical ones present general methods but typically more time complex ones.

The methods typically try to generalise the pure spatially based approaches but lack the temporal dimension effective handling. Thus effective method as [A.2] that can produce high-quality results concerning main medium property - dynamics - is needed.

Texture editing Visual texture modelling is the critical part for any computer-based visualization application because whatever size is the measured texture, it is always inadequate and requires its enlargement to cover the required visualized objects surface area. The available material sample size is either too small for rendering complex and large virtual scenes if we measure material visual properties of real existing objects, or the measurement technique does not allow us to measure larger material samples. The typical example is the recent most advanced visual surface material representation in the form of the bidirectional texture function [56]. The amount of such measured data similarly to dynamic textures is immense, e.g., in the range of terabytes even for such spatiotemporally restricted measurements thus any such texture visualization inevitably requires enlargement and some compression capability simultaneously.

Thus texture editing method[A.3] capable of creating photorealistic synthetic textures, which are either difficult to measure or which even do not exist in nature and is general enough to create a broad spectrum of dynamics textures is desired.

Textural Similarity Criterion The ranking of dynamic texture similarities is a difficult problem due to real material textures complex dependencies on 16 physical observation parameters with a combination of temporal dimension behaviour. This problem is not satisfactorily solved even for simpler static textures Currently the only reliable, but extremely impractical and an expensive option is to exploit the methods of visual psycho-physics [56].

Clearly, criterion[A.1] that correlates well with psycho-physical tests and are capable of ranking the synthesis or inpainting results is missing.

Dynamic Texture Inpainting Inpainting (mostly image inpainting) is intensively researched part of image processing. The original motivation for image inpainting was to reconstruct damaged images or videos frame by frame due to the ageing of stored media mechanical damages, or noise [70]. Later many methods were introduced to use an image and video inpainting not only for reconstruction but also for editing of image or videos in the object removal area.

Even if there are many effective approaches (mostly working with static background), current methods have still many drawbacks. The most of dynamic textures inpainting approaches suffer from false optical flow or produces results with visually easy recognizable artefacts. Even if the synthesized dynamic texture is valid, the shape of a mask is easy observable like a (moving) silhouette in the dynamic background. Another usual problem is time complexity. Time demanding operations are typical, i.e. for convolution nets approaches in which the synthesizing of an even small texture can take weeks to proceed. So some method[A.4] which can overcome these drawback is needed.

1.5 Problem Statement

This dissertation thesis aims to develop a novel dynamic texture model capable of synthesizing dynamic textures and improving the current methods. The proposed model is applicable in dynamic texture enlargement and capable of synthesising a significant amount of dynamic textures type from water, natural scenes, human-made objects, and many other.

The model generality leads to an ability to edit dynamic textures and even recreate photorealistic synthetic textures, which are either difficult to measure or which even do not exist in nature.

As the model is capable of effectively edit the dynamic textures, the intensively researched image processing topics called inpainting is handled with it.

As the only way to compare the quality of a synthesis and inpainting results are tedious psycho-physical experiments, the mathematical criterion of similarity is final key topics of this dissertation thesis.

1.6 Goals of the Dissertation Thesis

The main goals of this dissertation thesis sustain of:

- A) **DYNAMIC TEXTURE SYNTHESIS MODEL** To create a dynamic textural model capable of synthesizing high-quality results. The result should be visually represented with regard to the type of synthesized texture. The desired method should be capable of both spatial and temporal enlargement and synthesis.
- B) **DYNAMIC TEXTURE EDITING APPROACH** To develop a method to edit a dynamic texture with respect to a created dynamic textural model.
- C) **INPAINTING AND ERROR CONCEALMENT** Development a methods for restoration of error areas or unwanted parts in the dynamic textures.
- D) **DYNAMIC TEXTURE PERCEPTUAL SIMILARITY** To create a method capable of evaluating the results of the synthesized and inpainting process concerning the visual quality and human texture perception.
- E) **DYNAMIC BTF** To discuss and examine the potential of combining the BTF model and dynamic texture model.

1.7 Structure of the Dissertation Thesis

The thesis is organized into ten chapters as follows:

1. *Introduction*: Describes the motivation behind our efforts together with our goals. There is also a list of contributions of this dissertation thesis.
2. *Background and State-of-the-Art*: Introduces the reader to the necessary theoretical background and surveys the current state-of-the-art.
3. *Toroid-Shaped Patch Model*: Provides a description of our approach to the problem of dynamic texture modeling.
4. *Dynamic Texture Editing*: Demonstrate the ability of a proposed dynamic textural model to the dynamic texture editing problem and provides a description of model enhancements depending on DT editing issues.
5. *Dynamic Texture Inpainting*: Present a dynamic texture inpainting problem description and demonstrate the ability of a proposed dynamic textural model to deal with it. Next describes model enhancements depending on DT inpainting issues.
6. *Dynamic Texture Similarity Criterion*: Presents a novel Fourier transformation based criterion usable for dynamic textures comparison and ranking of synthesis and inpainting results.
7. *Dynamic BTF*: Demonstrates the possibility of creating novel DBTF media based on Dynamic texture and BTF model combination.
8. *Dynamic Texture Database and Criterion Validation*: Covers the data which were used for experiments to verify our approach.
9. *Main Results*: Contains examples of the results of the presented methods.
10. *Conclusions*: Summarizes the results of our research, suggests possible topics for further research, and concludes the thesis.

Background and State-of-the-Art

At the beginning of this chapter, in Section 2.1, the necessary theoretical minimum required to enter the topic is introduced. As the primary goals of this thesis is a novel approach to synthesize dynamic textures with using of a triple toroidal-patch, the dynamic texture as a term in relation to other complex textural media is explained first in Section 2.1.1 following with Section 2.1.2 where principles of a toroidal patch are recapitulated.

In the second State-of-the-art section (Sec. 2.2) goals coinciding with dynamic textures and field overlap between them are discussed first, followed with dynamic texture overview.

Whereas the using of synthesis approach to editing and inpainting is a crucial part of this dissertation thesis, the related areas of research covering still and dynamic textures synthesizing and inpainting are discussed next.

Hence any synthesizing method results must be compared in some ways, last part of state-of-the-art section summarises approaches for texture fidelity benchmarking, similarity computing and psycho-physical test performance to compare the quality of synthesis and/or inpainting results.

2.1 Theoretical Background

Let's mention here what $4D$ means in the context of this dissertation thesis, dynamic textures and common still texture models. The most general and accurate description of any still visual texture can be obtained by $16D$ general reflectance function describing the real material textures complex dependencies on 16 physical observation parameters as postulates Haindl in [56].

The more general $9D$ bidirectional surface scattering reflectance distribution function (BSSRDF) is defined by Nicodemus in [99] as light transport within the interaction of the light beam with material surface.

Bidirectional texture function as introduced by Dana[23, 24] is BSSRDF generalisation with no surface scattering and thus depends on illumination and viewing direction and texture spatial coordinates resulting in $7D$ function.

Presumptions on material properties influencing light scattering have resulted in more disposable approximative $5D$ material representation called bidirectional reflectance distribution function. The bidirectional reflectance distribution function derivation is based on BTF representation of spatial dependency on the points of light incidence and reflection.

The most straightforward way to describe a $2D$ texture with spatial dimensions, say, u, v (width and height), where values at given coordinates can be defined as magnitude on black-white spectre. Color still $3D$ textures are natural extensions with two spatial and one spectral dimension, where the last dimension can be one channel from RGB, HSV, or in any other color model.

The solid color $4D$ textures are very specific procedural textures of material (e.g. wood, marble, ...) with three spatial dimensions (width, height, depth) and one spectral dimension. On the contrary, the dynamic color textures (usually called as dynamic textures) are a direct extension of still color textures with two spatial, one spectral and one temporal dimension (see Table 1.1).

2.1.1 Dynamic Texture

The most general term relevant to dynamic textures are video textures as postulate Schödl in [113] or as a general temporal autoregressive model by Szummer[124]. While video textures represent a general infinitely varying stream of any pictures with no exact limitations and represent general medium, dynamic textures as defined by Doretto[27] [30] are a subset of video textures and as a linear dynamical system (LDS) are restricted by many limits.

Although neither static nor dynamic (or temporal) rigorous mathematical texture definition exists[56], dynamic textures can be vaguely defined as spatially repetitive motion patterns exhibiting homogeneous temporal properties[147] (i.e., spatial and clearly discernible temporal homogeneity between frames). Examples might be grass, haze, fire or liquids, also waving trees or straws or some moving mechanical objects. So the dynamic texture can be defined as a realisation of $4D$ stochastic random field. Like $2D$ and $3D$ textures, many of them can be modeled only as a mix of several dynamic textures[15, 16] or by layered dynamic textures[17].

In general it can be said that dynamic textures must have certain spatially invariant statistics[147] (be homogenous) and clearly discernible temporal homogeneity between frames. Similarly, Doretto[27] postulate dynamic texture as generative video model defined by a random processed with some observed variable and hidden variables.

Three significant properties[27, 30] are present in dynamic textures - the static texture itself, global dynamics and local dynamics. They intuitively define whole texture in the meaning of human perception and texture classification.

The static texture represents a structure of the scene, and its objects, its visual appearance, illumination, color in one DT frame. In the other words, the structure is analogous to still texture and represents whole frames of DTs. The preservation of these properties is crucial[46, 45, 47] to preserve the visual quality appearance of dynamic textures in the meaning of geometric (G^0) continuity.

The second significant property represents moving of whole scene (handshake, camera motion) and typically must be fixed by appropriate techniques[77].

The third property - local dynamics - is usually caused by the periodic motion of a small object in the texture (i.e. oscillation or overlapping and disappearing of small objects) and create the local optical flow (OF). Not preserving of the optical flow causes a violation of $G^{\{1,2,3\}}$ continuity curves - inconsistencies in dynamics due to the entire image, create a false optical flow, or abrupt changes in OF velocity.

Optical flow Optical flow determining is one of fundamental problems[65] in the processing of image sequences. Its goal is to compute an approximation of the 2D or 3D motion field of velocities and directions and thus add spatiotemporal information to spatial image structure. The great potency to use such information reflect to wide research determined to this extensive field - many methods for optical flow with various approaches has been proposed.

As the optical flow computation becomes more standard and widely used process, for example in many computer vision libraries, variation of approaches grows. Today the optical flow approaches can challenges problems associated with complex natural scenes, including nonrigid motion, real sensor noise, motion discontinuities, automated tracking systems and many more.

Clearly, a fully comprehensive survey is beyond the scope of this paper. For this dissertation thesis the dense optical flow[74], widely used traditional pyramidal Lucas-Kanade[88, 13] can be used with advantage of simplicity in case of larger structures movement detection. For handling both rough and detailed local dynamics the optical flow prove itself to be satisfactory[6, 100, 132, 4].

Synthesis Approaches The video synthesis (and thus the dynamic texture synthesis, too) is extensively studied in past three decades. Many contributors try to established sufficient and exact statistical generative models of video sequences by many different approaches which can be divided by many characteristics.

The two main approaches can be applied to the dynamic texture modelling - the deterministic structural models approach based on the measured data sampling (intelligent modeling) and the mathematical statistical model based approach. The mathematical models can be further divided to two basic categories, while the first is procedural model devoted to some particular physical phenomena. The second mathematical model category are adaptive models like i.e. LDS or autoregressive models.

Note that the 4D procedural models for color 3D-spatial material textures (i.e. models of marble, wood) are oftenly taken into this class even if their labelling as "dynamic" is questionable and comes from the number of dimensions. On the other hand models for behavior of water surface, hair, flames, or trees and plant is comes from this category.

The other distinction is the basis of synthesizing approach, which can be whole frames, spatial or spatio-temporal data patches, pixels, on other statistical basis functions (like Gabor filters or steerable filters).

Finally, the last major distinction is between a physically-based model that simulates some (usually very specific) DT types like water or gaseous phenomena i.e. by NavierStokes equations or fluid motion hydrodynamics and ad hoc experimental model with no more less physically-based principles i.e. on Potts field.

Mention that, of course, the division by dimension of output is possible as some methods can synthesise results only in some dimension (temporal enlargement) or are strictly limited in spatial dimension (synthesize only a multiple of original DT size) while others can create infinitely large results. Of course, an even infinitely large output may not be sufficient as the result of some method continually degrades.

Dynamic texture classes An extremely wide range of potential realizations of a particular dynamic texture requires their clustering to some particular classes. Although it is possible to use specific mathematical criteria for some sorting of textures (such as average color, homogeneity, entropy, ...), it is more practical[] to use a rather semantic category. Note that the dynamic texture classes are addressed as FOLIAGE (marked with small uppercase caps) and the concrete dynamic texture realisation (measured or synthesized) as WALK (small caps). Due to the inaccurate definition of the dynamic texture itself, their classification into these thematic classes is of course very fuzzy.

2.1.2 Tile and Patches

Deterministic structural models approach based on the measured data sampling varies in the form of used measured data. The used patch form greatly determinate the synthesise (inpainting, editing, ...) methods spectrum of outputs, their quality and possible method limitations.

Mention here, that the word "tile" (noted as \mathcal{T}) is used as the shape of the data used by given method regarding spatial and temporal dimensions. The term "patch" (or \mathcal{P}) then refers to the specific tile content given by the input data. Usually, one tile can generate more patches. The ambiguous term "patch tile" then covers both of the above meanings, including their properties simultaneously.

If more than only one input dynamic texture is used, the terms tile and patches are used in a similar way. The tile term is usually shared by more inputs and addresses its shared properties. The patch term is then bound to concrete data from concrete dynamic texture realization.

Double Toroidal-Shaped Tile One of the possible and well-known forms of the data patch tile is a double toroidal-shaped tile which is created from given input (dynamic or still) texture sample. The double toroidal-shaped tile is a tile of which the top edge optimally fits the bottom edge, and simultaneously the right edge optimally fits the left edge. For calculating such type of tiles an (sub)optimal fit of opposite edges must be achieved to suppress the visual perception of edge existence. For creating nontrivial and visually high-quality results more than one tile must be found, so computed suboptimal

tiles are a compromise between multiple tiles. For tile of size of M pixels in height and N pixels in width, the top vertical edge is common parametric curve:

$$Q^{v+}(u) = [x(u), y(u)], u \in \langle 0..N \rangle, \quad (2.1)$$

$$Q^{v-}(u) = [x(u) + M, y(u)], u \in \langle 0..N \rangle, \quad (2.2)$$

and subsequently for horizontal edges of tiles:

$$Q^{h+}(u) = [x(u), y(u)], u \in \langle 0..M \rangle, \quad (2.3)$$

$$Q^{h-}(u) = [x(u), y(u) + N], u \in \langle 0..M \rangle. \quad (2.4)$$

Thus tile convex surface can be written as:

$$Q(u, v) = Q^v(u, v) \cup Q^h(u, v) \quad (2.5)$$

$$Q^v(u, v) = [1 - v, v] \begin{bmatrix} Q^{v-}(u) \\ Q^{v+}(u) \end{bmatrix} \quad (2.6)$$

$$Q^h(u, v) = [1 - v, v] \begin{bmatrix} Q^{h-}(u) \\ Q^{h+}(u) \end{bmatrix} \quad (2.7)$$

Toroidal mapping To define mapping between toroidal surface and flat double toroidal tile let's use for simplicity rectangular tile (Figure 2.1) with sizes M a N .

$$Q^v(u) = [x(u), y(u)] \quad (2.8)$$

2.1.3 Toroidal Mapping

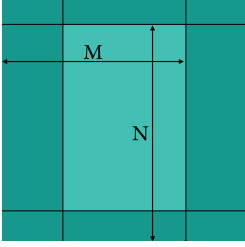


Figure 2.1: *Rectangular tile.*

The toroidal tile principles are based on a torus mapping. Here let's repeat the basics toroidal principles with illustration of a torus mapping. A rectangular double toroidal patch that corresponds with the unfolded surface of a torus has the size of M pixels in height and N pixels in width. Introducing both horizontal and vertical overlaps extends the $M \times N$ unfolded double toroidal patch by h additional rows and v additional columns.

The given rectangular texture patch is assumed to be indexed on the regular two-dimensional lattice:

$$I_{td} = (0, \dots, M-1) \times (0, \dots, N-1) , \quad (2.9)$$

where M denotes the number of rows and N the number of columns (See Figure 2.1, the darker areas on left/right and top/down are the overlapping areas.) of the patch tile. In case of rectangular double toroidal texture patch:

$$I_r = (0, \dots, M+h-1) \times (0, \dots, N+v-1). \quad (2.10)$$

The surface of a torus τ can be defined parametrically using poloidal and toroidal coordinates as follows:

$$x(s, t) = (R_1 + R_2 \cos s) \cos t , \quad (2.11)$$

$$y(s, t) = (R_1 + R_2 \cos s) \sin t , \quad (2.12)$$

$$z(s, t) = R_2 \sin s , \quad (2.13)$$

$$s, t \in [0, 2\pi) , \quad (2.14)$$

where s is poloidal coordinate, t is toroidal coordinate, R_2 is the radius of the tube and R_1 is the distance from the center of the torus to the center of the tube.

Mapping between rectangular lattice I_r and the torus surface τ , poloidal and toroidal angles are utilized as a generalization of poloidal and toroidal coordinates, respectively:

$$s, t \in [0, \infty) . \quad (2.15)$$

Then, $f_\tau : (r_1, r_2) \rightarrow (s, t)$ is the desired mapping function from the pixel position $r = [r_1, r_2] \in I^T$ to the poloidal angle s and toroidal angle t of a torus τ . They respectively, are defined as follows:

$$s = \frac{2\pi}{M} r_1 , \quad (2.16)$$

$$t = \frac{2\pi}{N} r_2 . \quad (2.17)$$

For all possible values from the range of f_τ , i.e., any pair of $s \in \{\frac{2\pi}{M}k \mid k \in \mathbb{N}_0^+\}$ and $t \in \{\frac{2\pi}{N}l \mid l \in \mathbb{N}_0^+\}$, inverse mapping $f_\tau^{-1} : (s, t) \rightarrow (r_1, r_2)$ can be meaningfully defined

using the following formulas:

$$r_1 = M \frac{s}{2\pi}, \quad (2.18)$$

$$r_2 = N \frac{t}{2\pi}. \quad (2.19)$$

Using parametric equations (2.11), (2.12), (2.13), and (2.15) of the torus τ , poloidal and toroidal angle can be projected onto the surface of τ . Obviously, the pair (s, t) projects to any point $P[x(s, t), y(s, t), z(s, t)]$ as well as pairs $(s + 2k\pi, t + 2l\pi)$, where $k, l \in \mathbb{Z}$. Such projection results in overlapping pixels.

2.2 Previous Results and Related Work

The spectrum of research work in this dissertation thesis covers several areas of Computer Vision, Texture Fidelity and Texture Analysis and Synthesis. Therefore this chapter consists of several themes and related topics overview. At first, related textural models are discussed here as key topics for dissertation thesistheme. Next, the other topics work as inpainting and texture similarity are discussed.

The classical and the intuitively best way to create a physically correct model of the natural scene for obtaining desired render (video). In this approaches the precise geometric created manually, semimanually using photogrammetry [121], or physical model is created and later by (usually approximated or simplified) physical laws simulated through time to create an output video, dynamic or static texture in the desired time momentum under given circumstances.

Many techniques for building physical models were applied for synthesizing textures, mostly the natural phenomena (for a more detailed overview see Barzel[5]). In most cases physically-based approaches are focused to one specific dynamic visual process for example gaseous phenomena [34, 42, 48], fire [98, 120], hair [7], trees and plants [14], or water [39, 49, 109] (with higher or lower approximation of physical laws). Some of them use particle systems which can be used to synthesize fire [102] or other types of textures and goals [103, 108, 117, 123] and [134] where some other approaches are used in cooperation of the particle-like model. Many methods work with three-dimensional procedural textures [69].

Physically-based approaches have proven successful also for walkinggaits [26, 41, 64], mechanical systems [5], and some of these approaches have already been used to synthesize videos representing artificial life scenarios [128].

These approaches have many good qualities - the quality of textures are directly proportional to the modeled scenes details and if the simulated laws are not approximated, but correctly simulated, the output scene can be real as its physical artwork is (which not trivially implies, that the quality of output texture do not decrease by the time). Also, the ability to accurately edit any scene is extremely advantageous as can be seen in Popovic [107].

Unfortunately, physically-based approaches have many disadvantages. Models are usually adapted to simulate a particular physical phenomenon. The flexibility of a created

model is low too since the model must often be created for every scene separately. Although they are physical models, still is in major part only approximation of phenomena, and in some cases (i.e., water) it is complicated to establish them.

It is obvious that in case of simulating scene by geometric models there are two crucial and mutually complementary areas - the exact geometry of scene and material / textural properties. The modelling of the complex scene consisting, i.e. natural objects trees, leaves, grass of complex shape like water surfaces is extremely difficult and time demanding even using auxiliary geometry modelling techniques such as photogrammetry [93, 94, 89, 121] or procedural modelling, i.e. L-systems [83].

Many other approaches attempt not to build a robust model, but simply compute directly on the texture. As we later explain, we can say these approaches belong to *statistical models*, hence they use some additional information but not to build the exact physical model.

The pioneering work by Nelson and Polana[96] perform dynamic texture (or temporal textures as called by authors) classification based on temporal features extracted by segments of dynamic textures. More interestingly Nelson and Polana suggest categories from visual motion in their later article [106]: temporal textures, activities and motion events. Both activities and motion events can be described more or less precisely - activities (walking, jumping, car movement, etc.) as motion patterns that are strictly periodic in time and localised in space, and motion events (e.g., opening door or falling tree) that do not show temporal or spatial periodicity. In contrast, temporal textures exhibit some statistical regularity but have indeterminate spatial and temporal extent. The term *temporal textures* has gradually changed into a *dynamic texture*. Authors continued with [105] which focus to recognition with a model based on optical flow normal component features. Their approach is based only on non-rigid motion instead of (grayscale) picture itself. By this only temporal characteristics are used, and a large amount of information remains unused. This approach based only on *optical flow* is, of course, inaccurate for many types of textures.

In a later article, Nelson and Polana used *spatio-temporal curves* to track parts of texture to characterize and find periodic motions by using a centroid of the pack of moving pixels through frames. Every pack has one bounding box divided into smaller rectangular cells, and the motion magnitude in each cell is computed and summed. These sums of motions vectors are used as features for finding a periodic pattern by using of Fourier analysis.

Similar like Nelson and Polana[105] who used only temporal characteristics, Allmen and Dyer[2] used *spatio-temporal curves* that are tangent to the optical flow by using of Runge-Kutta approximation method for differential equations. They presented some result on synthetic textures (probably to overcome problems with noise, occlusion or disappearing of small objects).

Seitz and Dyer expand their work with another article [114] that goals to detect the cyclic motion. Their work needs fixed viewpoint but works well for some specific types of textures (f.e. X-rays of a heart).

Some other minor work used this approach based on computing spatio-temporal curves.

Goulg and Shah[53] try to overcome a problem with disappearing object and regenerate continuity of curves, Anderson, Bur and van der Wall[3] similar like Heeger [63] used the motion energy as features and tries to use them with other techniques (pyramid representation instead of a rectangular grid) to track objects in the texture.

Approaches can also be divided according to their access to the texture data too - most common methods use Gabor filters (i.e. Bigun[12]), steerable filters by Freeman[51], Heeger[61], or Simoncelli[116]. Of course, the finding of the best filter can be part of model learning itself like in Zhu[147]. Many approaches like De Bonet[25], Efros[37], Wei[140] or Lin[82, 79] and others represent the texture in the form of Markov Random Field - the output texture is represented like a regular grid of nodes, where each node represents one pixel in the texture and marginal probability of a pair of nodes depend on the similarity of corresponding pixels data (typically red, green and blue values, but depends to a color model and similarity metrics). By this, the solution can be rewritten like the finding maximal likelihood of nodes in the grid. Usually, some approximation is searched, because this formulation called the problem of probabilistic inference is proven to be NP-hard.

There are some other statistical models, for example, based on fractals and their self-similarity across scales (which determine the fractal dimension). Heeger and Pentland[62] describe a fractal model for simulating gaseous phenomena like clouds, tree leaves and other turbulent flows. Similarly Stam[119] use Fourier analysis to model turbulent wind gaseous phenomena in similar ways as Li[76], who suggest similarly based segmentation model.

When some model, but not precisely physical base, is created and the output texture created by mathematical model simulating, we can speak about *parametric model*.

The modeling of a dynamic texture as a (linear) spatiotemporal auto-regressive model (STAR) was studied by Szummer[124, 125] who modeled grayscale dynamic textures like a spatiotemporal causal model. Szummers autoregressive model is intuitively, but (as he present it) grayscale limit and computationally demanding.

Later, as said before, Doretto[28, 30, 31, 29] and Soatto[118] presented auto-regressive moving average (ARMA) process based on SVD. This approach contains time-consuming iterative gradient method for parameters estimation and works in most cases on grayscale textures only, but allow to editing synthesized texture by changing of model parameters. Fast synthesis of dynamic color textures was later presented by Filip[44] Chan[15] or Constantini[20, 19] who use tensor decomposition and SVD techniques and others [132][84] presented similar approaches. Since parametric models are very general and usable for recognition[112] it can be difficult to model many types of dynamic and video textures.

The other approaches for the synthesizing of textures are non-parametric and use some samples to represent the texture. Many approaches from this class are focused on enlarging textures only in spatial dimensions, but the addressed problem is similar, and in some cases, it can be generalized to the time dimension (but in major part the simple and direct generalize is impossible). The sample of a texture can be the whole frame like presented Schödl[113] or noticeably smaller like published De Bonet[25] who used multi-scale-filter to analysis and synthesis of texture. Later works of Efros[36], Wei[140] and others im-

prove this area. The part of this class which represents texture, not pixel-per-pixel, but by some larger patch is intensively studied by Efros[37], Kwatra[73, 72] and others[143, 134]. Wang[133] represent texture like a linear superposition of deformable patches which position is derived by MRF. Basically, these approaches are create by intelligent sampling of input data subspaces in different form by copying whole patches from input texture with some offset.

Procedural techniques are a relatively quick solution for the purpose of synthesis but often offer lesser semantic and intertexture view because there are not learnable statistical models which result in the lack of flexibility for purposes as recognition and classification.

The dynamic texture models can be used to synthesizing, texture enlargement, restoration, or many other purposes. The prominent one is the *inpainting* and error concealment.

The term inpainting was introduced by Bertalmio[11], but methods for hole fitting were researched earlier, e.g., Masnou and Morel[90] replaced arbitrary shaped areas in images by other content by level-lines to disocclude the hole to inpaint or Roosmalen[131] use autoregressive models and MRF-like approach. Inpainting methods often use texture synthesis methods like quilting[36, 37], graphcut [73], exemplars [21], Fourier transformation and decomposing to structure image and a texture image[8, 9], and several others[68]. These methods still mainly concentrate only to image inpainting or works with videos in another ways - i.e., Liu[85] and Huang[67] used a similar method to video stabilization.

In the area of DT (or video) processing focused to video inpainting, there are two major approaches, mostly consisting of the previously mentioned synthesizing method and models. The first one is focused on temporal dimension and infinite looping data creation and the second one is focused on spatial dimensions and similarly as the image inpainting, on the hole fitting. The Schödl's method [113] is a typical temporal focused approach which can be with some adjustment utilised or used as inpainting methods. They typically work frame by frame, but it can only enlarge duration of a video. Similarly, Liao[77] and Agarwala[1] looped whole video by different ways, part of it or only some dynamic texture that occurs in videos as part of mix-of-DTs. The second main approach is focused mainly on spatial dimensions. Although this area was once primarily due to computational complexity neglected, it is now widely researched. Newson[97], Wexler[141], Granados[55], Patwardhan[101], Strobel[122], Ebdelli[33], Huang[66] or Xie[142] are the typical representative of video inpainting approaches, which can handle camera motion, occluding, and different illumination. These methods typically segment input to background and foreground and use small patches and energy minimizing border line error criterion to fill up the holes. Usually, the inpainting task required a user given mask [10, 33] even if is marked as an automatic algorithm.

It is interesting that only a very small part of current approaches (like Kwatra[73], Ding[26], Lin[78], Lguensat[75], Liao[77] or Huang[66]) works with input data as dynamic textures. The non-DT approaches typically suffer from a false optical flow, artifacts from deleted objects [55] or loss of frequencies in inpainted areas [97] and high computational requirements as they usually use computational demanding approaches as EM-like-optimization in [66], or neural convolution nets as in [142].

Nowadays popular energy-based convolution neural network models like [142] allow to synthesize and inpaint large part of dynamic textures. Even if these approaches are promising in restoring a large number of smaller discontinuous areas like salt and paper noise, they lack the consistent visual quality, creating the visually easy recognizable silhouette and suffer from time-consuming extremal calculations and large learning data requirements.

Another great part that has connection to this dissertation thesis are texture *similarity metrics*. Whereas the dynamic texture basically consists of a set of images (and moreover because of significant lack of dynamic texture criterions) the similarity ranking of still texture is taken into account.

Mutual similarity assessment and similarity ranking of two or more visual textures is a challenging problem due to real material textures complex dependencies on 16 physical observation parameters as described for example in [56].

Evaluation of how well various texture models conform to human visual perception is essential not only for assessing the similarities between model output and the original measured texture but also for optimal settings of model parameters, for a fair comparison of distinct models, material recognition, etc. This problem is not satisfactorily solved even for simpler static textures [58, 71].

Currently the only reliable, but incredibly impractical and expensive option is to exploit the methods of visual psycho-physics. The methods for psycho-physical testing require a lengthy process of experiment design, tightly controlled laboratory condition, and above all representative panel of human testing subjects. Such testing obviously cannot be performed on a daily basis. Several published static texture criteria allow testing selected texture properties such as the texture regularity as presented by Lin[80], etc. Others like Rubner[111] or Zujovic[148, 149] claim to test general texture quality.

The recent test [58] on a public texture fidelity benchmark of several state-of-the-art image quality measures and several recently published static texture criteria confirms their insufficient reliability and low robustness. The evaluated criteria were - the structural similarity (SSIM) index[136], the visual information fidelity (VIF) methods[115], the visual signal-to-noise-ratio[18] (VSNR), the mean-squared error[135] (MSE), the complex wavelet - structural similarity (CW-SSIM) index[138], and the structural texture similarity measure (STSIM-1, STSIM-2, STSIM-M) by Zujovic[148]. The results have demonstrated, that the standard image quality criteria (MSE, VSNR, VIF, SSIM, CW-SSIM) do not correlate well with a human quality assessment of textures at all. Inasmuch as the criteria are inaccurate in the field of static textures, their inappropriate behaviour in the time domain can be assumed.

Although the STSIM texture criteria have a significantly higher correlation with human ranking, they do not successfully solve similarity problem. The textural qualitative fully multispectral criterion by Kudělka[71] based on the generative Markov texture model statistics slightly outperforms the best alternative - the STSIM fidelity criterion.

All previous texture similarity criteria can be formally generalized also for dynamic textures if they are applied to the corresponding frame couples of compared dynamic textures and subsequently combined these partial results. While the mentioned methods

like regularity ranking by Lin[81], quality evaluation by Rubner[111], Zujovic[148, 149], or structural similarity by Wang[137] and other method intended for static textures can be used frame-by-frame the criterion directly fitted to a temporal domain are more appropriate.

The another possibility of similarity ranging is of course possibility to evaluate similarity of parameters from known dynamic texture models like ARMA[29], STAR[126] or others. However, admissible ranking only for some oversimplified tests such as identical dynamic textures which differ only in additive noise level can be expected.

Toroid-Shaped Patch Based Dynamic Texture Model

As there is no general textural model capable to model vast amount of possible dynamic texture realisation, this chapter consists of our novel model description which can overcome the most of current methods drawbacks. The examples of the enlargement and synthesis results are in Figure 3.1.

The sections in this chapter are as follows. Firstly, the basic principles and potential of the created model are stated in the Sec. 3.1. Next, in the Sec. 3.2, the crucial information about the toroidal method principles are defined. The toroidal method is then extend to temporal dimension in Sec. 3.3 in spatial and in Sec. 3.4 in temporal toroidal property keeping. Finally, the synthesis process is described in the Sec. 3.5.



Figure 3.1: *Synthesized dynamic texture teaser*: Examples or a proposed method results.

3.1 Method Principles Overview

The toroid-shaped patch dynamic texture model is capable of texture analysis, synthesis, enlargement, and editing. The model is developed concerning maintaining several vital requirements which consist mainly of the visual quality of the synthetic dynamic texture, editing capabilities of the approach, performance of the algorithm and dynamic texture compression. Also, the requirement of creating infinitely large output is a critical feature of the method. A synthesised dynamic texture should ideally produce such a visual perception so that original measured dynamic texture and the synthetic one are visually indiscernible.

The visual quality of the synthesized dynamic texture is naturally determined and limited by the chosen dynamic texture sample (or samples). Moreover, the used sample (or samples) represent whole dynamic texture - its dynamics, structure, color, etc. The representativeness and descriptiveness of the picked sample (or samples) is the essential prerequisite for visually plausible synthetic results. The intelligent sampling allows maintaining these requirements by choosing samples that consist of crucial representative texture parts. Also, observations-nuisance-factors like an artifact of unwanted regularity must be avoided.

The presented method principle extracts one or more mutually interchangeable texture patches with toroid-shape with borders optimized by minimal error boundary cuts. The extracted triple toroid-shaped dynamic texture patches can be due to toroidal property subsequently seamlessly repeated in all data space directions spatial as well as temporal. The method estimates optimal triple toroid-shaped tile and several triple toroid-shaped patches. Dynamic texture tiles are created by combining estimated (dense) optical flow and double toroid-shaped dynamic texture frame tiles. The basic principle of a presented method can be seen in Figure 3.2 together with a comparison to different typical approaches. The bottom three images illustrate methods patch and suggest the computational difficulty of the methods. The top three images demonstrate problems with the keeping of significant frequency from a given input texture.

The crucial characteristic of the proposed method is analytical part strictly separated from the synthesis step. Given these steps separated each part of approach can be implemented independently. The synthesis step then consists only by copying data patches and takes negligible computation. Due to separating and storing the crucial data patches only the compression of dynamic texture is possible. Moreover, this allows to implement different synthesis part - different arrangement, used driven arrangement, or even random arrangement only.

The presented toroid-shaped approach can be used on 4D multispectral dynamic textures, 3D monospectral dynamic textures or even still textures. As the temporal dimension has different visual perception properties than spatial dimensions, the principle of spatial and temporal cuts creating is similar but differs. Two distinct approaches for the spatial and optical flow driven temporal cuts respectively were then developed.

The reason of developing two distinct approaches the spatial and temporal approach instead of one general spatiotemporal method is widely used but an improper assumption of equality of spatial and temporal dimensions instead of accepting the temporal dimension

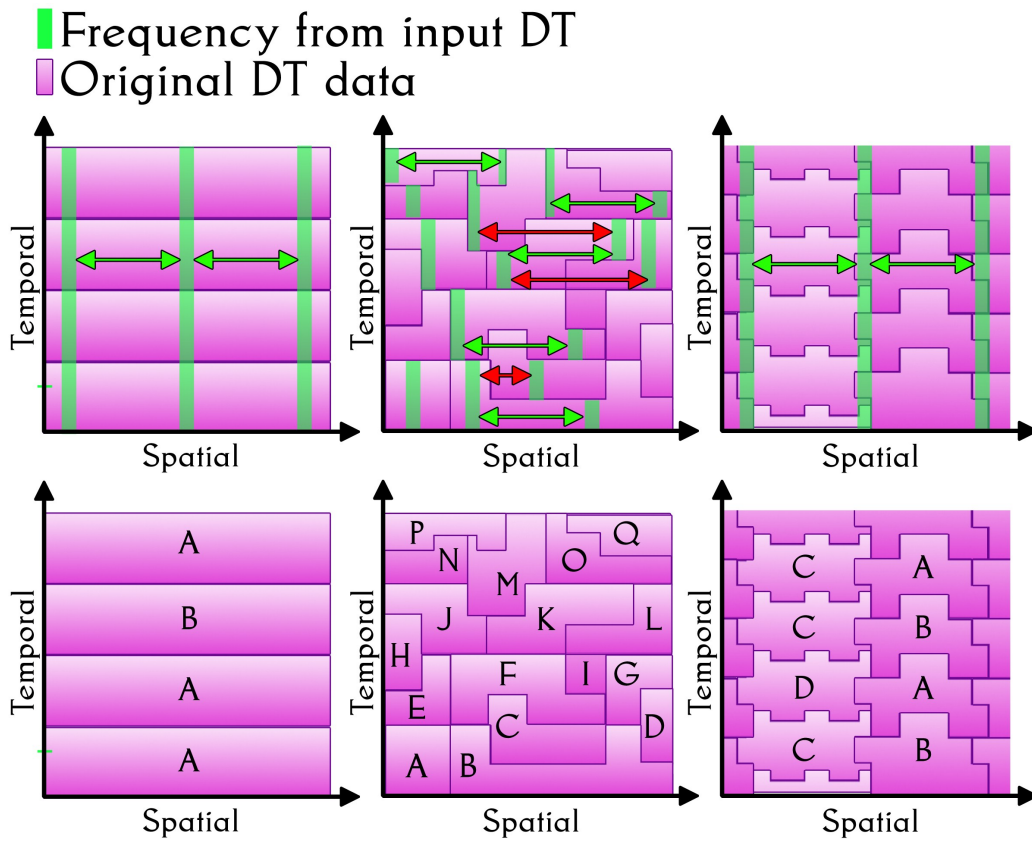


Figure 3.2: *Comparison of standard DTs synthesis approaches:* Left column only temporal enlargement [113], middle column spatiotemporal enlargement [97, 142, 73] and proposed spatial and temporal on the right. Each letter represents a different data patch and requires its own calculation.

as different and with a unique property. The justification can be easily seen from the spatial and temporal property of dynamic textures - a human perceives and detects errors differently in the space and time domain [35], e.g. the well-known visual phenomenon called filling-in have different behaviour in temporal and spatial domain [144].

3.2 Toroidal-Shaped Tile Approach

The triple toroid-shaped tile is a patch from the original data with the specific property to have toroidal border condition in each spatial (horizontal and vertical) and temporal dimensions. The textural tile is assumed to be indexed on the regular three-dimensional toroidal lattice. The lattice size depends on the estimated texture periodicity.

The optimal tile cut search algorithm which respects its toroidal border condition produces a tile which can be seamlessly repeated in both spatial and temporal dimensions; thus the boundary between two tiles placed abreast is invisible. The main idea of the triple toroid-shaped tiling is to find some relevant and sufficiently representative patches from the source dynamic texture, which can be directly seamlessly copied to the output dynamic texture, where the offset is driven by the patch placed on the three-dimensional lattice of the patch label.

Recall that the term *tile* noted as \mathcal{T} represent toroidal shape while term *patch* noted as \mathcal{P} is bounded to particular data that fill the tile shape. The dimensions of both media consist of two spatial dimensions only for tile and two spatial and one spectral dimension for a data patch. The extent of tile and patch to temporal dimension can be found in the following section (Sec. 3.3).

3.2.1 Double Toroid-Shaped Tile

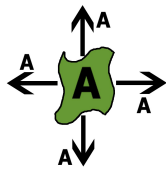


Figure 3.3:
Double toroid-shaped tile.

Before the full triple toroid-shaped tile method finding let's introduce the simpler double toroid-shaped tile finding and property. The double toroid-shaped tile, or patch from the original data, is two-dimensional shape with the specific property to have toroidal border condition in each spatial (horizontal, vertical) dimension (See Figure 3.3). The textural tile is assumed to be indexed on the regular two-dimensional toroidal lattice.

For simplicity, let's assume that only one double toroid-shaped tile. The toroidal property ensures that the top edge has precisely the same shape as bottom tile and simultaneously the left edge has exactly the same shape as right edge. Note that the double toroid-shaped tile described in this section is addressed in one particular frame in whole textural space \mathcal{Y} .

Overlap Error

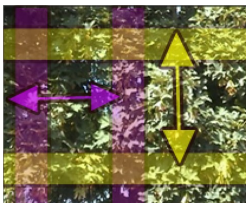


Figure 3.4: *Tile overlap*

For determining the overlap area size in which the optimal cut can be found the distance between the overlap areas need to be computed (the illustration is in the Figure 3.4). Evaluation of the overlap size is based on pixel error on areas and particularly on the error between corresponding pixels from overlapping marginal and corner parts. The similar pixel error is subsequently used to optimal cut findings in the founded areas.

Let us define the square overlap error arrays $\check{\epsilon}^h$ and $\check{\epsilon}^v$ and their appropriate elements for a multispectral pixel vector \mathcal{Y}_r as follows:

$$\check{\epsilon}_r^h = |\mathcal{Y}_r - \mathcal{Y}_{r+[N-h,0,\bullet,0]}|^2 \quad \forall r \in I_h, \quad (3.1)$$

$$\check{\epsilon}_r^v = |\mathcal{Y}_r - \mathcal{Y}_{r+[0,M-v,\bullet,0]}|^2 \quad \forall r \in I_v, \quad (3.2)$$

where the $dim = 2$ vector L_2 norm is defined as:

$$\epsilon = |X_{r_1, r_2, \bullet, r_4}^{p, q}|_{dim} = \left(\sum_{s=0}^{r_3} |X_{r_1, r_2, s, r_4}^{p, q}|^{dim} \right)^{\frac{1}{dim}}, \quad (3.3)$$

$$X_{r_1, r_2, s, r_4}^{p, q} = (\mathcal{Y}_{r_1, r_2, s, r_4}^p - \mathcal{Y}_{r_1, r_2, s, r_4}^q), \quad p, q \in I_r \quad (3.4)$$

The multiindex r has four components $r = [r_1, r_2, r_3, r_4]$, where the components are row, column, spectral, and frame indices, respectively. Operator \bullet denotes all values of the corresponding index. Remind that the overlap rectangular regions are defined as:

$$I_v = (0, \dots, h) \times (v, \dots, M), \quad (3.5)$$

$$I_h = (h, \dots, N) \times (0, \dots, v), \quad (3.6)$$

and remind that whole rectangular tile is defined as:

$$I_r = (0, \dots, M+h-1) \times (0, \dots, N+v-1). \quad (3.7)$$

Corner areas

Note that the corner area evaluation is more complicated. In fact, the overlapping corner area has more than only one corresponding area. The diagonal area part must be taken into account. The corner areas and its index set is defined as:

$$I_d = (1, \dots, h) \times (1, \dots, v). \quad (3.8)$$

To handle corner areas, the error arrays $\check{\epsilon}^h$, $\check{\epsilon}^v$ and $\check{\epsilon}^d$ with specific horizontal, vertical and diagonal shift and elements $\check{\epsilon}_r^v$, $\check{\epsilon}_r^h$ and $\check{\epsilon}_r^d$ for particular pixel Y_r are calculated as follows:

$$\check{\epsilon}_r^h = (\mathcal{Y}_r - \mathcal{Y}_{r+[N-h,0,\bullet,0]})^2 + (\mathcal{Y}_{r+[0,M-v,\bullet,0]} - \mathcal{Y}_{r+[N-h,M-v,\bullet,0]})^2, \quad r \in I_d, \quad (3.9)$$

$$\check{\epsilon}_r^v = (\mathcal{Y}_r - \mathcal{Y}_{r+[0,M-v,\bullet,0]})^2 + (\mathcal{Y}_{r+[N-h,0,\bullet,0]} - \mathcal{Y}_{r+[N-h,M-v,\bullet,0]})^2, \quad r \in I_d, \quad (3.10)$$

$$\check{\epsilon}_r^d = (\mathcal{Y}_r - \mathcal{Y}_{r+[N-h,M-v,\bullet,0]})^2 + (\mathcal{Y}_{r+[0,M-v,\bullet,0]} - \mathcal{Y}_{r+[N-h,0,\bullet,0]})^2, \quad r \in I_d, \quad (3.11)$$

which can be more comprehensive written down as $\check{\epsilon}$:

$$\check{\epsilon}_r^x = (\check{\epsilon}_r^v + \check{\epsilon}_r^h + \check{\epsilon}_r^d), \quad r \in I_d. \quad (3.12)$$

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

The horizontal and vertical overlap error are calculated as sum of all particular pixel overlap error weighted by appropriate overlap region size on the ζ area.

$$\zeta = I_v \cup I_h \cup I_d , \quad (3.13)$$

The optimal overlap is then found by minimizing the error:

$$\zeta^* = \min_{\zeta} \left\{ \frac{1}{[\zeta]} \sum_{\forall r \in I_{\zeta}} \epsilon_r^{\zeta} \right\} , \quad (3.14)$$

where $[\zeta]$ denotes the corresponding overlap area of ζ . I_{ζ} is the overlap area index set.

The particular total overlap error then consist of differences between overlapping horizontal and vertical regions. The appropriate errors $\mathring{\epsilon}$ and $\mathring{\epsilon}$ are calculated as the sum of all pixel errors in the overlap region weighted by the number of all pixels:

$$\mathring{\epsilon} = \left(\frac{1}{(h(N-v))} \sum_{r \in I_h} \epsilon_r^{\mathring{H}} \right) + \left(\frac{1}{h(N-v)} \sum_{r \in I_v} \epsilon_r^{\mathring{V}} \right) , \quad (3.15)$$

$$\mathring{\epsilon} = \frac{1}{2hv} \left(\sum_{r \in I_d} \epsilon_r^{\mathring{H}} + \sum_{r \in I_d} \epsilon_r^{\mathring{V}} \right) , \quad (3.16)$$

for every overlap area type. Or more readily for horizontal and vertical areas:

$$\mathring{\epsilon}^H = \frac{1}{h(N+v)} \left(\sum_{r \in I_h} \epsilon_r^{\mathring{H}} + \sum_{r \in I_d} \epsilon_r^{\mathring{H}} \right) , \quad (3.17)$$

$$\mathring{\epsilon}^V = \frac{1}{v(M+h)} \left(\sum_{r \in I_v} \epsilon_r^{\mathring{V}} + \sum_{r \in I_d} \epsilon_r^{\mathring{V}} \right) , \quad (3.18)$$

Considering one corner overlap only the overlapping corner error can be calculated as the following sum:

$$\mathring{\epsilon} = \frac{1}{hv} \sum_{r \in I_d} e_r^x . \quad (3.19)$$

Optimal Spatial Cut

With given optimal overlap the exact optimal cut which distinguishes between both cut sides can be found. Every optimal cut then divides overlap area to two-part - top/bottom and left/right which creates the desired toroidal shape (the illustration is in Figure 3.5). The searching for optimal cuts is adequate to find the minimal distance on simple vertex weight labeled undirected lattice graph

$$G = (V, E, w) , \quad (3.20)$$

where V is set of all vertices, E is set of all edges. Every edge is a pair of two vertices from V on rectangular 2D lattice with size $v \times h$, $h \times v$ and $(v - h) \times (h - v)$ for horizontal, vertical and corner overlaps respectively.

The E sizes can be - as the G is lattice graph - computed as $(M - 1) \times N + M \times (N - 1)$. The w is vertices weight function

$$w(V) \rightarrow \langle 0, \infty \rangle , \quad (3.21)$$

that is crucial for successfully representing the overlaps as graph G . The size of V is given by $\lfloor \zeta \rfloor$ and appropriate $\lfloor I_v \rfloor$, $\lfloor I_h \rfloor$ and $\lfloor I_d \rfloor$.

The search for optimal cuts for both the horizontal and vertical edge is equivalent to find minimum cost path in two-dimensional lattice G from the bottom to the top row and similarly from left to right column (with some adjusting). The optimal cuts cannot be found independently - the critical toroidal property must be kept thus some adjustment is necessary. To ensure that the tile optimally fits the adjacent one, horizontal and vertical minimal error boundary cuts are searched within the optimal horizontal and vertical overlap regions only with adjusting the start and end node which must have the same vertical/horizontal index or indices within the $\lfloor \Delta \rfloor$ neighbourhood. This implies that due to toroidal property desired optimal cuts parts have limited degree of freedom - apparently, the cut termination is strictly given by a start part. Moreover, intersections of spatial cuts correspond to vertices comprising the rectangle and cuts in the four corner part must be identical one.

Spatial minimal error boundary cuts are searched in the error lattice ϵ^h and ϵ^v defined by appropriate ϵ_r^h and ϵ_r^v values respectively. The whole spatial cut consists of two adjacent lattices defined as:

$$I_V = I_v \cup I_d , \quad (3.22)$$

$$I_H = I_h \cup I_d , \quad (3.23)$$

which are both defined simply as weighted composition of both appropriate lattice values:

$$\epsilon_r^H = \begin{cases} \frac{1}{6} (\epsilon_r^h + \epsilon_r^v + \epsilon_r^d) & r \in I_d , \\ \epsilon_r^h & r \in I_h \end{cases} \quad (3.24)$$



Figure 3.5: *Spatial cuts* illustration.

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

for the horizontal error lattice and for the vertical error array is defined as:

$$\epsilon_r^V = \begin{cases} \frac{1}{6} (\epsilon_r^h + \epsilon_r^v + \epsilon_r^d) & r \in I_d, \\ \epsilon_r^v & r \in I_v. \end{cases} \quad (3.25)$$

The $\frac{1}{6}$ ensures the same range of both error arrays parts. For illustration of cut search and arrays arrangement in DT see Figure 3.5 and Figure 3.6. Remind that due to toroidal property the first and last cut points are the same and thus the error lattices are cylinders. Moreover, as the corner diagonal parts are precisely the same, the lattice can be mapped as torus where a one cut is longitudinal and other is latitudinal.

The desired spatial cuts C_h and C_v are series of nodes on lattice graph indices in the spatial dynamic texture subspace - frame). The appropriate cuts from regions I_V and I_H are defined as:

$$C_h = \{C_r[i_r, j_r] \mid C_r \in I_H, r = 1, \dots, n_H, j_1 = 0, j_{n_H} = N-1\}, \quad (3.26)$$

$$n_H \in ([I_H^h], [I_H^h][I_V^h]),$$

$$C_v = \{C_r[i_r, j_r] \mid C_r \in I_V, r = 1, \dots, n_V, i_1 = 0, j_{n_V} = M-1\}, \quad (3.27)$$

$$n_V \in ([I_V^v], [I_V^v][I_H^v]),$$

where the length of cuts (total number of cut nodes) are denoted as n_V for vertical and n_H for a horizontal cut. Note that the n_V and n_H are not equal to the appropriate lattice size $[I_V]_v$ and $[I_H]_h$ as the series C_V and C_H are not monotone. In the case of $n_V = [I_V]_v$ and $n_H = [I_H]_h$ the spatial cuts C_* and subsequent i_0, \dots, i_{n_V} and j_0, \dots, j_{n_H} are monotone series $s_n = s_{n-1} + 1$. Say that in the case monotonic series the dominant dimension is horizontal one for C_h and vertical for C_v . If the $n_V > [I_V]_v$ the $C_k \rightarrow C_{k+1}$ should be any node that is not from $C_{0, \dots, k}$.

Neighbourhood

The pixel or node neighbourhood for a given r is denoted by Δ_r operator. The neighbourhood could be monotone with respect to some dimension which is in all cases the dominant dimension of the area in which the pixel/node come from. The monotone neighbourhood is noted as Δ_m .

The neighbourhood Δ_m^8 represent well known 8-neighbourhood which is, in the monotonic case of Δ_m^8 and limited to only 3 adjacent values (3.34). The Δ_m^4 similarly denotes 4-neighbourhood. The hierarchical monotonic neighbourhood for given r with size of p is then denoted as $\bar{\Delta}_m^p$ (see eq. (3.35)).

The neighbourhood Δ_{C_k} for node C_k can be defined as 4-neighbourhood ($\Delta_{C_k}^4$), 8-neighbourhood ($\Delta_{C_k}^8$), p -distance monotonic neighbourhood ($\Delta_{C_k}^{m_p}$), or any else. The neighbourhood definition is determined by a method used to found the optimal cut and by a textural property.

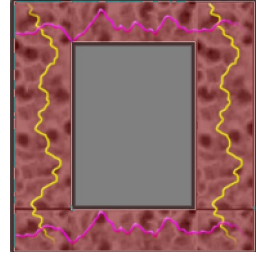
In case of using more general p -distance monotonic neighbourhood (Δ_m^p) the distance is defined in a similar manner with the possibility of greater step in the non-dominant direction.

3.2.2 Optimal Cut Search

The optimal cut total error $\tilde{\epsilon}$ is defined as sum of all pixel errors from which the cut consist. For the cut lines if is defined as:

$$\tilde{\epsilon}^H(C_h) = \frac{1}{n_H} \sum_{\forall r \in C_h} \epsilon_r^H, \quad (3.28)$$

$$\tilde{\epsilon}^V(C_v) = \frac{1}{n_V} \sum_{\forall r \in C_v} \epsilon_r^V. \quad (3.29)$$



Here, let's overlook the error lattice and optimal cut relation (see Figure 3.6, pixel error are illustrated as darker area, optimal cut are marked by yellow and violet lines. Note their the shape and temporal behaviour especially in the I_d areas.). For any given error lattice I and some series start point $[i_0, j_0]$ one of dimension values is defined as 0 where the another can take any valid value from I . For any start point $s = [0, j_0]$ or $s = [i_0, 0]$ the sub-optimal cut exist. Obviously, there exist $[I_H]_v$ and $[I_V]_h$ possible starting points. Denote the cut lines that starts (or even contains) any given point s as $C^{\{s\}}$. Hence the spatial cut $C^{\{s\}}$ is (in at least in one dimension) continuous line there exist at least $[I_H]_v$ and $[I_V]_h$ sub-optimal cuts that divide the lattice. The minimal spatial cut search then consists from minimizing the cuts error values and searching from appropriate optimal start points:

Figure 3.6: An optimal cut.

$$s^* = \arg \min_{\forall t \in I_V^{\{[0;n]\}}} \tilde{\epsilon}^V(C_h^{\{s\}}), \quad n \in (0, \dots, n_H), \quad (3.30)$$

$$t^* = \arg \min_{\forall t \in I_H^{\{[0;n]\}}} \tilde{\epsilon}^V(C_v^{\{t\}}), \quad n \in (0, \dots, n_V). \quad (3.31)$$

Subsequently, the optimal spatial error cuts searching consist of finding the minima error cuts than contains the start points s^* and t^* :

$$C_h^* = \arg \min_{C_h^{\{s\}}} \tilde{\epsilon}^H(C_h^{\{s^*\}}), \quad (3.32)$$

$$C_v^* = \arg \min_{C_v^{\{t\}}} \tilde{\epsilon}^V(C_v^{\{t^*\}}). \quad (3.33)$$

There are many ways to find particular optimal cuts specified as above. The alternative Kwatra approaches can be used, the graph algorithm as Dijkstra or A* for efficient cut search. Furthermore, the dynamic programming as more general and robust solution can be utilised.

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

Dynamic Programming One of many ways that are possible for an optimal toroidal cut search is to construct integrated error values arranged structures according to the selected adjacency nodal points (pixels). In cases of dynamic programming-like rules, the desired optimal cut is restricted to be monotonous in the way of dominant spatial dimensions for each horizontal and vertical case accordingly. For every horizontal and vertical cut search in the proper field I_h and I_v the lattice is taken as an oriented level graph in the direction of main spatial dimension. Each column/row for the horizontal/vertical case is considered as a single level of the graph structure, and the optimal cut is searched from the bottom to top level of a graph, and the appropriate start/end point has defined with the same row/column indices. The dynamic programming approach is limited to find the monotonous series only regardless of error definition. Naturally, this strictly limits the possibilities of using the neighborhood definition. The 8-neighbourhood (Δ_m^8) with limiting to be monotonic one in natural pick there. Note that due to restricting the connectivity is then defined as:

$$[i, j] \xrightarrow[\Delta_m^8]{v} \begin{cases} [i-1, j+1], [i-1, j+1] \in I_V, \text{ or,} \\ [i, j+1], [i, j+1] \in I_V, \text{ or,} \\ [i+1, j+1], [i+1, j+1] \in I_V, \end{cases} \quad (3.34)$$

in the vertical case and similarly in the horizontal dimension.

In case of using more general p -distance monotonic neighbourhood ($\bar{\Delta}_m^p$) the distance is defined in similar manner with possibility of greater step in the non-dominant direction. For a given p -distance is vertical connectivity defined as:

$$[i, j] \xrightarrow[\bar{\Delta}_m^p]{v} \begin{cases} [i-p, j+1], [i-1, j+1] \in I_V, \text{ or,} \\ [i-(p-1), j+1], [i-1, j+1] \in I_V, \text{ or,} \\ [i-(p-2), j+1], [i-1, j+1] \in I_V, \text{ or,} \\ \dots \\ [i, j+1], [i, j+1] \in I_V, \text{ or,} \\ \dots \\ [i+(p-2), j+1], [i+1, j+1] \in I_V, \text{ or,} \\ [i+(p-1), j+1], [i+1, j+1] \in I_V, \text{ or,} \\ [i+p, j+1], [i+1, j+1] \in I_V. \end{cases} \quad (3.35)$$

The horizontal and vertical cut search is then defined for Δ_+^8 by the according rules:

$$\varepsilon_r^{h+} = \overset{\circ}{\varepsilon}_r^H + \min(\varepsilon_{r+[-1,-1,\bullet,0]}^{h+}, \varepsilon_{r+[0,-1,\bullet,0]}^{h+}, \varepsilon_{r+[1,-1,\bullet,0]}^{h+}) , \quad (3.36)$$

$$\varepsilon_r^{v+} = \overset{\circ}{\varepsilon}_r^V + \min(\varepsilon_{r+[-1,-1,\bullet,0]}^{v+}, \varepsilon_{r+[-1,0,\bullet,0]}^{v+}, \varepsilon_{r+[-1,1,\bullet,0]}^{v+}) . \quad (3.37)$$

and for $\bar{\Delta}_m^p$ according to (3.35) as:

$$\varepsilon_r^{h+} = \overset{\circ}{\varepsilon}_r^H + \min \left\{ \varepsilon_{r-[c,1,\bullet,0]}^{h+}, \dots, \varepsilon_{r+[c,-1,\bullet,0]}^{h+} \right\} , \quad (3.38)$$

$$\varepsilon_r^{v+} = \overset{\circ}{\varepsilon}_r^V + \min \left\{ \varepsilon_{r-[1,c,\bullet,0]}^{v+}, \dots, \varepsilon_{r+[-1,c,\bullet,0]}^{v+} \right\} . \quad (3.39)$$

Like usually in the dynamic programming approaches the error lattice is filled from bottom to top (right to left) and traversed in the reverse direction. For lattice error composition and thus any given key start point p the values in the same row / column is set to ∞ . The recursive level per level error rule application that starts from the appropriate row/column fill the whole lattice by the corresponding values $\varepsilon_r^{h+} / \varepsilon_r^{v+}$. The optimal cut is then the minimal error series fun in the reversed main dimension direction.

Dijkstra Algorithm The search for optimal cuts for both the horizontal and vertical edge is equivalent to find minimum cost path in two-dimensional lattice G from the bottom to top row and similarly from left to right column.

The Graph G represent the error lattice. The edge relation is defined by the used neighbourhood relation. The full Δ^8 symmetric neighbourhood is the most general case which leads to non-monotonic cut lines while the Δ_+^8 or $\bar{\Delta}_m^p$ can create monotonic cut lines as the dynamic programming approach. The given node weights as defined as:

$$w(V^{\{i,j\}}) \leftarrow \varepsilon_{\{i,j\}}^\zeta . \quad (3.40)$$

The weight for every node is computed as $\overset{\circ}{\varepsilon}_r^H$ (3.24) and $\overset{\circ}{\varepsilon}_r^V$ (3.25) with $\tilde{\varepsilon}_r^H$ (3.28) and $\tilde{\varepsilon}_r^V$ (3.29) from the whole ζ area. The cut lines are then a result of a widely recognized Dijkstra algorithm.

Similarly, the optimal cuts C^* are found by multiple algorithm evaluations from the corresponding start and end points and final $\tilde{\varepsilon}$ minimization. Note that in the case of symmetric relation some sub-optimal cut lines $C^{\{s_j\}}$ and $C^{\{s_i\}}$ can be identical.

A* Algorithm If the resulted spatial cuts are desired to be a non-monotonical an A* algorithm can be used. The advantage of the pathfinding algorithm is apparently due to its possibility to evaluate distance to the desired final point with defining the heuristics to the final point as a L^2 distance from actual to the final point. Even if the tiles produces by an A* algorithm can be very effective, the problematic use of the algorithm results in a spatiotemporal domain event and its hard restrictions on assembling planes lads to its non-use.

Double toroid-shaped tile The subspace of a texture that is generated as an intersection of the space between horizontal and vertical cuts is called a tile. Even if the searching algorithm can vary, the tile has the same toroidal property. With a term tile, the found shape is called, contrary to term patch which is used to particular data including shift if

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

more patches are found in a given dynamic texture. The rectangular tile is a subspace of a whole dynamic texture or more precisely:

$$I_T = (0, \dots, M + h - 1) \times (0, \dots, N + v - 1) , \quad (3.41)$$

in which the I_h , I_v and I_d was detected. In the case that cuts are monotonic-series the spatial patch can be easily defined as union of the space between two vertical cuts and two horizontal cuts:

$$\mathcal{P} = (C_h^*, \dots, C_h^* + M) \cap (C_v^*, \dots, C_v^* + N) . \quad (3.42)$$

3.3 Dynamic Toroid-Shaped Tile

The located single frame double toroid-shaped tile found is propagated along the optical flow to subsequent frames to model the movement of the textural structures. The dense optical flow is used to represent a dynamic texture global dynamics. The propagated double toroid-shaped tile in every frame serves as the initialisation for local modification of such a double toroid-shaped tile to each frame-specific final shape.

3.3.1 Animated Double Toroid-Shaped Tile



Figure 3.7: *Animated tile.*

The toroidal animated tile (and subsequently all its derivate) is a three-dimensional shape where the bottom and down planes have the same shape as well as the right and left plane. The front and back plane are consist of double toroid-shaped tile I_T (eq. 3.2.2).

The overlap area in which the tile is computed (see Figure 3.7 for illustration, white arrows illustrates the tile movement) is moved by a local $\Phi_{\mathcal{P}}$ values to generate rectangular subspace I_{rt} . An optimal location for the triple toroid-shaped tile can be then found in the similar way that the double-toroid shaped tile

Temporal and dynamic dimension

As the animated double toroid-shaped tile (and all derived constructs) and patch are \mathcal{Y} subspace that posses more than only two spatial dimensions, the restriction of only one actual frame that was used in the previous section is no longer valid.

Moreover the multi-index r has from now five components $r = [r_1, r_2, r_3, r_4, r_5]$, where the components are row, column, spectral, frame and dynamics indices, respectively. The r_5 dimension is not stored in original data but computed. So exactly it is a property of function of a given r , but hence the optical flow Φ of the whole \mathcal{Y} is usually precomputed and assigned to appropriate locations. Note that as the r_5 is optical flow vector in the three-dimensional space, it has of course three values x, y and t respectively.

Temporal and dynamic error

Hence on now the r and all appropriate computing have more dimensions, the metrics between two corresponding overlapping pixels has to be updated. Moreover, the last dynamic dimension is three-vector. The simple difference of its components is not satisfying. As a r_5 is an optical flow vector the important properties are the vector size and their direction.

$$\epsilon(p, q) = |X_{r_1, r_2, r_3, r_4, r_5}^{p, q}|^{dim} , \quad (3.43)$$

$$|X_{r_1, r_2, r_3, r_4, r_5}^{p, q}|^{dim} = X_{dyn, r_1, r_2, s, r_4, r_4}^{p, q} + \left(\sum_{s=0}^{r_5} |X_{r_1, r_2, s, r_4, r_5}^{p, q}|^{dim} \right)^{\frac{1}{dim}} , \quad (3.44)$$

$$X_{dyn, r_1, r_2, r_3, r_4, r_4}^{p, q} = X_{dir, r_1, r_2, r_3, r_4, r_5}^{p, q} X_{val, r_1, r_2, r_3, r_4, r_5}^{p, q} , \quad (3.45)$$

$$X_{r_1, r_2, s, r_4, r_4}^{p, q} = (\mathcal{Y}_{r_1, r_2, s, r_4, r_5}^p - \mathcal{Y}_{r_1, r_2, s, r_4, r_5}^q) , \quad (3.46)$$

$$X_{dir, r_1, r_2, r_3, r_4, r_5}^{p, q} = 1 + \arccos \left(\frac{r_5^p \cdot r_5^q}{|r_5^p| |r_5^q|} \right) , \quad (3.47)$$

$$X_{val, r_1, r_2, s, r_4, r_5}^{p, q} = \left(\sum_{u=1}^3 |Z_{r_1, r_2, r_3, r_4, u}^{p, q}|^{dim} \right)^{\frac{1}{dim}} , \quad (3.48)$$

$$Z_{r_1, r_2, r_3, r_4, u}^{p, q} = (\mathcal{Y}_{r_1, r_2, r_3, r_4, u}^p - \mathcal{Y}_{r_1, r_2, r_3, r_4, u}^q) , \quad (3.49)$$

$$p, q \in I_{rt} . \quad (3.50)$$

The rectangular subspace of a \mathcal{Y} in which the \mathcal{T} is searched for is called a rectangular tile I_{rt} :

$$I_{rt} = (0, \dots, M) \times (0, \dots, N) \times (0, \dots, T) , \quad (3.51)$$

Even if the searching algorithm can vary, the tile has the same toroidal property. The animated (or dynamic) tile is a subspace of a whole dynamic texture \mathcal{Y} . The tile consists of temporal-horizontal (\hat{C}^{h*}) and temporal-vertical (\hat{C}^{v*}) spatiotemporal union with the exception of corner duplicate parts. The appropriate size of spatiotemporal cut planes are:

$$I_{h*} = (0, \dots, M) \times (0, \dots, T) , \quad (3.52)$$

$$I_{v*} = (0, \dots, N) \times (0, \dots, T) . \quad (3.53)$$

Mention, that for more fluently writing the temporal-horizontal plane and temporal-vertical spatiotemporal will be called as horizontal and vertical if the context is clear. The spatiotemporal \mathcal{P} is now defined as combination of the space between the horizontal and vertical spatiotemporal cut planes:

$$\mathcal{P}^A = (\hat{C}^{h*}, \dots, \hat{C}^{h*} + M) \cap (\hat{C}^{h*}, \dots, \hat{C}^{h*} + N) . \quad (3.54)$$

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

The most simplest spatiotemporal \mathcal{T} and \mathcal{P} can be created as set of \mathcal{T} and \mathcal{P} with increasing temporal indices.

The combination of a dynamics represented by a optical flow $\Phi_{\mathcal{P}}$ from every previous $\mathcal{P}_{(t-1)}$ can create an animated patch. The animated patch is then similarly defined as:

$$\mathcal{P}_i^A = (\hat{C}_r^{h*} + \Phi_{\mathcal{P}_{i-1}^A}^x, \dots, \hat{C}_r^{h*} + \Phi_{\mathcal{P}_{i-1}^A}^x + M) \cap (\hat{C}_r^{h*} + \Phi_{\mathcal{P}_{i-1}^A}^y, \dots, \hat{C}_r^{h*} + \Phi_{\mathcal{P}_{i-1}^A}^y + N) . \quad (3.55)$$

where i is from 1 to T and $\Phi_{\mathcal{P}_{i-1}^A}^x$ and $\Phi_{\mathcal{P}_{i-1}^A}^y$ are horizontal and vertical parts of an average optical flow computed on a appropriate \mathcal{P} and animated tile is defined in the similar manner. The resulting animated tile then replicate the texture movement but with boundary given from only one (first) frame.

3.3.2 Dynamic Double Toroid-Shaped Animated Tile

As the set of double toroid-shaped tiles of the same shape is called the animated tile, the tiles fitted to local dynamics are called a dynamic double toroid-shaped animated tile.

Hence the dynamic approach is chosen to double-toroid tile search the corresponding process is expended to the temporal domain with some changes that allows adapting to local and global optical flow and structural changes in the dynamic texture throughout time.

Theoretically, the optimal triple toroid-shape tile can be found by propagating the optimal double toroid-shaped \mathcal{T} tile from $\mathcal{Y}_{[r_4=0]}$ (0-th frame) to the rest of \mathcal{Y} .

$$\hat{\psi}_r^{v*} = \varepsilon_r^{v+} + \min \left(\hat{\psi}_{r+[-1,0,\bullet,1]}^{v*}, \hat{\psi}_{r+[0,0,\bullet,1]}^{v*}, \hat{\psi}_{r+[1,0,\bullet,1]}^{v*} \right) , \quad (3.56)$$

$$\hat{\psi}_r^{h*} = \varepsilon_r^{h+} + \min \left(\hat{\psi}_{r+[0,-1,\bullet,1]}^{h*}, \hat{\psi}_{r+[0,0,\bullet,1]}^{h*}, \hat{\psi}_{r+[0,1,\bullet,1]}^{h*} \right) , \quad (3.57)$$

and subsequently searching for the minimal error spatiotemporal cut \hat{C}_y^{h*} and simultaneously searching for the minimal error spatiotemporal cut \hat{C}_x^{v*} , where x and y values denote the order of starting point from optimal cuts \hat{C}^{h*} and \hat{C}^{v*} . The desired cut planes \hat{C}^{h*} and \hat{C}^{v*} is then union of all \hat{C}_y^{h*} and \hat{C}_x^{v*} respectively.

Unfortunately, this approach cannot lead to a visually satisfying solution. See this on an example of \hat{C}^{h*} : The founded local (in the manner of one column/row) optimal spatiotemporal cuts \hat{C}_y^{h*} and $\hat{C}_{y\pm 1}^{h*}$ has no relation and can be very different. Even if the spatiotemporal cut local changes are restricted to ± 1 due to (3.57) and (3.56) this property are not shared in the vertical dimension and values of $\hat{C}_{x,y}^{h*}$ and $\hat{C}_{x,y\pm 1}^{h*}$ can strongly vary.

To ensure that the \hat{C} property will be consistent in both spatial dimensions the \hat{C}_y^{h*} and \hat{C}_x^{v*} search has to be enhanced to keep the similar property in the primary dimension and in the secondary dimension too. In another word, the \hat{C} should be computed with respect to adjacent cuts.

Iterative spatiotemporal cut searching The simplest way to restrict the relation between two adjacent spatiotemporal cuts is iterative computation of \hat{C}_y^{h*} and \hat{C}_x^{v*} on base of previously found \hat{C}_{x-1}^{v*} and \hat{C}_{y-1}^{h*} respectively. The new spatiotemporal cut is then limited by previously found adjacent cut by its neighbourhood:

$$\hat{\psi}_r^{v*} = \varepsilon_r^{v+} + \min \left(\hat{\delta}_{\{r-[0,0,\bullet,1,\bullet]\}}^{vt} \cap \hat{\delta}_s^v \right) , \quad s = -\hat{C}_{r_2}^{v*} , \quad (3.58)$$

$$\hat{\psi}_r^{h*} = \varepsilon_r^{h+} + \min \left(\hat{\delta}_{\{r-[0,0,\bullet,1,\bullet]\}}^{ht} \cap \hat{\delta}_u^h \right) , \quad u = -\hat{C}_{r_1}^{h*} , \quad (3.59)$$

where $-\hat{C}$ denotes previously computed spatiotemporal cut and $\hat{\delta}_r^v$, $\hat{\delta}_r^h$, $\hat{\delta}_r^{ht}$ and $\hat{\delta}_r^{vt}$ are defined as spatial and spatiotemporal adjacent sets derived by used Δ neighbourhood, respectively. In the simplest case it consist of:

$$\hat{\delta}_r^v \xrightarrow[\frac{\Delta^s}{m}]{v} \begin{cases} \psi_{r+[1,0,0,0,0]}^{v+} , & \text{or} \\ \psi_{r+[0,0,0,0,0]}^{v+} , & \text{or} \\ \psi_{r+[-1,0,0,0,0]}^{v+} , \end{cases} \quad \hat{\delta}_r^{vt} \xrightarrow[\frac{\Delta^s}{m}]{v} \begin{cases} \psi_{r+[1,0,0,-1,0]}^{v+} , & \text{or} \\ \psi_{r+[0,0,0,-1,0]}^{v+} , & \text{or} \\ \psi_{r+[-1,0,0,-1,0]}^{v+} , \end{cases} \quad (3.60)$$

and $\hat{\delta}_r^h$ and $\hat{\delta}_r^{ht}$ is defined in similar manner.

The iterative spatiotemporal cut searching can be enhanced by using of different Δ , namely Δ_m^p . The used neighbourhood is not limited to be the same as in double toroidal-shaped tile search but should be consist. Note that using of general neighbourhood obviously lead to a general spatiotemporal plane and consequently higher computational time.

The iterative approach, of course, produces two solutions for every spatiotemporal plane search. The $\uparrow\hat{C}$ and $\downarrow\hat{C}$ according to chosen order of iterative process. Moreover, another problem occurs. Hence the tile borders are toroidal the repeated iterative finding of $\hat{C}_{C_y^h}^{h*}$ lead after $N - 1$ steps to $\hat{C}_{C_{y-1}^h}^{h*}$. Again, the problem is possibility inconsistency between this two cuts. The solution is excluding some space and thus limiting the possible solutions from the first cut in reverse order by repeating the neighbourhood driven δ relation.

Restrictive Spatiotemporal Cut Searching The drawback of an iterative solution process can by bypassed by two-steps optical flow driven tile animating process. The animated tile serves as the initialisation to define the rough area in which the exact dynamic spatiotemporal tile can be found (see Figure 3.8, note the areas marked around the already founded (sub)optimal cut).

At first an (dense) optical flow Φ in the whole \mathcal{Y} space is detected. The array with values of an optical flow vectors of whole dynamic texture \mathcal{Y} noted is as $\Phi_{I_{\mathcal{Y}}}$ and values of the optical flow vectors from corresponding pixel Y_r as $\Phi_{\mathcal{Y}_r}$. The average optical flow of some area, namely the patch \mathcal{P} , is noted as $\Phi_{\mathcal{P}}$

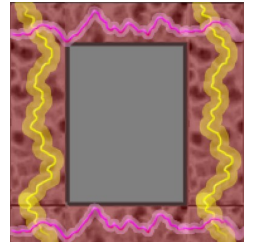


Figure 3.8: *Restrictive search.*

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

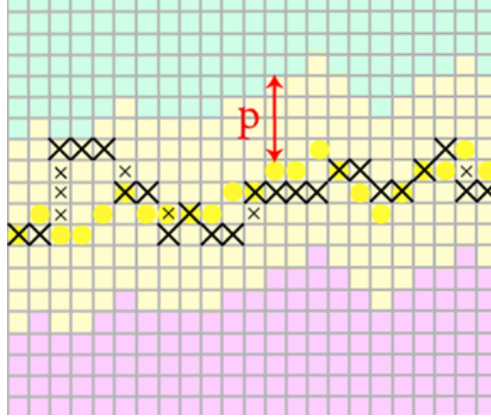


Figure 3.9: *Toroidal boundary location for the restrictive spatiotemporal cut searching:* Yellow squares represent border initialization, while the yellow area is the area where the precise location is searched for. Crosses represent the exact boundary, complementary points (small crosses) must be added to make the border continuous. The parameter p is driven by the actual optical flow.

The rough solution is generated by combining the $\Phi_{\mathcal{P}}$ with the \mathcal{T} boundaries frame by frame (Sec. 3.3.1). The resulting animated tile then replicate the texture movement but with boundary given from only one (first) frame. This animated tile represents two of the three main scene features. The global dynamics (scene or large objects movement, e.g., whole trees) is represented by the optical flow and the texture itself by the double toroid-shaped tile. However, local dynamic (like leaves or grass blades shivering) cannot be undoubtedly sufficiently represented this way.

The local dynamics problem can be solved by the spatiotemporal plane fitting. The animated tile then can be used as initialisation to plane fitting. The combination of the restricted neighbourhood with the animated tile generates an extended animated tile subspace area \mathcal{E} (see Figure 3.9) in which the optimal spatiotemporal cuts (3.56) and (3.57) can be found without violating the consistency of and the spatiotemporal cutting planes.

The extended animated tile subspace area remarkably speeds up the computation time needed to find the optimal tile shape in comparison to search the whole textural space. Moreover, this approach is beneficial also in avoiding some typical DT modeling problems like artifacts introduced by extremely varying speed or problems with highly dynamic texture, e.g., fire or water details.

$$\hat{\psi}_r^{v*} = \varepsilon_r^{v+} + \min \left(\hat{\psi}_{\hat{C}_{[i=r_2]}^{v*}+[0,-pv,\bullet,\bullet,-1]}^{v*}, \dots, \dots, \hat{\psi}_{\hat{C}_{[i=r_2]}^{v*}+[0,0,\bullet,\bullet,-1]}^{v*}, \dots, \hat{\psi}_{\hat{C}_{[i=r_2]}^{v*}+[pv,0,\bullet,\bullet,-1]}^{v*} \right) \quad (3.61)$$

$$\hat{\psi}_r^{h*} = \varepsilon_r^{h+} + \min \left(\begin{array}{c} \hat{\psi}_{\hat{C}_{[j=r_1]}^{h*}+[-p_h,0,\bullet,\bullet,-1]}^{v*}, \dots, \\ \dots, \hat{\psi}_{\hat{C}_{[j=r_1]}^{h*}+[0,0,\bullet,\bullet,-1]}^{h*}, \dots, \hat{\psi}_{\hat{C}_{[j=r_1]}^{h*}+[p_h,0,\bullet,\bullet,-1]}^{h*} \end{array} \right) \quad (3.62)$$

where $\hat{C}_{[i=r_2]}^{v*}$ denotes the r_2 -th value of spatiotemporal vertical cut plane and $\hat{C}_{[j=r_1]}^{h*}$ denotes the r_1 -th value of spatiotemporal horizontal cut plane and $p = [p_x, p_y, p_t]$ denotes the size of the boundary area (see Figure 3.9). The vector p is computed as average dense optical flow for the whole \mathcal{P} in the previous frame.

The resulting tile, whether created by an iterative or non-iterative method (restrictive approach is used in this dissertation thesis), respect all three dynamic texture properties. The global dynamics like whole scene movement is handled by tile animating and ensures the focusing of tile to the same structural part of a texture. The optimal spatial double toroid-shaped tile from the first frame roughly fit the exact spatial tile location. Despite the dynamics of the dynamic texture process through time the homogeneity of a dynamic texture keep this rough solution valid. The local structural changes, which are possible in the dynamic texture, are then handled by recomputing the exact tile shape by an iterative or restrictive method. This allows adapting the dynamic tile recomputing process to local dynamics which can vary throughout the time.

Multiple tiles

To disrupt visible periodicity of the synthetic texture generated using single double toroid-shaped tile, multiple mutually interchangeable tiles can be used. At first, a single, double toroidal tile of optimal size is found. Then multiple tiles are found as the most fitting patch with the same used tile or recomputed consensus tile. To allow efficient texture synthesis particularly from the visual quality point of view, it is necessary to find appropriate size of an optimal tile. Due to the assumption of homogeneity for the whole texture the computed tile size is optimal for the whole texture.



Figure 3.10: *Multiple rec. tiles.*

The desired patches $\mathcal{P}_{\{0,\dots,n_p\}}$ are then searched by minimizing the overlap error on shifted tile $\mathcal{T}_{r+\delta}$ with three dimensional δ shift (see the illustration in the Figure 3.10, where three tile locations with three generated patch are illustrated). Emphasize that the overlap area only has to be taken into account as the inner part of texture (non-colored areas in Figure 3.10) can vary.

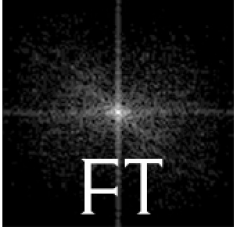
The final \mathcal{T} is then searched on the composite error array created as a sum of all particular errors. Apparently with the growing tiles number the particular total cut-planes error grows as the tiles are results of a compromise between more patches. Another approach is to find an optimal triple toroidal tile and the next tile search as minimizing

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

the error between original \mathcal{P}_i and shifted $\mathcal{P}_{r+\delta}$. Thus one optimal (primary) and some suboptimal (secondary) tiles are used.

Note that the satisfying number of patches (n_p) is usually low (see next chapter). The resulting tiles are sufficiently visually effective in the number of two, or three.

Fourier Transformation Tile Size Detection



Instead of the overlap error minimizing the Fourier transformation (illustrated by Figure 3.11 as block of proposed method scheme 3.18) can be utilized to determine an ideal \mathcal{T} size. The optimal tile size created by and dominant texture spatial frequency ensures that the patch will contain the sufficiently representative minimal information for the texture synthesis.

Hence the frequency property of a dynamic texture is the same (or very similar) thought time due to DTs temporal homogeneity, the dominant frequency is computed on first dynamic texture frame (the average from whole DT can be computed, but the test proposed to suggest is it is not necessary).

Now, let's assume color input texture t -th frame

$$\mathcal{Y}_{[r_4=t]},$$

where $[r_4 = t]$ notations is used to denote the same, but more lengthy $\mathcal{Y}_{[\bullet, \bullet, \bullet, t, \bullet]}$ notation. The frame has the size $M \times N \times S$. The S spectral planes are computed separately and the results are next combined.

Three mono-spectral texture samples \bar{Y} is subsequently transformed with the aid of discrete 2D Fourier transform from the spatial domain to its frequency domain \mathcal{F} :

$$\mathcal{F}_s(u, v) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \mathcal{Y}_{i,j,s} \exp \left\{ -2\pi i \left(\frac{ui}{M} + \frac{vj}{N} \right) \right\}. \quad (3.63)$$

Next, an amplitude spectrum $A(u, v)$ is calculated from the texture representation in the frequency domain:

$$A(u, v) = \sum_{s=1}^S A_s(u, v), \quad (3.64)$$

$$A_s(u, v) = |\mathcal{F}_s(u, v)| = \sqrt{\text{Re}^2(\mathcal{F}_s(u, v)) + \text{Im}^2(\mathcal{F}_s(u, v))}. \quad (3.65)$$

The result is then transformed to centered representation yielding in $A(u, v)$, $u \in [-\frac{M}{2}, \frac{M}{2}]$, $v \in [-\frac{N}{2}, \frac{N}{2}]$. Because an image is the real function of two real variables defined on a finite domain, amplitude spectrum $A(u, v)$ is symmetrical about the origin $A(0, 0)$. Hereafter it is of course sufficient to consider only $u \in [-\frac{M}{2}, \frac{M}{2}]$ and $v \in [0, \frac{N}{2}]$.

The dominant spatial frequency can be found as maximum from amplitude spectrum $A(u, v)$ with coefficient A_m :

$$A_m = A(f_r, f_c) = \max \left\{ A(u, v) \mid v \in \left[0, \frac{N}{2}\right], u \in \left[-\frac{M}{2}, \frac{M}{2}\right] \mid |u| > 1 \vee v > 1 \right\}. \quad (3.66)$$

Of course the central strongest frequency ($A(0, 0)$) are excluded from the process. Coefficient $A(0, 0)$ is image mean of \mathcal{Y} , and it is not used. In order to create more optimal double toroidal tiles, it is necessary to detect dominant frequency which is repeated in \mathcal{Y} in both spatial directions at least twice. It follows constraints on u and v in equation (3.66). Integer indices f_r and f_c give information how many times the dominant frequency is repeated in vertical and horizontal direction, respectively. If $f_r < 0$, then it is set to $f_r = |f_r|$.

In case that $f_r = 0$ or $f_r = 1$, detected dominant frequency f_c is aligned with horizontal direction and it is necessary to determine dominant frequency in vertical direction. Frequency f_r is determined according to the largest coefficient A_m^r :

$$A_m^r = A(f_r, \cdot) = \max \left\{ A(u, v) \mid u \in \left[-\frac{M}{2}, -2\right] \cup \left[2, \frac{M}{2}\right], v \in [0, 1] \right\}. \quad (3.67)$$

In case that $f_c = 0$ or $f_c = 1$, detected dominant frequency f_r is aligned with vertical direction and therefore dominant frequency f_c in horizontal direction is determined according to the largest coefficient A_m^c :

$$A_{\max}^c = A(\cdot, f_c) = \max \left\{ A(u, v) \mid u \in [-1, 1], v \in \left[2, \frac{N}{2}\right] \right\}. \quad (3.68)$$

Estimated parameters f_r and f_c are used for the following precise determination of double toroid-shaped tile size.

3.4 Triple Toroid-Shaped Tile

The dynamic double toroid-shaped tile can be used to synthesize infinitely large output in spatial dimensions only. The temporal transition which can create the toroidal cut plane in the already founded dynamic double toroid-shaped patch allows to seamlessly repeat the patch and create an infinitely large sequence in the temporal meaning.

The resulting tile \mathcal{T} that consists of six spatiotemporal cut planes is called a Triple Toroidal-Tile. Firstly let's suppose only one temporal transition is used.

Similar to spatial overlap the temporal overlap has to be computed to determine the area in which the temporal cut can be searched for. This can be easily done as computing the all-to-all frames distance. This well-known principle is used e.g. in [113] and many other approaches with distance ϵ as defined in eq. (3.43):

$$\Gamma^* = \min_{t,u} \left(\sum_{\forall x,y} \epsilon(\mathcal{Y}_{x,y,\bullet,t,\bullet}, \mathcal{Y}_{x,y,\bullet,u,\bullet}) \right), x \in \mathcal{P}, y \in \mathcal{P}, |t - u| > l_t, \quad (3.69)$$

where t and u are unknown temporal coordinate of the first of most similar frames, the l_t is a minimal distance between the searched frames. The l_t of course depends on the source data to keep the consistent frequency property the l_t is set as:

$$l_t = \min([\mathcal{I}^V]_v, [\mathcal{I}^H]_h). \quad (3.70)$$

The temporal overlap is then defined in the same manner as spatial overlaps:

$$I_{to} = (0, \dots, M - 1) \times (0, \dots, N - 1) \times (0, \dots, T - |u - t|). \quad (3.71)$$

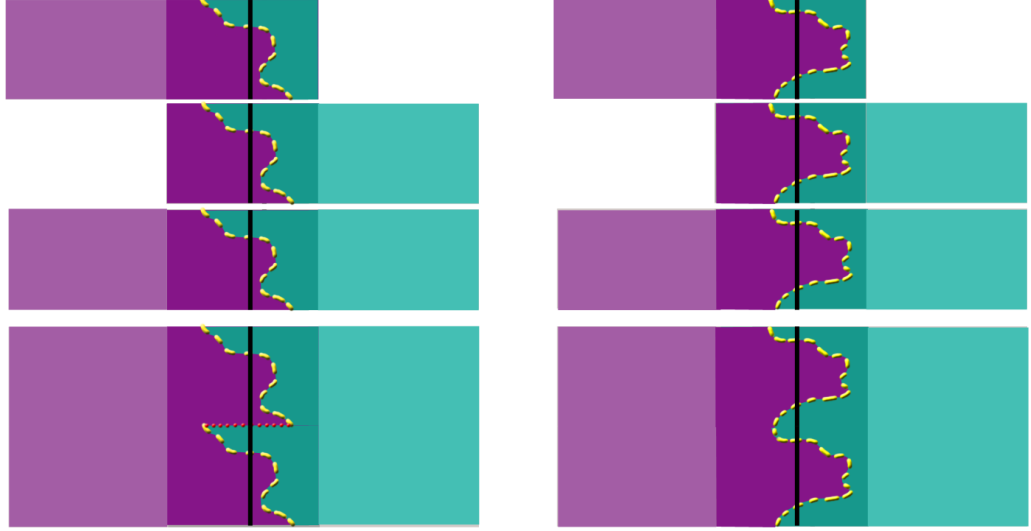


Figure 3.13: *Temporal cuts*: Illustration of temporal cuts without (left) and with spatial toroidal property. Input texture (twice), temporally enlarged and spatiotemporally enlarged texture shown in top to down order.

3.4.1 Temporal Cut

The creating of the temporal cut which seamlessly loop the tile can be done e.g. by the graph-cut algorithm[73] or by a frame-to-frame loop[113] with using of similarity matrix(Figure 3.12) where every $x - y$ location is representation of x and y frames similarity.

The computing of triple toroid-shaped tiles from the dynamic double toroid-shaped tile is not trivial. Toroidal properties of the sample must be maintained. On the Figure 3.13 this problem is illustrated: The temporal and *any* spatial dimensions are represented by the horizontal and vertical dimension respectively. First, two rows show *the same* texture with the cut in it; black line denotes the most similar patch frame u and t founded by eq. (3.70). The second row show textures synthesized in the temporal dimension. The third row presents synthesis in temporal and spatial dimensions. The red dotted line shows the location where the spatial continuity is violated. It can be easily seen, that if the temporal cut is not double toroidal itself the border violates the spatial homogeneity.

The cost of whole (permissible) temporal cut is minimized (see 3.69) so as to preserve the spatial properties of the toroid-shaped patch (maximal change of t between adjacent values of $I_{x,y,t}$ and $I_{x+i,y+j,t}$ is c_t)

$$\Psi^{T*} = \min_{\hat{C}^{T+}} \left(\sum_{i=0}^N \tilde{\varepsilon}^{Tx} \left(\hat{C}_{[r_2=i]}^{Tx+} \right) \right), \quad (3.72)$$

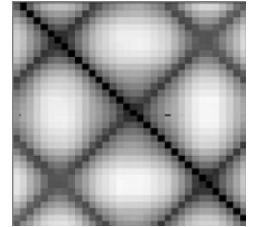


Figure 3.12: *Similarity matrix*.

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL



Figure 3.14: *Set-of-cuts illustration*: First three iteration steps of set-of-cuts estimation illustrated. Every row represents one iteration of fining the transition plane; each column represent different frames : $t - 2, t - 1, t, t + 1, t + 2$ from first to fifth respectively; every pictures is one frame of the texture where colors represents shifted and original data; first row - one cut that put together two most similar frames; second row - in each halfplane of next and previous frame another fining cut is found (and the best is used); third row - final result of additional iterations present ability to choose to change slope.

or similarly as:

$$\Psi^{T*} = \min_{\hat{C}^{T+}} \left(\sum_{j=0}^N \hat{C}_{[r_1=j]}^{Ty+} \right), \quad (3.73)$$

where \hat{C}^{Ty+} and \hat{C}^{Tx+} represent horizontal and vertical cut lines and $\tilde{\varepsilon}^{Tx}$ their appropriate total error. \hat{C}^{T+} then denotes whole temporal cut plane. Now lets look to ways how to find an optimal \hat{C}^{T+} :

Set-of-cuts

It is oblivious that the most similar temporal cut consists from the simple plane (i.e. double toroid-shaped tile). This solution is clearly not satisfying as the dynamic texture usually does not consist sets of the same frames. The set-of-cuts method is based on the similarity of not only one-pixel value but whole frame areas. However, rather than finding a single transition time for the whole image one way to determine the temporal cut is to find whole similar areas for the transition.

The overlaying frames are iteratively cut with minimal error cutlines to two halves (See Figure 3.14). Then the transition error of both halves is evaluated and compared. The final frame then consists of one half from frame t and another from frame $t + \lfloor I^T \rfloor_t$.

Next cuts are computed iteratively on both planes (second row in Figure 3.14 for the second iteration and so on.). The cutlines are computed until the areas consist of at least N pixels.

Of course, this approach can lead to the greedy-like algorithm. Optimal set-of-cuts can be easily found by not determining the next state but recursive applying the filling rules. Every cutline then create two possible states consisting from two halfplanes A and B which are filled by a frames t and $t + \lfloor I^T \rfloor$ or $t + \lfloor I^T \rfloor$ and t respectively. It is easy to see that the number of possible states is limited to $2^{2^{i-1}}$ for $i > 0$ where i is an iteration. Fortunately, the method reaches final state very quickly in less than five iterations. The all possible set-of-cuts (4-tree of cuts) is then compared by their transition error.

This result is very similar to determining the time transition per-pixel basis, but local spatial properties of the texture are kept. By this, the infinitely long sequence of a texture can be found, and because patch satisfies the global texture dynamic, this simple approach can work with it. Every spatial cut line \hat{C}^{t+} are computed on the appropriate error arrays composed of error between two overlaid frames. The error ϵ^t for every pixel is defined as eq. (3.43). The error array is computed exactly like in the dynamic toroidal shape tile as ϵ^{h+} (3.38) and ϵ^{v+} (3.39) as the set-of-cut approach is focused to spatial areas. The set of cuts then consist from set of C_{h0}^* computed on the given areas.

The first cut is computed on the error array as eq. (3.74):

$$\hat{C}_h^* = \arg \min_{C_h^{\{s\}}} \tilde{\epsilon}^H \left(C_h^{\{s^*\}} \right) , \quad (3.74)$$

$$(3.75)$$

and two adjacent cuts on the upperhalfplane and bottom halfplane:

$$C_{h+}^* = \arg \min_{C_h^{\{s\}}} \tilde{\epsilon}^H \left(C_h^{\{s^*\}} \right) , \quad (3.76)$$

$$C_{h-}^* = \arg \min_{C_h^{\{s\}}} \tilde{\epsilon}^H \left(C_h^{\{s^*\}} \right) , \quad (3.77)$$

Then the set-of-cuts is searched by minimizing all particular errors:

$$\begin{aligned} \tilde{\epsilon} \left(\hat{C}_h^* \right) &= \tilde{\epsilon} \left(\hat{C}_{h0}^{\{I_{[C_T, C_B]}\}} \right) \\ &+ \min \left(\tilde{\epsilon} \left(\hat{C}_{h+}^{\{I_{[C_T, C_{h0}]}\}^+} \right), \tilde{\epsilon} \left(\hat{C}_{h+}^{\{I_{[C_T, C_{h0}]}\}^-} \right) \right) \\ &+ \min \left(\tilde{\epsilon} \left(\hat{C}_{h-}^{\{I_{[C_{h0}, C_B]}\}^+} \right), \tilde{\epsilon} \left(\hat{C}_{h-}^{\{I_{[C_{h0}, C_B]}\}^-} \right) \right) , \end{aligned} \quad (3.78)$$

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

$$\begin{aligned}
\tilde{\varepsilon} \left(\hat{C}_{h^+}^{\{I_{[C_t, C_m]}\}} \right) &= \tilde{\varepsilon} \left(\hat{C}_{h^0}^{\{I_{[C_t, C_m]}\}} \right) \\
&+ \min \left(\tilde{\varepsilon} \left(\hat{C}_{h^+}^{\{I_{[C_t, C_{h^0}]}\}^{t+}} \right), \tilde{\varepsilon} \left(\hat{C}_{h^+}^{\{I_{[C_t, C_{h^0}]}\}^{t-}} \right) \right) \\
&+ \min \left(\tilde{\varepsilon} \left(\hat{C}_{h^-}^{\{I_{[C_{h^0}, C_m]}\}^{t+}} \right), \tilde{\varepsilon} \left(\hat{C}_{h^-}^{\{I_{[C_{h^0}, C_m]}\}^{t-}} \right) \right) ,
\end{aligned} \tag{3.79}$$

$$\begin{aligned}
\tilde{\varepsilon}_v \left(\hat{C}_{h^-}^{\{I_{[C_m, C_b]}\}} \right) &= \tilde{\varepsilon} \left(\hat{C}_{h^0}^{\{I_{[C_m, C_b]}\}} \right) \\
&+ \min \left(\tilde{\varepsilon} \left(\hat{C}_{h^+}^{\{I_{[C_m, C_{h^0}]}\}^{t+}} \right), \tilde{\varepsilon} \left(\hat{C}_{h^+}^{\{I_{[C_m, C_{h^0}]}\}^{t-}} \right) \right) \\
&+ \min \left(\tilde{\varepsilon} \left(\hat{C}_{h^-}^{\{I_{[C_{h^0}, C_b]}\}^{t+}} \right), \tilde{\varepsilon} \left(\hat{C}_{h^-}^{\{I_{[C_{h^0}, C_b]}\}^{t-}} \right) \right) ,
\end{aligned} \tag{3.80}$$

where every cut line is found as its own subtree minimization eq. (3.74). The index set $I_{[C_a, C_b]}^t$ denotes area between two cuts C_a and C_b in the frame t . The $t+$ denotes next frame and $t-$ previous frame. The \hat{C}_v^* is defined in the similar manner.

The problem with this approach can be viewed on Figure 3.13. This can be easily overcome by enforcing the start and end coordinates (in temporal dimension) to be the same similarly as in the double toroidal tile. Moreover, the highest and the lowest areas must be continuous (so it must consist of the same frame ± 1). It can be resolved by creating a tree of all potential cuts and then traversing them to minimize this error. This approach is effective in many textures, and because of every cut is created by the single way in one frame it can fit the objects in the texture very well.

The double-tree structure for ${}^T\hat{C}_h^*$ can be defined as:

$$\begin{aligned}
{}^T\hat{C}_h^* &= \hat{C}_{h^0}^{\{I_{[C_T, C_B]}\}} \\
\cup L1 &= \hat{C}_{h^+}^{\{I_{[C_T, C_{h^0}]}\}^{t+}}, R1 = \hat{C}_{h^+}^{\{I_{[C_T, C_{h^0}]}\}^{t-}} \\
\cup L2 &= \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_B]}\}^{t+}}, R2 = \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_B]}\}^{t-}} ,
\end{aligned} \tag{3.81}$$

$$\begin{aligned}
\hat{C}_{h^+}^{\{I_{[C_t, C_m]}\}} &= \hat{C}_{h^0}^{\{I_{[C_t, C_m]}\}} \\
\cup L1 &= \hat{C}_{h^+}^{\{I_{[C_t, C_{h^0}]}\}^{t+}}, R1 = \hat{C}_{h^+}^{\{I_{[C_t, C_{h^0}]}\}^{t-}} \\
\cup L2 &= \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_m]}\}^{t+}}, R2 = \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_m]}\}^{t-}} ,
\end{aligned} \tag{3.82}$$

$$\begin{aligned}
\hat{C}_{h^-}^{\{I_{[C_m, C_b]}\}} &= \hat{C}_{h^0}^{\{I_{[C_m, C_b]}\}} \\
\cup L1 &= \hat{C}_{h^+}^{\{I_{[C_m, C_{h^0}]}\}^{t+}}, R1 = \hat{C}_{h^+}^{\{I_{[C_m, C_{h^0}]}\}^{t-}} \\
\cup L2 &= \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_b]}\}^{t+}}, R2 = \hat{C}_{h^-}^{\{I_{[C_{h^0}, C_b]}\}^{t-}} ,
\end{aligned} \tag{3.83}$$



Figure 3.15: *Temporal cut plane*: Each picture represents one frame in texture from left to right and top to down order; all pictures is one frame of texture where colors represent offset and original data; the general shape of the plane and its double toroidal property can be easily seen.

where every cut line \hat{C} has up to four (L1, R1, L2, R2) descend from next iteration. And appropriate \hat{C}_h^* is created by filling the areas between upper and bottom plane with corresponding t values. The errors and cut lines structure (and thus cut plane) for the vertically oriented set is defined in the exact same manner.

Note that apparently the two optimal set-of-cuts can be found - ones with dominant horizontal dimension (like Figure 3.14 and ones with dominant vertical dimension. Obviously, both version can be computed with the one with lowest transition error can be picked.

Interestingly this the set-of-cuts approach is well suited for dynamic textures with very low dynamics (i.e. mostly static scene) and for dynamic textures with extremely high dynamics like (e.g. FIRE texture).

3.4.2 Fitting plane

In some cases, the previous approaches for the temporal cut are not satisfying. Every single cut is computed in one frame and derived by *spatial* data, but in fact, it drives the *temporal* offset. If the temporal dynamic of the texture is significantly greater than the local spatial dynamic, the temporal cut plane can be computed differently. Like in set-of-cuts the problem is to create a cut with no boundary spatial errors. This can be done by using a similar procedure like in propagating the double toroidal tile though temporal dimension. This process depends on the direction (top-down or left-right), but in most cases, the results are nearly the same. The finding the plane by the second way is better to the texture with lesser local spatial dynamics (e.g. leaves or grass in string wind, flowers or leaves). As can be seen in Figure 3.15 the planes generated by this approach is more general, but satisfy the toroidal boundary rules and are double toroidal like a spatial patch (this can be easily done to setting error cost of ending unwanted coordinates to infinity

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

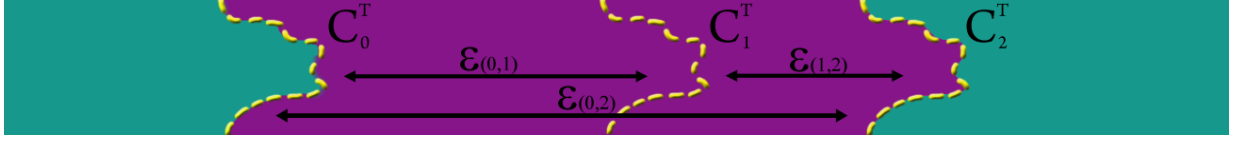


Figure 3.16: *More temporal jumps*: Illustration of a triple toroidal patch (violet) with two temporal cut planes in the given DT (whole image). The vertical image dimension illustrates any spatial dimension; horizontal dimension denotes time dimension. The yellow dotted line is time cut-plane illustration of cut plane C_0^T , C_1^T and C_2^T . The $\epsilon_{(0,1)}$, $\epsilon_{(1,2)}$ and $\epsilon_{(0,2)}$ illustrates the possibly jumps and its Ψ^{T*} error.

before computing the structure).

The three dimensional error array are defined in the exact same manner as eqs. (3.62) and (3.61):

$$\hat{\psi}_r^{Th*} = \epsilon_r^{th+} + \min(\hat{\psi}_{r+[0,-1,\bullet,\bullet,-p_t]}^{tv*}, \dots, \hat{\psi}_{r+[0,-1,\bullet,\bullet,0]}^{tv*}, \dots, \hat{\psi}_{r+[0,-1,\bullet,\bullet,p_t]}^{tv*}) , \quad (3.84)$$

$$\hat{\psi}_r^{Tv*} = \epsilon_r^{tv+} + \min(\hat{\psi}_{r+[-1,0,\bullet,\bullet,-p_t]}^{th*}, \dots, \hat{\psi}_{r+[-1,0,\bullet,\bullet,0]}^{th*}, \dots, \hat{\psi}_{r+[-1,0,\bullet,\bullet,p_t]}^{th*}) , \quad (3.85)$$

where the temporal-horizontal and temporal-vertical error arrays are defined as:

$$\hat{\psi}_r^{th*} = \epsilon_r^{h+} + \min(\hat{\psi}_{r+[-1,0,\bullet,\bullet,-p_t]}^{v*}, \dots, \hat{\psi}_{r+[-1,0,\bullet,\bullet,0]}^{v*}, \dots, \hat{\psi}_{r+[-1,0,\bullet,\bullet,p_t]}^{v*}) , \quad (3.86)$$

$$\hat{\psi}_r^{tv*} = \epsilon_r^{v+} + \min(\hat{\psi}_{r+[0,-1,\bullet,\bullet,-p_t]}^{h*}, \dots, \hat{\psi}_{r+[0,-1,\bullet,\bullet,0]}^{h*}, \dots, \hat{\psi}_{r+[0,-1,\bullet,\bullet,p_t]}^{h*}) . \quad (3.87)$$

The desired \hat{C}^T plane can be found by iterative spatiotemporal cut search (Sec. 3.3.2) and/or restrictive spatiotemporal cut search (Sec. 3.3.2).

More temporal jumps

Similar problem to the visually repetitive pattern which is solved by using patches can occur in the temporal dimension. The short sequence of a dynamic texture can be easily recognised and degrade the visual impression. As in the spatial dimension, even in the temporal dimension, such an impression is undesirable.

The approach that can solve this is simple: to find more than one temporal cut-plane in the dynamic double animated toroidal-shape (see Figure 3.16). Another temporal cut-planes are found inside the founded triple toroidal tile by the similar process as the first one and similarly a case of finding more suboptimal tiles. Note that in the case of more temporal cut-planes only primary-secondary approach is used. The cut planes founded by fitting to more (at least three total transition error - Figure 3.16) are not visually satisfying. For the determining of more temporal jumps two approaches could be utilized: The every next cut-plane could be founded by minimizing of the transition error to the already existing cuts (see Figure 3.16). By this way only the same shaped cut-planes shifted temporally can be found. The second approach is to step

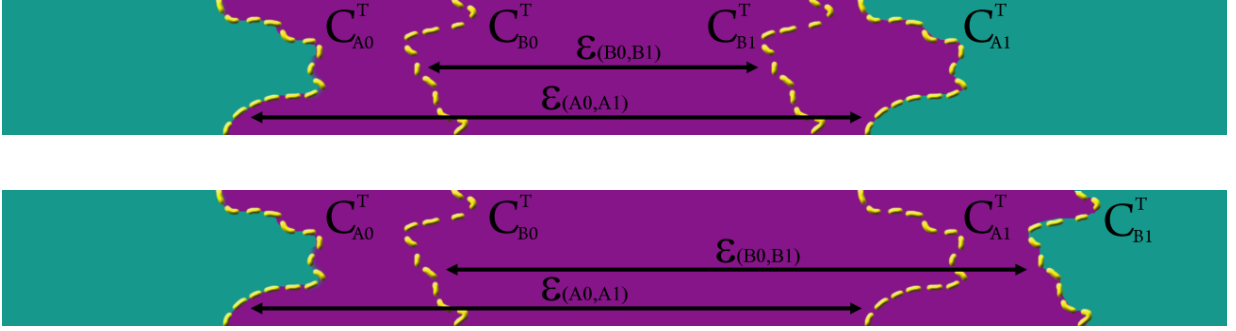


Figure 3.17: *More temporal jumps, two possible arrangements*: Illustration of a triple toroidal patch (violet) with two sets of a temporal cut planes in the given DT (whole image). The vertical image dimension illustrates any spatial dimension, horizontal dimension denotes time dimension. The yellow dotted line is time cut-plane illustrate cut-planes C_{A0}^T , C_{B0}^T , C_{A1}^T and C_{B1}^T . The $\mathcal{E}_{(A0,A1)}$ and $\mathcal{E}_{(B0,B1)}$ illustrates the possibly temporal jump and its Ψ^{T*} error.

back to similarity metric and found the second similar frames and compute whole cut-planes again in the different time overlap. Note that the triple toroidal patch should be extended this way (see Figure 3.17) if one (or both) cut-planes are found outside already founded triple toroid-shaped tile but inside the dynamic animated double toroid-shaped tile (illustrated by the whole rectangular area in both figures).

Note that the defined error distance based on color and dynamics solves the well-known problems with periodical movements (e.q. pendulum) in which the image structure can be identical, but the dynamics differ.

3. TOROID-SHAPED PATCH BASED DYNAMIC TEXTURE MODEL

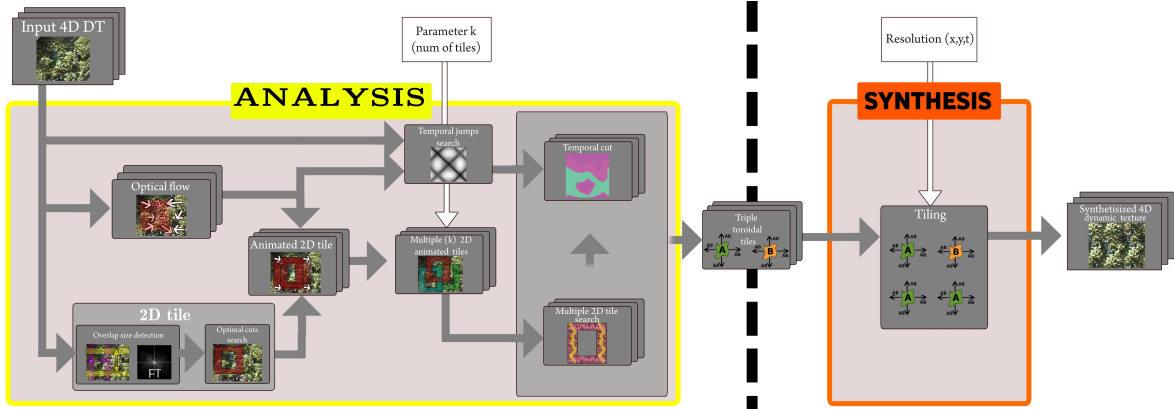


Figure 3.18: *DT synthesis overall flowchart*: The method is as follows: the double spatial toroidal tile is computed in the *2D tile* block parallel to *optical flow* for whole \mathcal{V} . The *animated 2D tile* \mathcal{T} is then created with possibility of extending to *multiple 2D animated tiles*. The frame similarity matrix is computed in the *Temporal Jump search* block. The analysis is finished by exact dynamic double toroidal tile(s) search in block *Multiple 2D tile search* and *Temporal cut*. After that the patches can be stored and then in fully separate synthesis part any *arrangement* can be generated.

3.5 Synthesis

The synthesis part is entirely separated from the analytical part and thus is extremely fast. Texture analysis part, which is inspired by the idea of optimized but straightforward repetition of the input texture sample, consists in the determination of parameters δ_h and δ_v controlling the overlaps and subsequent minimal error boundary cut based edge optimisation. The output of the analytical part is one or more triple toroidal patch and hence the toroidal shape they can be easily put abreast without any additional computation.

As follows from the analytical part an idea, the synthesis of any texture of required size is an efficient repetition of the created triple toroid-shaped texture patch in all directions until the texture of any required size is generated.

The synthesis of any size textures is merely the repetition of the created triple toroidal patches in all dimensions until the desired size is fulfilled. The only computation in synthesis is to generate once a random sequence of patches labels and next computing the offset of pixels, which can be implemented in real time just from to the actual to the next frame.

The generated lattice of \mathcal{P} labels are called arrangement and can be user-driven or randomly generated. The process of generating arrangement is then called tiling. The overall method flowchart is in the Figure 3.18.

Dynamic Texture Editing

This chapter describes a set of specific synthesis algorithm enhancements that serve to increase the limited number of results and, above all, to consider their visual quality of creating mix-of-DTs (Figure 4.1). The pure texture synthesis methods can produce results with visually recognizable artifacts and patterns as recapitulated in the introductions section 4.1 followed with an overview of presented editing methods which simultaneously allows to spatially and temporarily enlarge the original dynamic texture(s) can be found in the section 4.2. Finally, the mix-of-textures principles and transition textures and temporal shift utilizing are described in the section 4.3.



Figure 4.1: *Edited dynamic texture teaser: Examples of results.*

4.1 Dynamic Texture Editing Overview

Natural dynamic texture modelling is a very challenging and difficult task, due to an unlimited variety of possible materials, illumination, and viewing conditions, simultaneously complicated by the strong discriminative functionality of the human visual system. Visual texture modelling is the critical part of any computer-based visualization application because whatever size is the measured texture, it is always inadequate and requires its enlargement to cover the required visualized object's surface area. The available material sample size is either too small for rendering complex and large virtual scenes. The amount of such measured data similarly to dynamic textures is immense, e.g., in the range of terabytes even for spatiotemporally restricted measurements. Moreover, the textural enlargement may not be sufficient to cover some complex scene. Texture editing enables to create photo-realistic synthetic dynamic textures, which are either difficult (or unable) to measure or which even do not exist in nature. The editing principles allow synthesizing result composed from \mathcal{P} from more than only one input dynamic texture \mathcal{Y} . The carefully selected \mathcal{P} with one consistent \mathcal{T} or even more different shapes \mathcal{T} are then put together to create a desired result.

The resulting edited dynamic texture is, therefore, a mixture of several color dynamic textures that realistically match the given color textures appearance and respects their original dynamics. The presented method simultaneously allows to spatially and temporarily enlarge the original dynamic textures to fill any required four-dimensional dynamic texture space.

Dynamic texture editing and its results allow to reach huge compression ratio if numerous DTs can be constructed from several bases small DT samples, or to study the relationship between model parameters and their impact on visual DT appearance, etc. The common visual problem repetition in the spatial and temporal dimension is suppressed by not only using more samples and their random placement in the synthesized texture but with temporal deformation of the texture dynamics too.

The realistic appearance of the dynamic textures mix requires to edit the patch color space and to find transition patches which consist of more than one *type of the texture*. These border patches are found similarly to the multi-texture analysis patch step. Again, all time-consuming processing such as the finding of optimal spatio-temporal triple toroidal patches, are done in the analytical step only. The analytical and synthesis step are still separated. Therefore the synthesis of the edited and enlarged texture can be done very efficiently by applying the simple set of repeating rules to fill the three-dimensional arrangement lattice of \mathcal{P} labels.

4.2 Method Overview

The presented editing method is based on using several triple toroid-shaped tiles (Sec. 3.4). Moreover, with patches from more than one DT types, the number of possible outputs is significantly higher. Few triple toroid-shaped tiles can create numerous noticeably larger synthesized DTs with similar stochastic properties. By using of toroidal-shaped tile and fitting them by optical flow, the synthesized texture can be recreated from various input textures to generate many arrangements of output textures or even mix-of-textures.

The repetition of the spatial and temporal dimension is suppressed both by using overlapping samples and their random placement in the synthesized texture and temporal deformation of the texture dynamics. The method can create a large scale of dynamic texture types and create results with high-quality, visually satisfying realistic appearance.

Editing method Presented approach allows DT editing in several ways. The system of toroidal samples allows to apply the textural operators to tone colors, change dynamics or differs texture itself without the possibility to affect the homogeneity of the whole dynamic texture. For example to change the dynamics of one object in the dynamic texture usually requires area segmentation which must be consistent in all frames, detection of a given object and finally changing its dynamics.

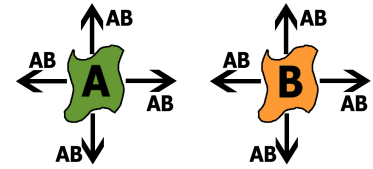


Figure 4.2: Data patches **A** and **B** with the same tile shape.

Primary and secondary texture

Hence the editing methods take more than one input dynamic textures to distinguish them one of them is called primary, and rest are called secondary. The primary dynamic texture is used to compute (multiple) optimal triple toroidal tile while other textures only serve as a patch source for the same shaped-tile textural data (Figure 4.2). Whereas the quality of a tile decrease with the number of patches. Using more dynamic textures for finding the optimal global tile (tile optimal for more than one DT) has the similar effect. Even if in some cases the optimal global tile can be found, usually the primary-secondary approach has more satisfying results.

Presented approach obviously allows to skip the segmentation step since the texture is divided into logically consistent patches (given by spatial similarity and similar dynamics) from previous analysis steps so the changes made in one patch will not affect other areas. Elementary, but not trivial type of texture editing is to synthesize texture consisting of multiple input textures. The illustrative example of a resulting arrangement can be seen in figure 4.3. The result is created from \mathcal{P} from more input dynamic textures (different \mathcal{P} colors in Figure 4.3) with the same \mathcal{P} shape. The resulting texture is, therefore, a mix of textures, which can vary in two of the three fundamental properties of DT - the texture

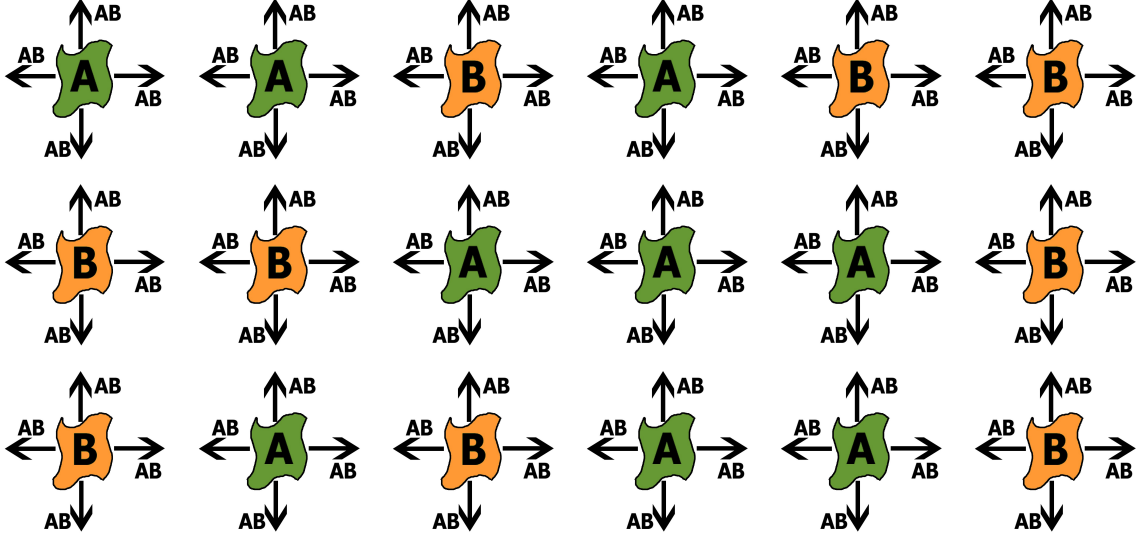


Figure 4.3: *Simple mix-of-DTs arrangement*: Random arrangement from two patches type from two input dynamic textures.

itself (this includes changes in lighting, but also the other conditions - detail, different viewing angle, etc.) and local dynamics (intensity and direction of wind, rain, etc.).

Of course, the resulting combination of given inputs should be meaningful. For example, the combination of DT types WATER and CLOUDS textures can satisfy some trivial texture similarity, but of course are preposterous in most cases (except horizon and similar arrangement). The example of a mix between RIVER and SHRUBS which can be on first look correctly synthesized is in the Figure 4.4. Therefore, the rest of the text assumes that this requirement is met.

More DT types tiling

Lets look at the possible amount of synthesized results. It is obvious that the maximal number of possibly synthesized textures is:

$$(N_{in_tex} \cdot N_{p_per_tex})^{(1 + \lceil \frac{|\mathcal{S}|_v}{\lceil \mathcal{T} \rceil_v} \rceil)} \cdot (\lceil \frac{|\mathcal{S}|_h}{\lceil \mathcal{T} \rceil_h} \rceil + 1) , \quad (4.1)$$

if we assume a constant distribution of patches in input textures. N_{in_tex} is number of input textures, $N_{p_per_tex}$ is average numbers of patches per input texture. The \mathcal{S} denotes the resulted synthesized dynamic texture. Therefore $|\mathcal{S}|_w$ and $|\mathcal{S}|_h$ are the vertical and horizontal size of the synthesized texture respectively, and $\lceil \mathcal{T} \rceil_h$ and $\lceil \mathcal{T} \rceil_v$ are the horizontal and vertical size of the used optimal \mathcal{T} , respectively.

More generally and because we often use the different number of patches in every texture (to ensure the better visual appearance of synthesized textures) the (4.1) can be simplified to:

$$(N_{patches})^{(1 + \lceil \frac{|\mathcal{S}|_h}{\lceil \mathcal{T} \rceil_h} \rceil)} \cdot (\lceil \frac{|\mathcal{S}|_v}{\lceil \mathcal{T} \rceil_v} \rceil + 1) , \quad (4.2)$$

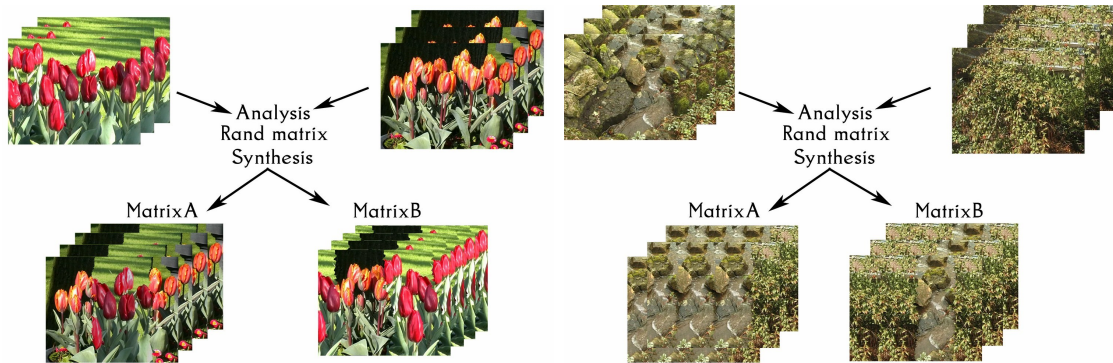


Figure 4.4: *Two DT types combining*: Example of problem in combining two very different types of dynamic textures. Right: DT RIVER and SHRUB with different dynamics, Left - DTs LIGHT TULIPS and DARK TULIPS.

where first product in (4.1) becomes to $N_{patches}$ as the number of all patches found by analysis part of the algorithm.

The synthesis of the mix-of-DTs consists just from finding the same triple toroid-shaped tiles or - more accurately - by finding one (or more) optimal triple toroid-shaped tiles in one primary DT and finding the minimal cost of placing this shape to other textures by finding the optimal time-spatial shift δ for the optimal tile. If there are many input textures the computational time can be reduced by using texture pyramid, testing only local optical flow, or testing global optical flow first (respectively in backward order).

Colortone Using several input textures can cause the problem (see Figure 4.4) with slightly different color tone or illumination of particular input DTs. This problem can be overcome, i.e., by tone mapping [43, 32, 110], by tone operators[38] and many others.

The color tone of patches can be fitted to the color tone of the selected input texture or a consensus of all textures. There are many advanced algorithms and approached for color tone mapping problem [129] which can be used for mapping color tone property of one texture to another. Again, proposed solution enables to mapping color property of only used patches and not the whole texture. Here straightforward but satisfying approach - fitting average H , S , L values of one texture to another by per pixel multiplication. This simple approach creates good results in the most of cases (see Figure 4.5) of synthesized DTs, but any advance method, i.e. Durand[32] method is of course possible. Of course, this straightforward approach works well primary on the DT with the same type and DT that are visually similar. The advanced tone reproducing operators like Ward[139] or Tumblin[130]are required (see Durand[32] for comprehensive overview).

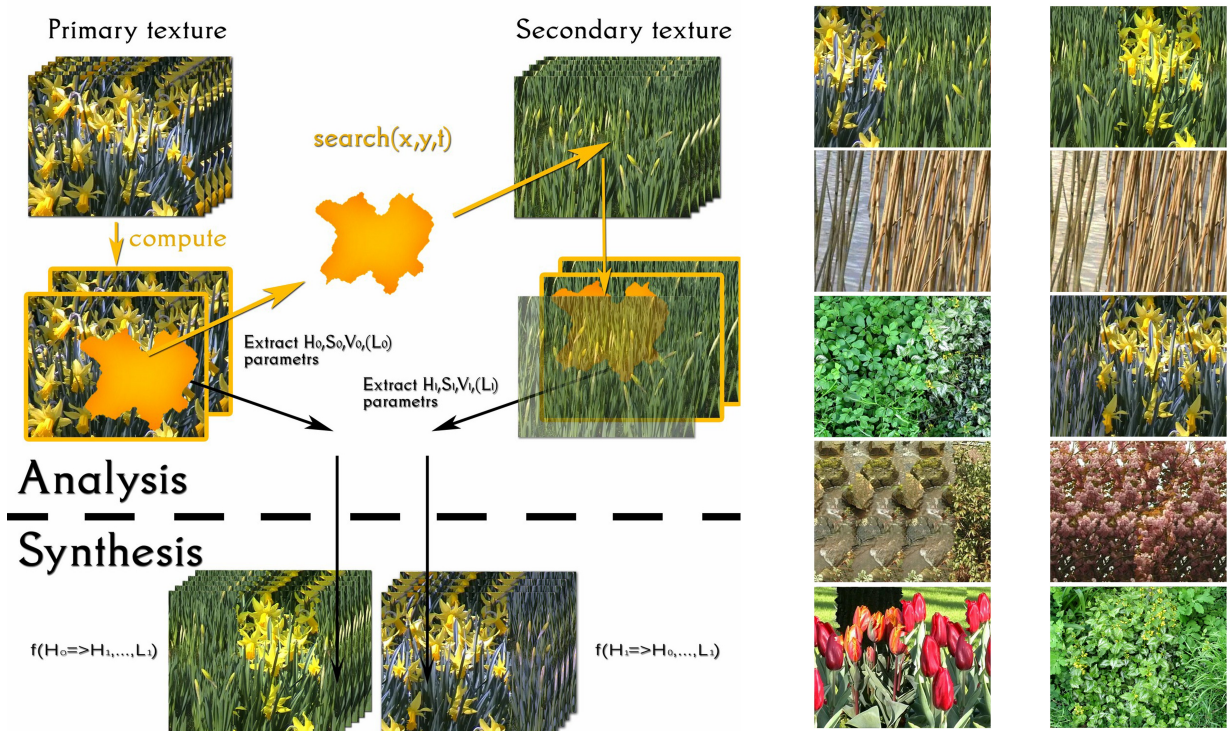


Figure 4.5: *Direct mix-of-DTs with color toning*: Illustration of the two color toning results for synthesized textures consisting of two DTs patches (left). Examples of synthesized results without color toning are in the middle column. Examples of results with color toning are on the right side.

More DT color types tiling

With varying color tones the number of possibly synthesized textures obviously then grows to:

$$N_{in_tex}^* \cdot N_{patches} \left(\left\lceil \frac{|S|_v}{|T|_v} \right\rceil + 1 \right) \cdot \left(\left\lceil \frac{|S|_h}{|T|_h} \right\rceil + 1 \right), \quad (4.3)$$

where $N_{in_tex}^*$ is a number of input textures (color tones) to which the synthesized result can be mapped. In this automatic approach, we assume that the input DTs are selected manually and are visually similar and do not differ drastically in color (i.e. grass and clouds). The similar dynamic texture structure is therefore needed.

Figure 4.5 illustrates the method. An optimal shape is computed in the primary texture and found by a spatio-temporal shift in the secondary texture. Colortone operators are extracted from all input textures. In the synthesis step, the color tone operators of textures are fitted to one input texture to create a similar color appearance.

This trivial approach can handle the camera automatic color balance (as in the IRISES), different illumination or similar problems and increase the credibility of the result. Of course, as the color tone operator are vast and popular research area the developing of an

advanced dynamic texture color tone operators is beyond the scope of this work and is part of one of the possible follow-up of this work.

4.3 Mix of Dynamic Textures

Even if the triple dynamic toroid-shaped tile approach can successfully handle the mix-of-DT (see Figure 4.5) only by trivial color tone matching, in many cases it is insufficient. Sometimes dynamic textures strongly vary in both structure and dynamics, and patches that satisfy the similarity cannot be found.

The solution to this problem is based on using *transition textures*. We assume the existence of the transition (border) between types of DT but also the lack of one (or both) texture itself. Figure 4.6 illustrate basics method idea with subfigures consisting from two DTs (labelled non-mutually).

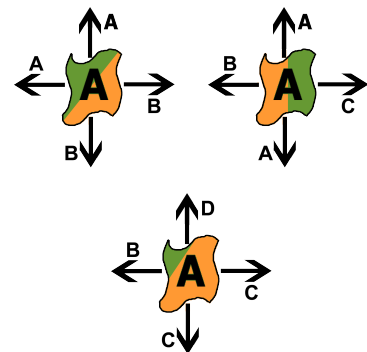


Figure 4.6: *The transition patches illustration.*

4.3.1 Transition patches

The precomputation of a transition patches (Figure 4.7, right) may be difficult to obtain, but once (and if) they are computed, the synthesis and analysis are fully separated, and thus thy creation of any arrangement is extremely quick. This is called as an offline method from now. The second approach consists of transition patch founding by the process of establishing every arrangement. Hence the transition patches are created on the flow the method is called as online from now.

The offline method is, of course, a variant of a well-known marching cubes[86] algorithm. The basics 2D examples (subspace form 16 needed cases) can be seen on the Figure 4.7. The drawback of this method is clear - the high requirements on the input data. Note that usually, it is not possible to find and precompute all possible cases even in the spatial dimensions only, but some consistent subspaces are possible.

In the transition patches approach the optimal \mathcal{T} is computed in the primary texture only and by appropriate cut-plane errors minimizing found by a spatio-temporal shift in the secondary textures. Finally, when the arrangement is filled, the δ^* to desired patch in the transition texture is searched for.

The \mathcal{P} difference is composed of difference between at least two of temporal-horizontal (top or bottom), temporal-vertical (left or right) or horizontal-vertical (front and end) cut-

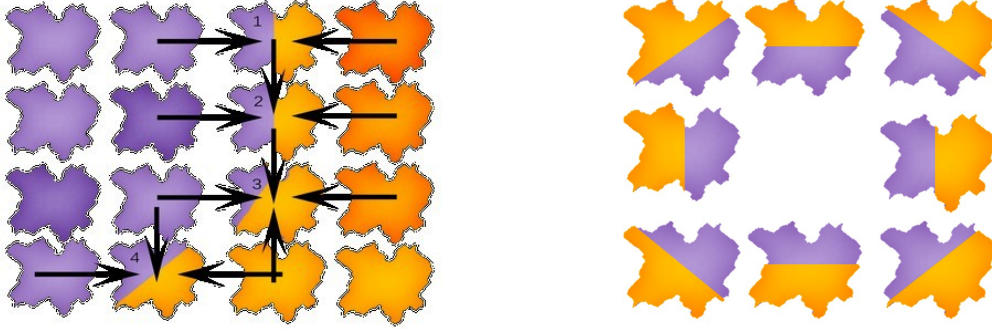


Figure 4.7: *Transition texture patches illustration*: A scheme of a transition texture for 4-neighbourhood is illustrated on the left. Transition texture patches are computed by top-down order. New transition patches are computed from surroundings patches, even transition texture patches found in the previous step. Many orders of founding patches are possible, i.e. weighted by a number of already know neighbourhood patches; Illustrative subspace of 8-neighborhood transition patches are on the right side.

planes. The appropriate shift (location of \mathcal{P}_t in the \mathcal{Y}_t) are found by error minimization:

$$\delta_r^* = \arg \min_{\delta_r} \left(\begin{aligned} & \epsilon_{th}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, 0, \mathcal{P}_{bottom}) + \epsilon_{th}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, N, \mathcal{P}_{top}) \\ & + \epsilon_{tv}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, 0, \mathcal{P}_{left}) + \epsilon_{tv}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, M, \mathcal{P}_{right}) \\ & + \epsilon_{vh}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, 0, \mathcal{P}_{front}) + \epsilon_{vh}^*(\mathcal{Y}_T, \mathcal{T}_P, \delta_r, 0, \mathcal{P}_{end}) \end{aligned} \right) \quad (4.4)$$

where parameters of every ϵ function are transition texture \mathcal{Y}_T , an optimal tile \mathcal{T}_P , desired shift δ_r , parameter δ_s that established if appropriate cut-plane (from transitional texture) is left / bottom / front or right / top / end and finally the primary or secondary texture patch \mathcal{P} ,

The error is computed with using of pixel error (3.43) between appropriate pixels from both textures. Their location is given by the \mathcal{T} which has its own location in the primary and secondary textures and desired location in the transition texture. As the empty arrangement location are filled iteratively, some location on the arrangement lattice could be empty. Therefore the difference is computed as:

$$\epsilon_{th}^*(\mathcal{Y}, \mathcal{T}, \delta_r, \delta_r, \mathcal{P}) = \begin{cases} 0 & \text{if } \mathcal{P} \text{ is empty} \quad , \\ \hat{\epsilon}_{th}(\mathcal{Y}, \mathcal{T}, \delta_r, \delta_s, \mathcal{P}) & \text{else} \quad , \end{cases} \quad (4.5)$$

or if the arrangement hole is in the straight line:

$$\epsilon_{th}^*(\mathcal{Y}, \mathcal{T}, \delta_r, \delta_r, \mathcal{P}) = \begin{cases} \hat{\epsilon}_{th}(\mathcal{Y}, \mathcal{T}^{\leftarrow}, \delta_r^{\leftarrow}, \delta_s^{\leftarrow}, \mathcal{P}^{\leftarrow}) & \text{if } \mathcal{P} \text{ is empty} \quad , \\ \hat{\epsilon}_{th}(\mathcal{Y}, \mathcal{T}, \delta_r, \delta_s, \mathcal{P},) & \text{else} \quad , \end{cases} \quad (4.6)$$

where \leftarrow denotes results from previous steps and therefore the toroidal knowing possibility of the future patch (as illustrate purple to green and yellow to purple areas in Figure 4.8).

The particular errors are computed as:

$$\hat{\epsilon}_{th}(\mathcal{P}, \mathcal{Y}, \mathcal{T}, \delta_r, \delta_s) = \sum_{i=0}^T \sum_{j=0}^N \epsilon \left(\mathcal{P}_{\left[\hat{C}_{[i,j],j,\bullet,i,\bullet}^{th} \right]}, \mathcal{Y}_{\left[\delta_{\mathcal{P}+\delta^+} + \left[\delta_s + \hat{C}_{[i,j],j,\bullet,i,\bullet}^{th} \right] \right]} \right), \quad (4.7)$$

where $\delta_{\mathcal{P}}$ is patch location (global indices for the $[0,0,0]$ of appropriate I_{th}) in the appropriate texture. \hat{C}^{th} , \hat{C}^{tv} and \hat{C}^{vh} are derived as part of \mathcal{T} . The ϵ^{tv} , $\hat{\epsilon}^{tv}$, ϵ^{vh} , and $\hat{\epsilon}^{vh}$ are computed in the exact manner with previously error (3.43).

The random DT arrangement is generated for an online method by iterative growing (erosion) of DT types on a 3D lattice from two or more arbitrary coordinates. Areas of contiguous type grow in the random direction by given defined neighbourhood until it is possible. The areas on the lattice which remain unfilled are then filled by transition patches iteratively for every particular location and adjacent patches.

For each area labeled as a transitional (labeled '??' in Fig 4.8) the best fitting patch from border texture is found by minimizing the error between this sample itself and each already completed adjacent sample (the one patch for every location or one patch for every type of surroundings DT). Patches that are computed in the second step are searched for every empty location the best-fit patch is found in the given transition textures. The $\hat{\epsilon}$ computed to different \mathcal{Y} is illustrated by violet in Figure 4.8. The yellow marked location (and thus $\hat{\epsilon}$) and green marked location denotes another cut-planes. One of them (or both if this is last empty space, none is this is first filled space) are computed in the previous step. The optimal patch is then computed by minimizing ϵ^* consisting from at least two border error cost $\hat{\epsilon}$ to different \mathcal{P} or even \mathcal{Y} . The distance is computed even for each already transition patch completed as adjacent in the previous step as can be seen on yellow and green marked areas. By this, the second analysis phase is added, but still, the analysis and synthesis step are separated.

Although such type of the source data is difficult to obtain, it is visually sufficient to have even a small data sample. The dominant component that determines the quality of the appearance of the synthesized result tends to be both significant pure DTs.

Mix of textures with border tiling

Again, lets look at the amount of possibly existing results. In transition texture based approach, the number of possibly synthesized textures is limited by:

$$N_{res} = \prod_{i=1}^{\mathcal{Y}^N} N_{res} + N_t, \quad (4.8)$$

where N_t denotes the possible number of transition texture combinations which is of

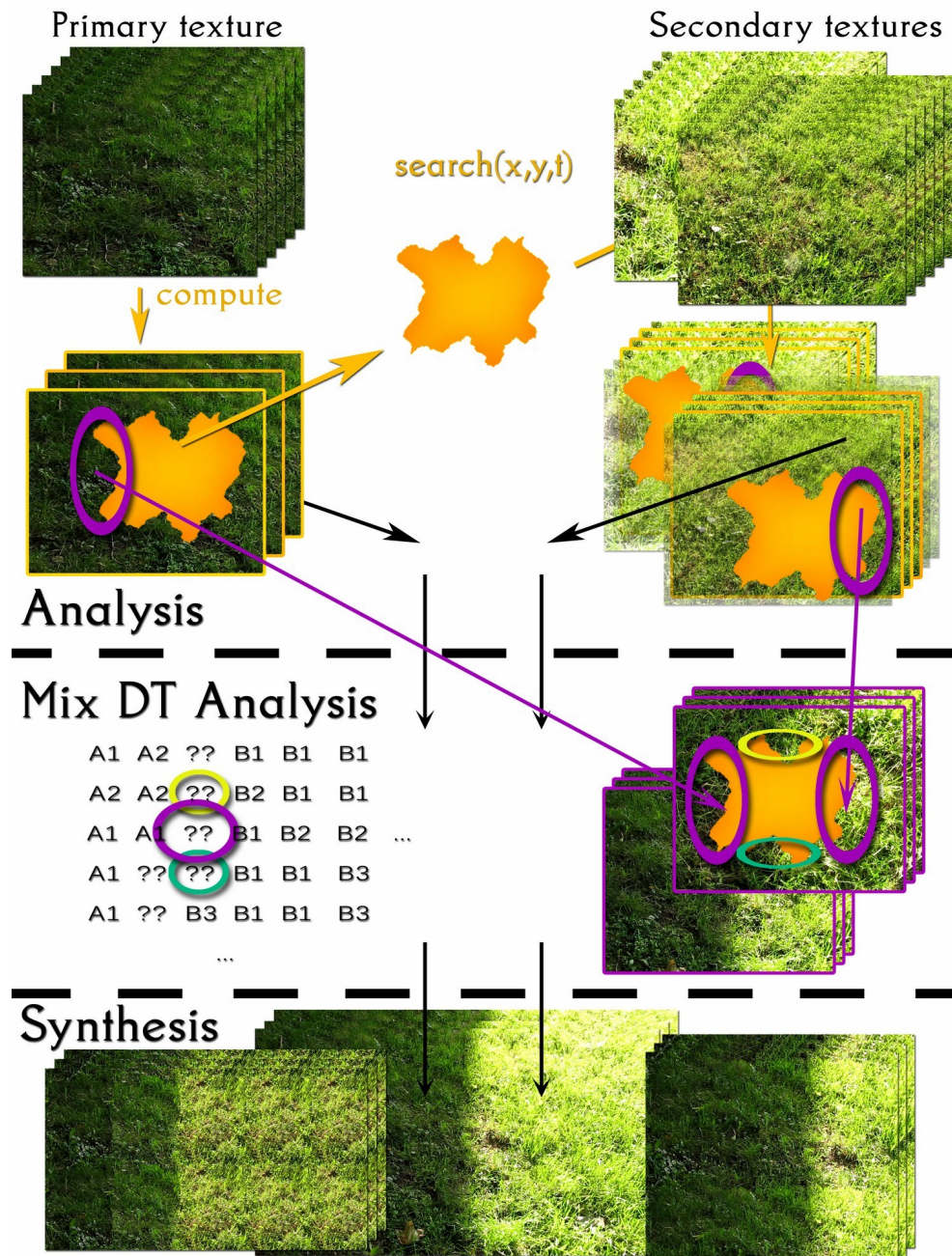


Figure 4.8: *Transition patches principle scheme*: Illustration of the mix of dynamic textures with transition patches.

course limited by theoretically all combinations on the maximal side and using of only one texture (transition is a line) on the minimal side:

$$N_t \in \langle 1, \dots, \mathcal{P}_t! \rangle . \quad (4.9)$$

The ${}^i N_{res}$ then denotes all possible result combinations from every DT type area i :

$${}^i N_{res} = [{}^i \mathcal{S}]^{i \mathcal{P}_{num}} \cdot {}^i \mathcal{Y}_{ct} , \quad (4.10)$$

where $[{}^i \mathcal{S}]$ is size of the appropriate i -th area and ${}^i \mathcal{Y}_{ct}$ possibly color-tones variations.

Many orders in which the right transition patch is, of course, possible, e.g., weighted by a number of already know neighbourhood patches. Recall that the transition patch detection can be in online version seen as the second analysis step which follows the arrangements assembly. Again the final result synthesis follows this second analysis step. Moreover, if the assumption of temporal homogeneity is used, we can presume that optimal temporal overlap can be found in the first analysis part of a whole dynamic texture. Because the temporal jumps detection is the dominating time-consuming part of an algorithm, is evaluating in first analysis step greatly reduce the second step computational time.

4.3.2 Multiple shaped tiles

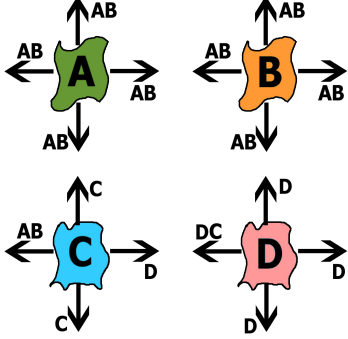


Figure 4.9: *Multiple shaped tiles*: A and B are from one DT, D from another. Patch C is transition between them.

The using of more patches from more inputs, even border type is not satisfactory in many cases. The one universal tile is in many cases not capable of representing the various dynamic texture content.

In this cases, the easy but powerful enhancement could be done. Of course, every approach increasing the generality of the method brings its own limitations and limits the fundamental advantages of the method. In this case, it is a crucial toroidal property. If we give up the toroidal properties on a limited number of samples (transition tiles), we can provide a higher possibility of adapting the method to the desired result mix-of-DTs arrangement.

Instead of locating the transition patches with a given toroidal shape the new optimal transition tile is searched for.

The transition tile can be found not only with a given adjacent content but as a transition between two DTs types and two corresponding optimal tiles. The desired transition tiles then consist of (at least) two cuts from two optimal tiles. The last cuts are searched ad hoc in the online variant, or can be computed as toroidal to transition tile itself.

Again, the optimal rectangular tile for the transition tile searching is founded by error minimization. Lets denote the original pure dynamic texture as \mathcal{Y}^A and \mathcal{Y}^B and the dynamic texture in which the transition is searched as \mathcal{Y}^T . The appropriate index I_{rt}^T set for a rectangular tile is then:

$$I_{rt}^T = I_{rt}^A + \delta^{*A} , \quad (4.11)$$

$$I_{rt}^T = I_{rt}^B + \delta^{*B} , \quad (4.12)$$

The optimal shift δ^{*A} and δ^{*B} are found by minimizing the error cost between at least two cut planes. The fixed tiles cut planes from both pure dynamic textures and all possible locations in the transition texture are searched by spatiotemporal shift. The \mathcal{T} difference is composed of a difference between at least two of temporal-horizontal (top or bottom), temporal-vertical (left or right) or horizontal-vertical (front and end) cut-planes. The appropriate shift is found by error minimization:

$$\delta_r^* = \arg \min_{\delta_r} \left(\begin{aligned} & \epsilon_{th}^*(\mathcal{Y}_T, \mathcal{P}_{bottom}, \mathcal{T}_{bottom}, \delta_r, 0) + \epsilon_{th}^*(\mathcal{Y}_T, \mathcal{P}_{top}, \mathcal{T}_{top}, \delta_r, N) \\ & + \epsilon_{tv}^*(\mathcal{Y}_T, \mathcal{P}_{left}, \mathcal{T}_{left}, \delta_r, 0) + \epsilon_{tv}^*(\mathcal{Y}_T, \mathcal{P}_{right}, \mathcal{T}_{right}, \delta_r, M) \\ & + \epsilon_{vh}^*(\mathcal{Y}_T, \mathcal{P}_{front}, \mathcal{T}_{front}, \delta_r, 0) + \epsilon_{th}^*(\mathcal{Y}_T, \mathcal{P}_{end}, \mathcal{T}_{end}, \delta_r, T) \end{aligned} \right) \quad (4.13)$$

where parameters of every ϵ function are transition texture, pure texture patch and tile, desired shift, and the last parameter established if appropriate cut-plane (from transitional

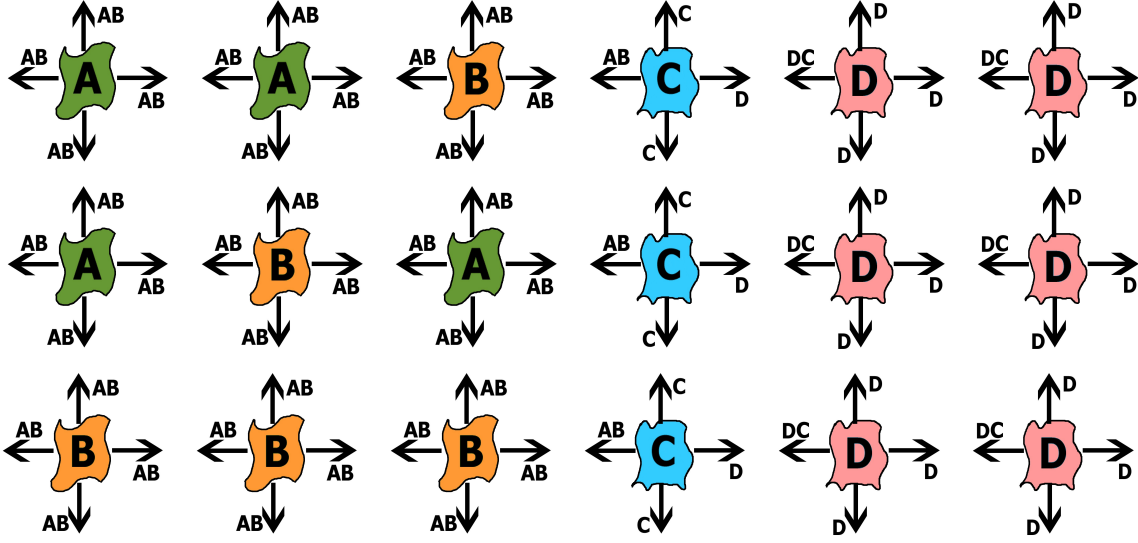


Figure 4.10: *Multiple tiles shore arrangement*: Arrangement from four patches type with one transition tile type.

texture) is left / bottom / front or right / top / end. The \mathcal{P}_S and \mathcal{T}_S are driven by appropriate arrangements. Again, the ϵ^* and $\hat{\epsilon}^*$ are computed as before:

$$\epsilon_{th}^*(\mathcal{Y}, \mathcal{P}, \mathcal{T}, \delta_r, \delta_s) = \begin{cases} 0 & \text{if } \mathcal{P} \text{ is empty and not straight ,} \\ \hat{\epsilon}_{th}(\mathcal{Y}, \mathcal{P}^{\leftarrow}, \mathcal{T}^{\leftarrow}, \delta_r^{\leftarrow}, \delta_s^{\leftarrow}) & \text{if } \mathcal{P} \text{ is empty ,} \\ \hat{\epsilon}_{th}(\mathcal{Y}, \mathcal{P}, \mathcal{T}, \hat{C}, \delta_r, \delta_s) & \text{else ,} \end{cases} \quad (4.14)$$

where \leftarrow denotes results from previous steps and therefore the toroidal knowing possibility of the future patch (as illustrate purple to green and yellow to purple areas in Figure 4.8).

The particular error are computed as:

$$\hat{\epsilon}_{th}(\mathcal{P}, \mathcal{Y}, \mathcal{T}, \delta_r, \delta_s) = \sum_{i=0}^T \sum_{j=0}^N \epsilon \left(\mathcal{P}_{\left[\hat{C}_{[i,j]}^{th}, j, \bullet, i, \bullet \right]}, \mathcal{Y}_{\left[\delta_{\mathcal{P}+\delta^{++}} \left[\delta_s + \hat{C}_{[i,j]}^{th}, j, \bullet, i, \bullet \right] \right]} \right) , \quad (4.15)$$

where $\delta_{\mathcal{P}}$ is patch location (global indices for the $[0, 0, 0]$ of appropriate I_{th}) in the appropriate texture and \hat{C}^{th} is from \mathcal{T} . The ϵ^{tv} , $\hat{\epsilon}^{tv}$, ϵ^{vh} , and $\hat{\epsilon}^{vh}$ are computed in the exact manner.

4.3.3 Temporal Dimension Editing

Lets remind here, that a deterministic finite state machine[52] is a quintuple $(\Sigma, S, s_0, \delta, F)$, where:

- Σ is the input alphabet (a finite, non-empty set of symbols).
- S is a finite, non-empty set of states.
- s_0 is an initial state, an element of S .
- δ is the state-transition function:
 - $\delta : S \times \Sigma \rightarrow S$ (deterministic),
 - $\delta : S \times \Sigma \rightarrow \mathcal{P}(S)$, δ return a set of states (nondeterministic).
- F is the set of final states, a (possibly empty) subset of S

with this on the mind, the interactive textures can be defined as a medium with the (nondeterministic) state machine behavior, where every state $q \in S$ is represented as appropriate patch \mathcal{P} in the meaning of temporal persistence. The \mathcal{P} here represent the whole texture \mathcal{Y} .

Interactive texture The DTs mixing and editing can be done directly in temporal meaning only. The input textures and their corresponding patches can be manually labeled by its global and local dynamics (i.e. strong of wind, the density of rain and so on).



Figure 4.11: THUJA in various rain condition with probabilities of change of states.

The patches from their labeled textures are then found and labeled so on. The SM states then represent \mathcal{P} . The transitions δ represents temporal cut:

$$\mathcal{P}_A \xrightarrow{\delta} \mathcal{P}_A \quad (4.16)$$

and/or transition to another \mathcal{P} :

$$\mathcal{P}_A \xrightarrow{\delta} \mathcal{P}_B \quad (4.17)$$

where the (4.3.3) represent loop and (4.3.3) state change. Note that if more temporal jumps exist, the more loops are defined. Likewise more very similar patches can be labelled as one state with inside transitions or the superpositioned state can be defined. s_0 is any state, F is empty. The Σ consist of one symbol that drives the state change every T frames. Alternatively, Σ consist of one symbol per every $q \in S$ and state changes are driven. In the automatic approaches every δ the state-transition function has its probability p .

For example for THUJA *SM* in figure 4.11 four states of a dynamic texture is illustrated: HEAVY RAIN, START RAIN, STOP RAIN and NO RAIN. All states except NO RAIN are represented by one patch, while the NO RAIN is represented by two patches NO RAIN (TEXTURE-1) and NO RAIN (TEXTURE-2). The arrows denotes possible changes in the states of a dynamic texture with corresponding probability p :

$$\begin{array}{lcl}
 \text{NO RAIN} & \xrightarrow{0.3} & \text{START RAIN} \\
 \text{START RAIN} & \xrightarrow{1} & \text{HEAVY RAIN} \\
 \text{STOP RAIN} & \xrightarrow{1} & \text{NO RAIN} \\
 \text{HEAVY RAIN} & \xrightarrow{0.3} & \text{STOP RAIN}
 \end{array}$$

Note the $p = 1$ probabilities handle the nonrepeatable textures. This transition textures has the same meaning like transition textures in the section 4.3, but with strictly temporal meaning. With rule that can chance the inner state of the NO RAIN:

$$\begin{array}{lcl}
 \text{NO RAIN (TEXTURE-1)} & \xrightarrow{0.4} & \text{NO RAIN (TEXTURE-2)} \\
 \text{NO RAIN (TEXTURE-2)} & \xrightarrow{0.4} & \text{NO RAIN (TEXTURE-1)}
 \end{array}$$

and with complementary rules to ensure a possible repeat of the same texture:

$$\begin{array}{lcl}
 \text{HEAVY RAIN} & \xrightarrow{0.3} & \text{HEAVY RAIN} \\
 \text{NO RAIN (TEXTURE-2)} & \xrightarrow{0.3} & \text{NO RAIN (TEXTURE-2)} \\
 \text{NO RAIN (TEXTURE-1)} & \xrightarrow{0.3} & \text{NO RAIN (TEXTURE-1)}
 \end{array}$$

the rules set is complete.

The resulting interactive texture could be played fully automatically with predefined probabilities to vary its appearance. The state changes, even if not strongly vary itself and its nondeterministic repeating strongly enhance the visual quality of the results.

For the nondriven, automatic dynamic texture the patches are organised with the transition from state to state:

$$q_s \longrightarrow \left\{ \begin{array}{l} \delta(q_0, x), \text{ with probability } p_0 \\ \delta(q_1, x), \text{ with probability } p_1 \\ \dots \\ \delta(q_j, x), \text{ with probability } p_j \end{array} \right. , \quad \sum_{i=0}^j p_i = 1, q_s \in S, x \in \Sigma. \quad (4.18)$$

where $card(q)$ denotes the state cardinality.

Local Temporal Shift In order to avoid the typical synthesizing problem - periodicity (usually given by the small amount of input data) - the spatial editing methods like multiple patches, several inputs DTs or a mix-of-DTs are utilised. Besides editing of the spatial dimension to increase results variability, another pure temporal method can be used.

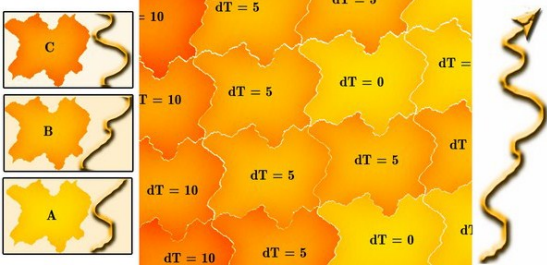


Figure 4.12: *Arrangement with temporal shift*: Example of patch arrangement with temporal shift. Three patches with different time shift (0, 5 and 10 frames) and different local (arrows) optical flow, respectively.

analysis steps.

The time shift Λ should be applied gradually with increasing size with the distance from the appropriate \hat{C} cut plane. The particular gradual λ step should be limited by T to avoid recognizable distortions. Obviously, the size of the area in which the time is gradually changed is:

$$[\mathcal{E}_\Lambda] = \frac{\Lambda}{\lambda} . \quad (4.19)$$

If $\lambda < 1$, the desired values, of course, does not exist. The desired data could be approximated (which typically leads to blurring) of the nearest existing frame, or a rounded t -th $[\lambda_t]$ one could be used. The Λ and appropriate λ values are chosen as:

$$2[\mathcal{E}_\Lambda] < \min(h, v, t) , \quad (4.20)$$

of if the interior patches are supposed to have only consistent time shift as:

$$[\mathcal{E}_\Lambda] < \min \left(\begin{array}{l} \min \left(\min_{\forall i,j \in I_{tv}} \hat{C}_{[i,j]}^{tv} , M - \max_{\forall i,j \in I_{tv}} \hat{C}_{[i,j]}^{tv} \right) , \\ \min \left(\min_{\forall i,j \in I_{th}} \hat{C}_{[i,j]}^{th} , N - \max_{\forall i,j \in I_{th}} \hat{C}_{[i,j]}^{th} \right) , \\ \min \left(\min_{\forall i,j \in I_{hv}} \hat{C}_{[i,j]}^{hv} , T - \max_{\forall i,j \in I_{hv}} \hat{C}_{[i,j]}^{hv} \right) \end{array} \right) , \quad (4.21)$$

This technique strongly increases the quality of results with suppressing the visual periodicity. While the patches remain structurally the same, the different dynamics drastically suppress the visual periodicity. For examples and the detailed view, we refer the results section.

Dynamic Texture Inpainting

This chapter overview the inpainting problem and capability of our dynamic texture model to deal with it (see Figure 5.1 for results illustration). First, the inpainting problem is addressed in section 5.1 in the relation of this dissertation thesis. Next, in section 5.2, the adaptation of the presented synthesizing and editing model to inpainting problem is explained together with addressing the different condition and circumstances like an inability to obtain exact ground truth and ways to handle it. The section 5.3 consist of core definition of a inpainting model principles. Finally, in the last section, 5.4 is explained how to bypass the need for a mask for inpainting algorithm and cases where it is possible.

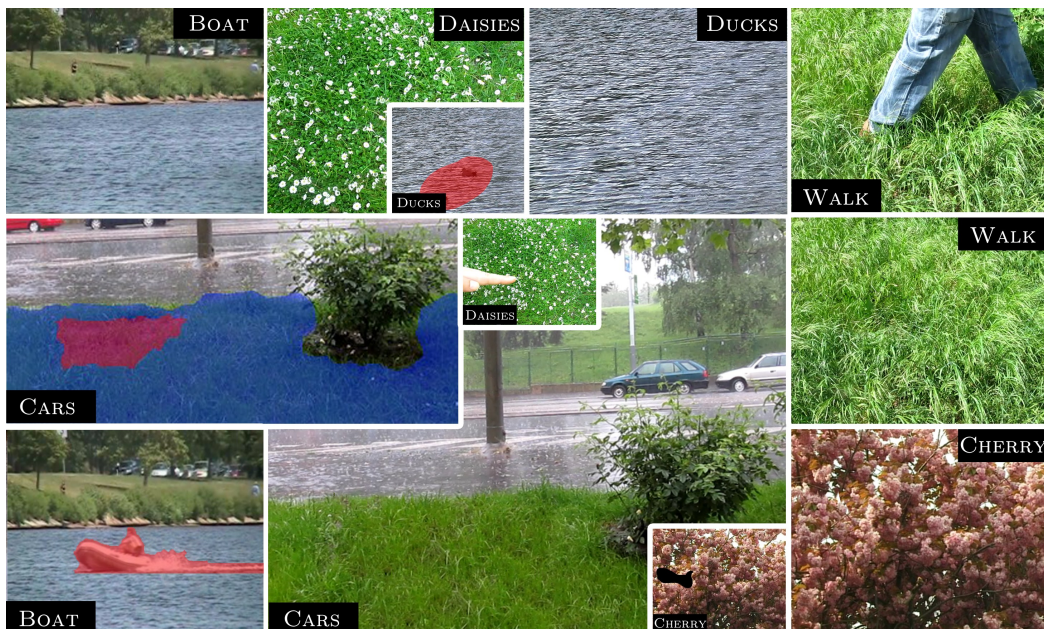


Figure 5.1: Several inpainted dynamic textures: Source and inpainted frames showed.

5.1 Problem definition

As discussed previously in chapter 2, the inpainting and error concealment problem addresses many particular problems with various assumptions and circumstances. The main key property for this problem consists of:

- Does a part of a covered texture occur in any axial moment?
- Does the scene consist of one dynamic texture or more dynamic textures?
- Is area to inpaint artificial or naturally occurs in given scene?

This very basic question could determine the circumstances and method limitations. It is obvious that in the case of using an artificial mask (like in all THUJA DTs) there is already ground truth information. Regrettably, this does not mean that the original DT part and newly inpainted data could be easily compared, i.e. directly by pixel-by-pixel distance. As the texture (and subsequently dynamic texture) definition says, the texture as some random field realization could vary in its exact (pixel) values. In other words, the inpainted part of texture can be very different and still have the property of the original texture. This drastically limits the possibilities of comparison or even determination which method is suitable, better, or even satisfyingly working (more at chapter 6). The existence of the original dynamic texture, of course, could be used for visual comparing or psycho-physical test.

The critical role in problem has, of course, the object which is meant to be inpainted. Even if the occluding object naturally occurs in the scene, there are several cases that may more occur. In the first case, the object is moving through scene, or during the duration of the scene, it exits (or enters). So there is a time momentum for every location in the texture, for which it is unoccluded. Note that the specific amount of time depended on scene dynamics is required and so short sequence even for whole texture is not enough. Moreover, the size of the area to inpaint must be taken into account. The size could be expressed precisely as a percentage of the actually viewed data and to whole dynamic texture (if the object varies in size) or inaccurately with relevance to the property of the inpainted dynamic texture. In the latter case, it is appropriate to determine whether the area to inpaint is substantially smaller than the underlying objects (mostly lower frequencies) from which the dynamic texture is composed, the same size or greater size (i.e. waves on water DT, or leaves/branches in the trees containing DT).

Of course, the primary case of a scene containing more than only one dynamic texture is different from scene consisting of only one. Ideally, the complex scenes could be divided into segments which could be handled separately.

The case with object naturally occurs in the scene can be extended to the case with many objects. The higher number of objects themselves may not be problematic. The problem occurs if we want to keep some of them and inpaint only the others. The case with many moving objects all of which are intended to be removed can be addressed as

above. Let's note that the problem becomes drastically more difficult if the original texture is supposed to preserve.

With the above-defined assumptions, it is possible to define the following cases to be used in this work:

- Dynamic texture with an artificial mask.
- Complex scene containing more DT with an artificial map.
- Naturally occluded dynamic texture with a significantly smaller object (objects).
- Naturally occluded dynamic texture with a significantly larger object (objects).
- Naturally occluded complex scene with a significantly smaller object (objects).

5.2 Toroidal-Patch Model Adaptation

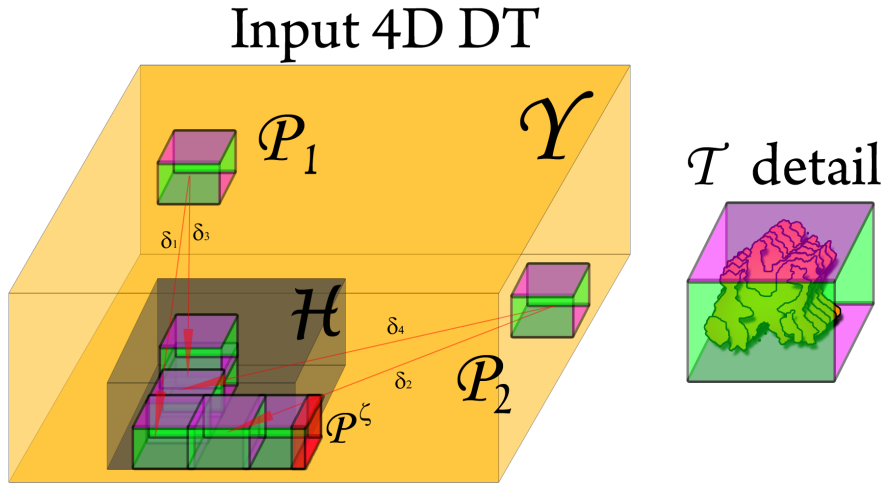
The presented triple toroid-shaped patch model could be easily adapted to inpainting problem by inpaint by synthesis approach. The analysis part of the algorithm that focuses on the dynamic texture which is damaged could determine the property of a damaged texture (i.e. optimal tile size). The input 4D DT's frequencies and the optimal double toroid-shaped tile are computed by using the Fourier analysis. Additionally, time properties are analysed again by a pyramidal Lucas-Kanade optical flow of the whole 4D and Fourier analysis. Finally, the search for similar patches of the same shape is done together with and computing the optimal temporal jumps.

Synthesizing part of the approach is based on finding small triple toroid-shaped tiles which can be used to create seamlessly fill the \mathcal{H} of arbitrary size in any spatial and temporal dimension. The exact location and shapes of triple toroid-shaped tiles can be improved by the error mask which is usually given by the user but is not necessary for presented toroidal approach.

Thus the method could be easily modified to synthesise the data to fill the error hole or replace the unwanted object with newly synthesised texture.

The presumption of toroidal tile shape is (in almost all cases) in contradiction to the shape of area \mathcal{H} to inpaint. This could be easily solved by recomputing the patch in significantly smaller and restricted area. In fact and with using of some assumptions and textural property this does not need to be taken as a disadvantage.

On the other hand, the toroidal approach allows reducing the computational time drastically. Classical intelligent sampling and statistical methods typically try to minimize cost around the three-dimensional spatial area \mathcal{H} , so the amount of method calculation is therefore dependent on the size of a \mathcal{H} . This lead to significantly increasing computational cost with the \mathcal{H} size. Furthermore, as the synthesized area consists from more samples, the inside borders and patch require their own optimization and thus the methods computational time depends on both error hole surface and volume. Compared to that, the toroidal approach computation is only weakly dependent (or complete independent in some cases)

Figure 5.2: *Inpainting principle illustration.*

on \mathcal{H} size at all as the computation time is given only by tile size which used multiple tiles. The all possible hole dependent calculation consists only on a one-off calculation of the optical flow value and its comparison and outside border error minimization.

The error hole size

The area to inpaint, both artificial or natural, is denoted as \mathcal{H} . Usually is visualised as a black or white area in the dynamic texture visualization in the case of artificial mask and as a reddish area in case of removing some objects from the scene (see Figure 5.1). In the particular case of dynamic texture with no mask (as WALK and DAISIES) no mask is shown.

The key \mathcal{H} property is, of course, its size. Due to dynamic texture analysis and \mathcal{T} size given by the dominant texture frequency, the optimal tile \mathcal{T} could be smaller or larger than the existing \mathcal{H} . In the first case, the \mathcal{H} in texture is smaller than the primary visual part of surrounding dynamic texture. The dominant frequency given optimal tile could be relatively small in case of very stochastic texture (like water, sand, ...) but relatively large for texture consists of trees of more complex scenes too. In this case, the set of patches must be used to cover the whole \mathcal{H} area as illustrated on Figure 5.2. The smaller patch size allows utilizing some degrees of freedom - moving with the patch in both spatial and temporal axis to find the minimal boundary cost. Of course, hence the combining size of used \mathcal{P} is larger than the \mathcal{H} and some overlap is created (see Figure 5.2, the ζ^* area) In the second case the \mathcal{T} is significantly larger than the \mathcal{H} and again, some overlap is created.

Of course, the using only necessary subpart of a used patch is possible (again with possible shifts) but this, even for well-computed data patches, usually leads to visually easy recognizable silhouettes (see Figure 5.6 and accompanying videos) . The final approach is to use I_ζ area to find the \hat{C} between the \mathcal{H} inside similar as interior part of

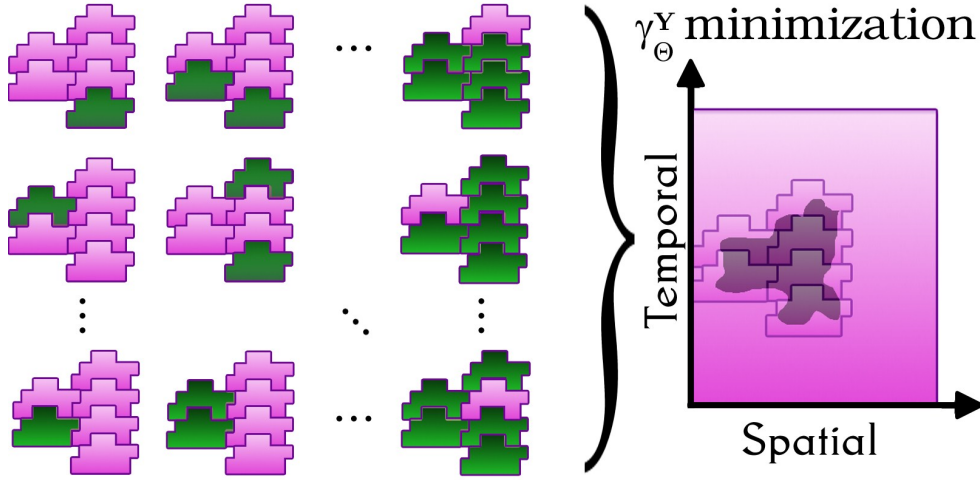


Figure 5.3: *Inpainting by synthesis*: Inpainting hole by synthesizing a set of textures (left) arranged from six patches of two types.

a rectangular tile I_T and surrounding dynamic texture. The area and appropriate index set could be easily approximated by a rectangular area $I_\zeta^r = I_\zeta \setminus I_{\mathcal{H}}$. Emphasize that in this case the presented method partially lost its great advantage of independence to \mathcal{H} size. Moreover, as the proposed test showed, the using of an optimal synthesizing tile is sufficient, and then the \mathcal{H} -optimal patch is not necessary.

The synthesis of dynamic texture to fill the \mathcal{H} consists of the appropriate use of the computed patches \mathcal{P} and to cover it. The main idea is to cover a hole and minimal amount of patches placed on the three-dimensional tile label lattice. The toroidal property allows covering the \mathcal{H} area with a seamlessly consisting texture that sufficiently represents surroundings dynamic texture with an advantage of the possibility of assembling many arrangements of tiled tiles (see Figure 5.3).

The \mathcal{H} neighbourhood

Lets note \mathcal{H}^ζ as the \mathcal{H} neighbourhood area. The whole rectangular index set $I_\zeta^r = I_\zeta \setminus I_{\mathcal{H}}$ is its more straightforward and rough form and consist of possible unnecessary areas and could be fitted better.

As the ζ area of a particular \mathcal{P}_i is $\mathcal{P}_i^\zeta = \mathcal{P} \setminus \mathcal{H}$. The more suitable \mathcal{H}^ζ can be defined as a symmetric difference of \mathcal{H} and all used \mathcal{P} 's area: $\mathcal{H}^\zeta = \mathcal{H} \Delta \cup_{i=0}^n \mathcal{P}_i$ The desired $[\mathcal{H}^\zeta]$ area size is of course minimal with meaning of pixel count.

The error cost then consists similarly like in the transition patches founding approaches from minimizing the appropriate cut-planes error with a given tile shape. The desired δ_r^* shift is due to (combined) \mathcal{T} size with relation to \mathcal{T} size limited.

Lets denote the rectangular index of \mathcal{H} as $I_{\mathcal{H}}$:

$$I_{\mathcal{H}} = (0, \dots, \lfloor \mathcal{H} \rfloor_h) \times (0, \dots, \lfloor \mathcal{H} \rfloor_v) \times (0, \dots, \lfloor \mathcal{H} \rfloor_t). \quad (5.1)$$

The surrounding area in which the tile could occur is then of limited by its original optimal \mathcal{T} size. With relation to $I_{\mathcal{H}}$, the appropriate index set is:

$$\begin{aligned} I_{\zeta} &= (-\lfloor \mathcal{P} \rfloor_h, \dots, \lfloor \mathcal{H} \rfloor_h + \lfloor \mathcal{P} \rfloor_h) \\ &\times (-\lfloor \mathcal{P} \rfloor_v, \dots, \lfloor \mathcal{H} \rfloor_v + \lfloor \mathcal{P} \rfloor_v) \\ &\times (-\lfloor \mathcal{P} \rfloor_t, \dots, \lfloor \mathcal{H} \rfloor_t + \lfloor \mathcal{P} \rfloor_t). \end{aligned} \quad (5.2)$$

The I_{ζ} is then the neighborhood of the hole \mathcal{H} in which the final (possibly combined) tile shape could be placed to encompass it.

The desired δ_r^* naturally could come only from I_{ζ} . Moreover, the only some subspace of I_{ζ} is appropriate. The desired subspace of I_{ζ} is its left-bottom-front quadrant.

Lets denote the left-bottom-front I_{ζ^-} quadrant as I_{ζ^-} .

$$\begin{aligned} I_{\zeta}^- &= (0, \dots, \lfloor \mathcal{H} \rfloor_h - \lfloor I^T \rfloor_h \times 2) \\ &\times (0, \dots, \lfloor \mathcal{H} \rfloor_v - \lfloor I^T \rfloor_v \times 2) \\ &\times (0, \dots, \lfloor \mathcal{H} \rfloor_t - \lfloor I^T \rfloor_t \times 2). \end{aligned} \quad (5.3)$$

or as a I_{ζ} subspace:

$$\begin{aligned} I_{\zeta}^- \subset I_{\zeta} &= (-\lfloor \mathcal{P} \rfloor_h, \dots, -\lfloor \mathcal{P} \rfloor_h + \lfloor \mathcal{H} \rfloor_h - \lfloor I^T \rfloor_h \times 2) \\ &(-\lfloor \mathcal{P} \rfloor_v, \dots, -\lfloor \mathcal{P} \rfloor_v + \lfloor \mathcal{H} \rfloor_v - \lfloor I^T \rfloor_v \times 2) \\ &(-\lfloor \mathcal{P} \rfloor_t, \dots, -\lfloor \mathcal{P} \rfloor_t + \lfloor \mathcal{H} \rfloor_t - \lfloor I^T \rfloor_t \times 2). \end{aligned} \quad (5.4)$$

then the optimal shift searching is based on minimizing appropriate errors:

$$\begin{aligned} \delta_r^* &= \sum_{\forall \mathcal{P}_i^B} \arg \min_{\delta_r \in I_{\zeta}^-} \left(\hat{\epsilon}_B(\mathcal{Y}, \mathcal{P}_i^B, I_{\zeta}, \delta_r) \right. \\ &\left. + \sum_{j=0}^T \sum_{\forall \mathcal{P}_i} \epsilon^{\phi}(\bar{\phi}(\mathcal{P}_{r_4=t}), \bar{\phi}(\mathcal{Y}_{[r_4=t+\delta_{r_4}]})) \cdot |\mathcal{P} \cap \mathcal{H}| \right) \end{aligned} \quad (5.5)$$

where \mathcal{P}^B are border patches only, $\hat{\epsilon}_B$ is border cut-planes error and ϵ^{ϕ} is the dynamic error. Note that ϵ^{ϕ} should be expected to zero in time-homogenous dynamic texture (as the all \mathcal{P} are representative texture samples). Moreover, the $\bar{\phi}$ values are precomputed, so their difference estimation is extremely effective. Remind that $\bar{\phi}$ is average optical flow

vectors and ϵ^ϕ their difference. The $\lfloor \mathcal{P} \cap \mathcal{H} \rfloor$ factor scale the error down for only partially used patches.

The particular error for cut planes are computed as:

$$\begin{aligned} \hat{\epsilon}_B(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r) = & \left(\begin{aligned} & \epsilon_{th}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, 0) + \epsilon_{th}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, N) \\ & + \epsilon_{tv}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, 0) + \epsilon_{tv}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, M) \\ & + \epsilon_{vh}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, 0) + \epsilon_{vh}^*(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta_r, t) \end{aligned} \right) \quad (5.6) \end{aligned}$$

where the particular cut-plane error are computed only if the plane is outside plane or, in another words, for only unique indices:

$$\begin{aligned} \hat{\epsilon}_{th}(\mathcal{P}, \mathcal{Y}, \mathcal{T}, \delta_r, \delta_s) &= \sum_{i=0}^T \sum_{j=0}^N \hat{\epsilon}_{i,j} \quad (5.7) \\ \hat{\epsilon}_{i,j} &= \begin{cases} 0 & \text{if } \delta_s = 0 \wedge (\hat{C}_{[i,j]}^{th} - \lfloor \mathcal{P} \rfloor_h) \notin I_\zeta \\ 0 & \text{if } \delta_s \neq 0 \wedge (\hat{C}_{[i,j]}^{th} + \lfloor \mathcal{P} \rfloor_h) \notin I_\zeta \\ \epsilon \left(\mathcal{P}_{[\hat{C}_{[i,j]}^{th}, j, \bullet, i, \bullet]}, \mathcal{Y}_{[\delta_{\mathcal{P}} + \delta^+ + [\delta_s + \hat{C}_{[i,j]}^{th}, j, \bullet, i, \bullet]]} \right) & \text{, otherwise.} \end{cases} \quad (5.8) \end{aligned}$$

Here, $\delta_{\mathcal{P}}$ is patch location (global indices for the $[0, 0, 0]$ of appropriate I_{th}) in the appropriate texture and \hat{C}^{th} is from \mathcal{T} . The $\hat{\epsilon}^{th}$, $\hat{\epsilon}^{tv}$, ϵ^{vh} , and $\hat{\epsilon}^{vh}$ are computed in the exact manner.

An example can be seen in Figure 5.4 where lime, green, pink and violet ellipses mark the bot verticals and both horizontal cuts. Every color marked cuts is then computed with distance to appropriate toroidal cuts (lime to green and pink to violet) and the appropriate color marked area in the \mathcal{H}^ζ .

5.3 Dynamic Texture Inpainting

The proposed inpainting method (see Figure 5.4) is as follows: *Input* - obtaining an input 4D DT and given mask (the retained parts are denoted by the mask in blue while the parts to be removed parts are in red) or computed of one. *Double Toroid-Shaped Tile* - the optimal tile computing using FT. The red ovals marks area for vertical toroidal cut, violet ovals marks area for horizontal toroidal cuts. *A set of Double Toroid-Shaped Tiles* - if the mask is given, the double toroid-shaped tile borders are fitted to hole (denoted by red in input mask). The tiles are (as suggest layered image) propagated through time by the OF and their exact shape is computed. The color ovals mark corresponding areas for recomputing the patch borders. *Triple Toroid-Shaped Tiles* - the set of double toroidal-shaped tiles are used to find optimal temporal jump (blue circle arrow) and create the triple toroidal-shaped tiles. The red round arrow denotes the exact continuity of the trimmed temporally toroidal sample.

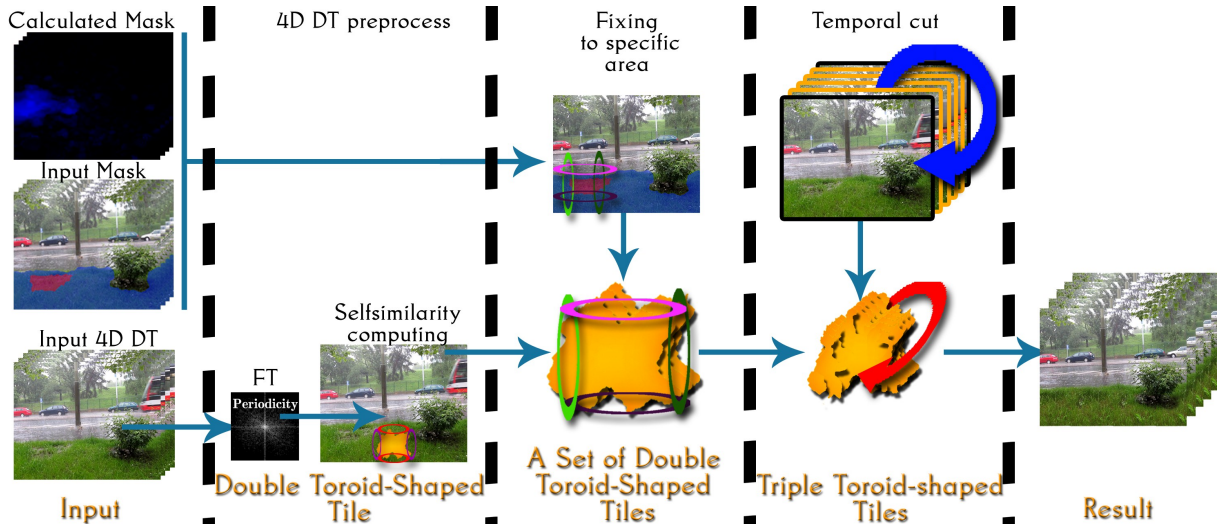


Figure 5.4: The overall flowchart of the presented DT inpainting method. The straight blue arrows indicate the procedure of the present method.

The input 4D DT's frequencies and the optimal double toroid-shaped tile are computed by using the Fourier analysis, their time properties and pyramidal Lucas-Kanade optical flow of whole 4D input are analyzed in the parallel pipeline, followed by the search for similar patches of the same shape and computing the optimal temporal jumps. This information is used for its generalization to the dynamic triple toroid-shaped tile as in pure synthesis variant.

The exact location and shapes of triple toroid-shaped tiles can be improved by the error mask which is usually given by the user but is not necessary for the presented approach. Usually, we use only rough hand-drawn area which is entirely sufficient for good results.

The static double toroid-shaped tiles are found by the pure synthesis method and later expanded using DTs optical flow and propagated throughout the time. The tiles are finally fitted to local dynamics by minimizing the boundary error (5.6) and estimating the optimal tiles shift (5.5). The specific \mathcal{H} cut-planes can be computed in the whole I_{ζ}^r area in a similar manner as whole new optimal not-toroidal tile or as a composition of more planes for the I_{ζ} .

Emphasize that as the \mathcal{H} do not need to have similar size in all dimension, the ideally fitted cut plane could consist of several cut planes for a general nonrectangular I_{ζ} . The different plane fitting method could be used but as their result are visually problematic (See Figure 5.6) that approaches are not used in this thesis.

Moreover, with the assumptions that the input data are a homogenous stochastic dynamic texture or mix-of-DTs and that the used patches \mathcal{P} and so every particular used patch are a dynamic texture of the same type with the same (or very similar) property the created output will be consistent DT with the same type. Thus in case of temporal homogeneity:

$$\epsilon^{\phi} \rightarrow 0, \quad (5.9)$$



Figure 5.5: *Randomly placed patched with globally optimal shape*: The original DTs WITHERED and CHERRY (left column) and randomly placed inpainted patches (right column, marked patches middle column). Both ϵ^ϕ and $\hat{\epsilon}_B$ simplifying assumption used.

as the dynamics is consistent through temporal dimension.

The border error and visually discriminable inpainted part correlates with the increasing occurrence of complex structures, and hence with declining randomness. For a significant part of dynamic textures this is satisfied, and thus the tile can be computed globally, and the exact fitting is not required. Thus, moreover, the

$$\hat{\epsilon}_B(\mathcal{Y}, \mathcal{P}_i^B, I_\zeta, \delta) \approx \hat{\epsilon}_B(\mathcal{Y}, \mathcal{P}_j^B, I_\zeta, \delta) \quad \mathcal{P}_i^B, \mathcal{P}_j^B \in \mathcal{S}_{inp} \quad , \quad (5.10)$$

where \mathcal{S}_{inp} is arrangement of patches that cover the error hole. And thus the $\hat{\epsilon}$ estimation could be bypassed for very stochastic and selfsimilar dynamic textures.

The illustration of this textural property can be seen in Figure 5.5, where founded optimal patch are randomly placed to and dynamic texture. Both simplifying cases are used in both CHERRY and WITHERED. Note that in WITHERED the first simplifying assumptions are used very loosely and mostly as an example for the test. Both simplifying

cases are not of course mutually exclusive. Note that in the dynamic texture the main and mostly dominating property is texture's dynamics and so the similar dynamics of the inpainted patch and its neighbourhood is crucial.

Typical inpainting problems

Every inconsistency criterion has its own typical manifestation (see Figure 5.6). The sufficiently synthesized dynamic texture of a patch is, of course, crucial for a suitable inpainting result. The inadequate texture synthesis results are easy to observe as it is de facto just replacing one unwanted texture (object) with another. Usually, this criterion is satisfied. The manifestation of wrongly synthesized texture could be missing or shifted high frequencies, which are easily recognizable even in still frames. The insufficiently optimized border error leads to visually easy observable patch location and discontinuity of the dynamic texture in both spatial and temporal dimensions even if synthesized texture itself could have similar textural property. It is important to distinguish between border error caused by wrongly synthesized dynamic texture (waves DT in Figure 5.6) and sufficiently synthesized texture with high border error (stairs).

Mostly the border error optimizing is insufficient only in the temporal domain - the BOAT texture (bottom in Figure 5.6) seems to be sufficient and challenging to detect in the still picture visually, but are easily recognizable as moving silhouette when observed as a video. This is a typical problem of methods that optimize the patch frame by frame or do not distinguish between spatial and temporal dimensions. As for the dynamic texture synthesis, the approach from chapter 3 is used, the essential texture frequencies together with structural and color property are preserved. Toroid-shaped patches of a suitable size naturally respect and reproduce the lost frequencies without loss of structural information and without the need to adjust the results through an artificial supply of texture features, estimating parameters and other techniques like orthomorphologies [97, 55] which can enhance the quality of the result.

Local dynamics problems are usually caused by the periodic motion of small objects in the texture and create the optical flow. Not preserving of the optical flow causes a violation of $G^{\{1,2,3\}}$ continuity curves - inconsistencies in dynamics due to the entire image, create a false optical flow, or abrupt changes in optical flow velocity.

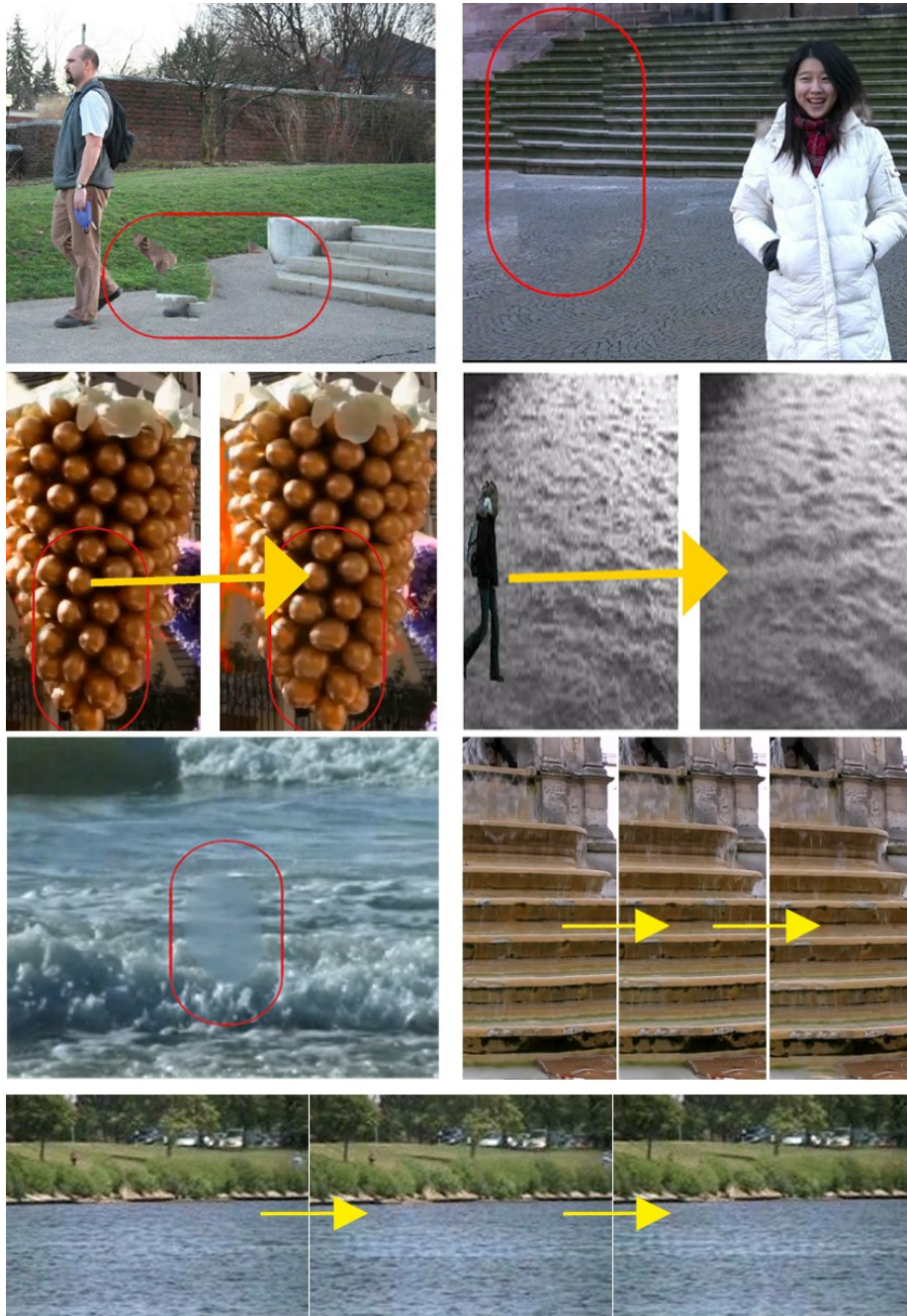


Figure 5.6: *Typical inpainting problems*; The first row left and right respectively: Synthesis fails due to inadequate patch size [97], shifted vertical frequencies [55]. The second row left and right respectively: temporal distortion and inconsistencies between frames [77, 26]. Third row: missing high frequencies, inconsistency between frames and moving mask silhouette [97]. Fourth row: recognizable (better to see in videos) moving mask silhouette [142].

The potential maximum number of needed patches c for rectangular I_S trivially depends on the hole and patch area sizes and equals to:

$$c = c_v \cdot c_h \cdot c_t , \quad (5.11)$$

$$c_h = \begin{cases} \left\lceil \frac{|\mathcal{H}|_h}{|\mathcal{P}|_h} \right\rceil \\ \left\lceil \frac{|\mathcal{H}|_h}{|\mathcal{P}|_h} \right\rceil + 1 \end{cases} , \quad c_v = \begin{cases} \left\lceil \frac{|\mathcal{H}|_v}{|\mathcal{P}|_v} \right\rceil \\ \left\lceil \frac{|\mathcal{H}|_v}{|\mathcal{P}|_v} \right\rceil + 1 \end{cases} , \quad c_t = \begin{cases} \left\lceil \frac{|\mathcal{H}|_t}{|\mathcal{P}|_t} \right\rceil \\ \left\lceil \frac{|\mathcal{H}|_t}{|\mathcal{P}|_t} \right\rceil + 1 \end{cases} . \quad (5.12)$$

But of course can by lower is the \mathcal{H} area is strongly non-convex (see Figure 5.3). The iterative border patch \mathcal{P}_{B_i} selection process would lead only to a local optimum. This greedy approach could naturally lead to a non-optimal solution. The global optimum can be achieved by the synthesis of all possible \mathcal{P} -arrangements \mathcal{S} (see Figure 5.3) for effectively minimize (5.6) error.

The detection of a toroid-shaped patch does not require a knowledge of the mask. The criterion of patch minimal error is trivially minimized for samples consisting entirely from one single dynamic textures from which the \mathcal{H} surrounding consist from. As the rough tile size and thus even possible arrangement can be known before computing the tiles itself, the optimal patch tile could be computed can be created concerning the future $\hat{\epsilon}$ minimization. Thus using of, even consistent and optimal toroidal, patches from different dynamic texture occurring in given input (mix-of-texture) naturally broke criterion and are excluded. Detection of multiple samples and their mutual errors then ensure elimination of a sample that consists entirely of \mathcal{H} . This can be arranged by excluding the \mathcal{H} data by its mask (is exist) and using a positive mask (denoting the source texture for tile computation).

5.4 Automated DT inpainting

The hole or an object to be inpainted is usually given by the mask [55] or segmentation as the preprocessing is needed. Since presented approach deals mainly with DTs, which must preserve textural property, these undesirable parts can be roughly detected automatically. v, Υ

Presented approach can detect inconsistencies primary due to different optical flow [87]. The 5D optical flow field Φ is used to detect parts of texture with different local dynamics.

The dissimilarity measure Υ detects a measure from a some (small) area $\Delta_{\{r\}}^p$ with dimensions given by $p = [p_1, p_2, p_3]$ and centered around r to another the most diverging area $\Delta_{\{r+\delta_r\}}^p$. $\phi(\Delta_{\{r\}}^p)$ denotes dynamics of the area and $I_{\{r\}}^{\Delta^p}$ its index set.

The dissimilarity measure of an area $\Delta_{\{r\}}^p$ to the rest of \mathcal{Y} is basically sum of area-to-area measures from original to every another area in the texture (with excluding overlaps)

$$\Upsilon^*(\Delta_{\{r\}}^p, Y) = \sum_{i=[\Delta^p]_x}^{M-[\Delta^p]_x} \sum_{j=[\Delta^p]_y}^{N-[\Delta^p]_y} \sum_{k=[\Delta^p]_t}^{T-[\Delta^p]_t} \Upsilon(\Delta_{\{r\}}^p, \Delta_{\{[i,j,k]\}}^p) , \Delta_{\{r\}}^p \cap \Delta_{\{[i,j,k]\}}^p = 0 . \quad (5.13)$$

The dissimilarity measure Υ between two areas and their corresponding inside dynamics vectors is defined as:

$$\Upsilon(\Delta_{\{r\}}^p, \Delta_{\{s\}}^p) = \max \left\{ \frac{\sum_{\forall d \in I_{\{r\}}^{\Delta^p}} v \left(\mathcal{Y}, \Delta_{\left\{ r + \left(d - \frac{\lfloor \Delta^p \rfloor_{[x,y,t]}}{2} \right) \right\}}^p, \Delta_{\left\{ s + \left(d - \frac{\lfloor \Delta^p \rfloor_{[x,y,t]}}{2} \right) \right\}}^p \right)}{\lfloor \Delta^p \rfloor} \right\} \quad (5.14)$$

$$- \min \left\{ \frac{\sum_{\forall d \in I_{\{s\}}^{\Delta^p}} v \left(\mathcal{Y}, \Delta_{\left\{ r + \left(d - \frac{\lfloor \Delta^p \rfloor_{[x,y,t]}}{2} \right) \right\}}^p, \Delta_{\left\{ s + \left(d - \frac{\lfloor \Delta^p \rfloor_{[x,y,t]}}{2} \right) \right\}}^p \right)}{\lfloor \Delta^p \rfloor} \right\} ,$$

where $v(\mathcal{Y}, r, s)$ is a measure between \mathcal{Y}_r^ϕ and \mathcal{Y}_s^ϕ . $\lfloor \Delta^p \rfloor_{[x,y,t]}$ is size vector $c = [c_1, c_2, c_3]$ composed from three x, y, t values respectively. \mathcal{Y}_r^ϕ is equal to $\phi(\mathcal{Y}_r)$ and therefore to optical flow vector for corresponding pixel r from dynamic texture \mathcal{Y} . Both \mathcal{Y}_r^ϕ and \mathcal{Y}_s^ϕ values are part of the superstructure areas with the corresponding shift inside. The particular measure of corresponding pixel vectors is:

$$v(\mathcal{Y}, r, s) = \mathcal{Y}_r^\phi * \mathcal{Y}_s^\phi \frac{\max \left\{ \left| \mathcal{Y}_r^\phi \right|, \left| \mathcal{Y}_s^\phi \right| \right\}}{\min \left\{ \left| \mathcal{Y}_r^\phi \right|, \left| \mathcal{Y}_s^\phi \right| \right\}} , \quad (5.15)$$

r and s are multiindices. $\lfloor \Delta^p \rfloor$ denotes the size of the used areas, $*$ is a scalar product. High values of Υ indicate parts with different dynamics than the rest of the dynamic texture. The equation (5.16) computation between some area and whole rest of the dynamic texture can be effectively enhanced by Υ computation to not the whole \mathcal{Y} but only to some randomized placed Δ areas. We usually use $\Upsilon_n = 5$ to 7 areas with the size of $\propto \lfloor \mathcal{P} \rfloor$ plus forbidding computation of very near (adjacent) pairs which forbids local similarity in an inpainted area. The 5.16 then becomes:

$$\Upsilon(\Delta_{\{r\}}^p, Y) = \sum_{n=0}^{\Upsilon_n} \Upsilon(\Delta_{\{r\}}^p, \Delta_{\{\{rand_i, rand_j, rand_k\}\}}^p) \quad (5.16)$$

$$rand_i \in \langle 0, \dots, r_x - 2 \times \lfloor \Delta^p \rfloor_x, r_x + 2 \times \lfloor \Delta^p \rfloor_x, \dots, M \rangle \quad (5.17)$$

$$rand_j \in \langle 0, \dots, r_y - 2 \times \lfloor \Delta^p \rfloor_y, r_y + 2 \times \lfloor \Delta^p \rfloor_y, \dots, M \rangle \quad (5.18)$$

$$rand_k \in \langle 0, \dots, r_t - 2 \times \lfloor \Delta^p \rfloor_t, r_t + 2 \times \lfloor \Delta^p \rfloor_t, \dots, M \rangle . \quad (5.19)$$

Emphasize that if the locations of $\Delta_{\{r\}}^p$ regions are randomized and Υ is calculated for $\lfloor \Delta \rfloor \propto \lfloor \mathcal{P} \rfloor$ sized regions, the automatic inpainting in the resulting DT can leave small inconsistent areas (see DT WATER). The high Υ values suggest that given area should be inpainted. Due to the Υ distribution, the high and low values are well separated and can be easily detected (i.e. by moving average and standard deviation abruptness). The higher values are sufficient indicator for inpainting. Since analysis and synthesis are divided an adjustment of the number of samples can be made with the negligible computational time.

Dynamic Texture Similarity Criterion

In this chapter, an overview of presented criterion for comparison dynamic textures is presented. At first in the section 6.1 the frequency characteristics and analysis are discussed. Next, in the second section 6.2, the developed criterion model is presented in its most general form. The adaptation of the criterion to the synthesis problem and the inpainting problem are in section 6.3. Finally, the criterion is compared to the psycho-physical test in the last section 6.4 at the end of this chapter.

6.1 Frequency characteristics

There are two typical basic approaches to similarity / recognition / classifying problem in still texture domain. The first ones focus on pixel-based features and characteristics of texture and their clustering and their subsequent comparing. The second approach then focuses on whole texture description as its color, structural, wavelet characteristics etc.

While pixel-based characteristics can be relatively easily extended to more dimensions, the comparable structural approaches development is notably more complex. The fundamental problem in extending pixel-based features is a different property of a temporal dimension in the first place and a close relationship to the importance of spatial relationships. The computing of metrics frame by frame is also inadequate as this approaches calculates similarities only in the spatial dimension and lack the temporal property. Even when computed from the all or several time points (frames) the criterion only approximate a fully dynamical criterion.

Despite these drawbacks, spatial metrics have their place in calculating the likeness of dynamic textures. The color shift in the synthesized result, the occurrence (or vice versa) of noise, and the occurrence of artifacts are cases that are well-detectable by these methods. Without strictly dynamic metrics, however, their use - as the psycho-physical tests (see Sec. 6.4) show - is still insufficient.

Moreover, the criterion should have an ability to adapt to human vision system and take into account the human possibility to recognize and distinguish particular spatial and temporal frequencies.

Contrast Sensitivity To take into account the properties of a human visual system and its sensitivity to contrast the criterion that suppresses frequencies difficult to obtain or distinguish is suitable. As the crucial frequency principle is a binary decision-making process, another method to suppress unwanted and enhanced important frequencies is desired. As the weighting function which well suited our method the contrast sensitivity function (CSF) model of Burbeck and Kelly[14] can be used. The factor can be computed as:

$$CSF(f_s, f_t) = 4\pi^2 f_s f_t \cdot \exp\left(\frac{-4\pi(f_r + 2f_s)}{45.9}\right) \cdot \left(6.1 + 7.3 \left|\log_{10} \frac{f_t}{3f_s}\right|^3\right), \quad (6.1)$$

where f_t is temporal frequency and f_t spatial frequency without any difference in direction.

6.2 General Similarity Model

The proposed DT similarity criterion for dynamic textures is based on the three-dimensional Fourier transformation. As the notion of Fourier transformation properties varies in literature, the short introduction based on defines notation and terminology used in this dissertation thesis follows. The Fourier transformation of a function $f(x_1, x_2, x_3)$ finds the spatial frequencies $\xi = (\xi_1, \xi_2, \xi_3)$. The 3-dimension Fourier transformation for the $f(x_1, x_2, x_3)$ function can be written as:

$$\mathcal{F}f(\xi_1, \xi_2) = \int_{R^n} e^{-2\pi i(x_1 \xi_1 + x_2 \xi_2)} f(x_1, x_2) dx_1, dx_2 \quad (6.2)$$

where the set of all spatial frequencies is called the spectrum. The intensities of one given frequency is called a magnitude $|\mathcal{F}f(\xi_1, \xi_2)|$. The energy spectrum or the power spectrum, is symmetric about the origin because $|\mathcal{F}f(\xi_1, \xi_2)| = |\mathcal{F}f(-\xi_1, -\xi_2)|$. The function $f(x_1, x_2)$ is the intensity of light at each point (x_1, x_2) . Even when we take $x = (x_1, x_2)$ as spatial variable and $\xi = (\xi_1, \xi_2)$ as frequency variable, the frequency values of ξ is still strictly related to spatial dimensions of $f(x_1, x_2)$. Similarly the $\xi = (\xi_1, \xi_2, \xi_3)$ of $f(x_1, x_2, x_3)$ (of f^3 for simplicity) has only temporal meaning only if the vector ξ has direction parallel with temporal axis.

There is a natural geometric interpretation of the zero phase condition which will be often used in this dissertation thesis as important properties of the complex exponential. For a fixed ξ the equations

$$\xi_1 x_1 + \xi_2 x_2 = n \quad (6.3)$$

determine a family of parallel lines in the (x_1, x_2) - plane. Take $n = 0$. Then the condition on x_1 and x_2 is

$$\xi_1 x_1 + \xi_2 x_2 = 0 \quad (6.4)$$

and we recognize this as the equation of a line through the origin with (ξ_1, ξ_2) as a *normal vector* to the line. Then (ξ_1, ξ_2) is a normal to each of the parallel lines in the family.

In higher dimensions the words to describe the harmonics and the spectrum are analogous and due the linear separability - the 3-dimension Fourier transformation 6.2 for the $f(x_1, x_2, x_3)$ function can be rewritten as:

$$\mathcal{F}f(\xi_1, \xi_2, \xi_3) = \int_{\mathbb{R}^3} e^{-2\pi i(x_1\xi_1+x_2\xi_2+x_3\xi_3)} f(x_1, x_2, x_3) dx_1, dx_2, dx_3$$

The harmonics are the complex exponential $e^{\pm 2\pi i x \cdot \xi}$ with n spatial frequencies $\xi = (\xi_1, \xi_2, \dots, \xi_n)$. Again we single out where the complex exponential is equal to 1 (zero phase), which is when $\xi \cdot x$ is an integer.

In three-dimensions a given ξ defines a family $\xi \cdot x = \text{integer}$ of parallel planes (of zero phase) in the (x_1, x_2, x_3) -space. The normal to any of the planes is the vector $\xi = (\xi_1, \xi_2, \xi_3)$ and adjacent planes are a distance $\frac{1}{\|\xi\|}$ apart. The exponential is periodic in the direction ξ with period $\frac{1}{\|\xi\|}$.

Significant Dynamic Texture Frequencies

The combination of 2D and 1D Fourier transformation is used to detect dynamics of significant local and global spatial frequencies. Although there is no pure 3D Fourier transformation, due to linear separability resulting values respond to 3D Fourier transformation. Note that spectral dimension data are solved multiple over the entire detection, and therefore 4D dynamic texture is taken as an n-tuple of 3D monospectral dynamic textures.

Whereas the crucial part for video similarity perception is its structures, dynamic behavior[40] the key principle of the presented criterion is to detect spatially significant frequencies ξ from the set of f^2 and observe their behaviour by another Fourier analysis eq. (6.2). This behavior is mainly described by a complex exponential with the normal ξ with dominant temporal component ξ_3 . Components ξ_3 similar to $\xi_{1,2}$ are less significant due to combining spatial and temporal directions.

Because of the separability of the individual vectors information and for simplicity of computational model the harmonics with unit normal vectors $\hat{\xi}$ are projected to ξ^t with only temporal dimension. In other words, the any given $\hat{\xi}$ is represented by nearest $\hat{\xi} \parallel t$, $t = (0, 0, 1)$, where the distance d between (ξ_1, ξ_2, ξ_3) and $(\xi_1^t, \xi_2^t, \xi_3^t)$ is defined as standard L_2 norm in \mathbb{R}^3 space:

$$d(\xi, \xi^t) = \|\xi - \xi^t\| = \sqrt{(\xi_1^t - \xi_1)^2 + (\xi_2^t - \xi_2)^2 + (\xi_3^t - \xi_3)^2} \quad (6.5)$$

The set of all unit vectors $\hat{\xi}$ are denoted as $\hat{\Xi}$. The $\hat{\Xi}^t$ then denotes all unit vectors $\hat{\xi} \parallel t$, $t = (0, 0, 1)$. In order to try to separate the key frequencies

$$\tilde{\xi} \in \tilde{\Xi}$$

from the information unnecessary for the similarity finding process, it is possible to neglect certain spatial frequencies ξ . It is obvious that the noise frequencies are not necessary, but moreover, there are many other which can be omitted with maintaining the criterion efficiency. The group of spatial frequencies

$$\Xi_g = (\xi_1, \xi_2, \dots, \xi_n)$$

can be roughly represent by the centroid vector ξ_c , or more efficiently by $\tilde{\xi}$ where the $|\mathcal{F}f(\xi_1, \xi_2)|$ is maximized thought all grouped cluster vectors.

Provided there is a group of representative normal (unit) vectors $\hat{\xi}$ we assume a new harmonic $\tilde{\xi}$. Note that obviously the

$$\tilde{\Xi} \subset \tilde{\Xi}^t.$$

This strongly simplifies the criterion evaluation with negligible loss of generality.

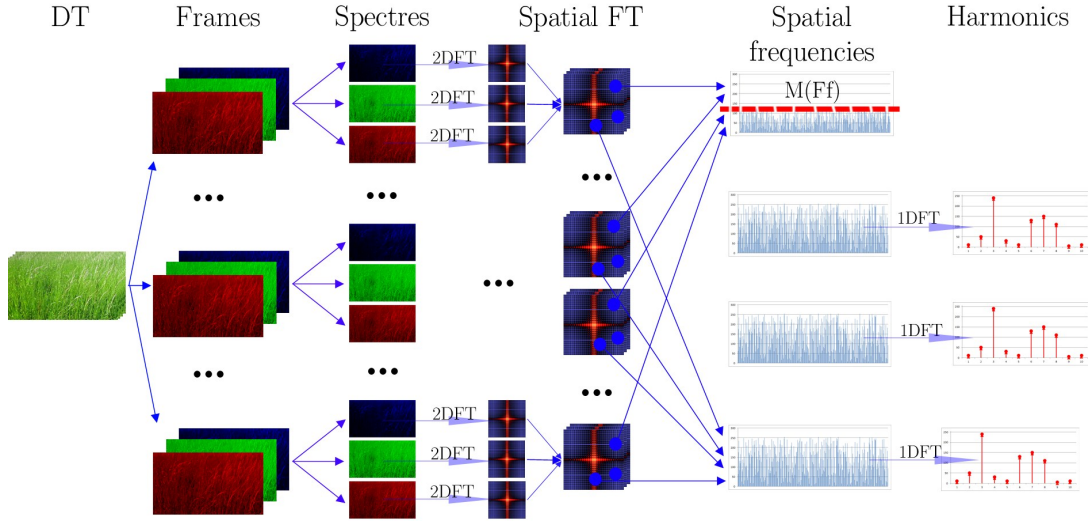
The crucial frequency

Whereas the harmonics set expresses the behavior of one spacial frequency alongside temporal dimension, it is essential to use only those harmonics that carry essential visual information ($\tilde{\xi}$). Only if the spatial frequency temporal behavior can be considered as significant, it should be used in the criterion. Note that the significant frequencies designation characterises the given frequencies through *all* the compared DTs (i.e., Y_s and Y_c). In other words, when a specific spatial frequency shows high magnitude values and is therefore crucial for the reconstruction of a given dynamic texture, this particular frequency and its time behavior are used for comparison.

Of course, not only the $\tilde{\xi}$ harmonics carry the necessary information. One can admit that even information about *non-occurring* spatial frequencies can carry a considerable amount of information, but with respect to the general aspect of image processing and it Fourier transformation usage, we consider this information as a complement to strong frequencies.

There are many problems in crucial frequency determination: How to separate crucial and inessential frequency? What values of magnitude should be included in these groups? Should be the priority the spatial frequencies distribution or magnitude values maximization? Also, how many harmonics are sufficient to represent the dynamic texture for the comparison?

For now let's just suppose that there is significantly smaller number of $\tilde{\xi} \ll MN \approx w \log_2(MN)$. For more detailed analysis and testing, see next chapter.

Figure 6.1: *The similarity criterion computation flowchart.*

6.3 Criterion model

The DT similarity criterion is based on the comparison of harmonic frequencies time series related to significant spatial frequencies in the compared dynamic textures (Figure 6.1). Let's denote a set of harmonics (of Y_s or Y_c DT) $\xi_Y^{\zeta, \bullet}$ for δ spectral band:

$$\xi_Y^{\zeta, \bullet} = \begin{cases} \int_{R^1} \exp\{-2\pi i x_3 \xi_3\} \mathcal{F}\{f^\delta(\xi_1, \xi_2)\} dx_3, & |\mathcal{F}\{f(\xi_1, \xi_2)\}| \in \tilde{\Xi} \\ 0, & |\mathcal{F}\{f(\xi_1, \xi_2)\}| \notin \tilde{\Xi} \end{cases} \quad (6.6)$$

where $\tilde{\Xi}$ is defined as $w_f \log_2(NM)$ -th largest vales from $Y_c \cup Y_s$. \bullet denotes all corresponding index values (e.g., temporal frames) and $\zeta = (\xi_1, \xi_2)$ is some fixed spatial frequencies ξ_1, ξ_2 . $w_f \geq 1$ is the parameter set to speed up the process and to better separate the noise from crucial spatial frequencies. We set experimentally this value to $w_f = 2$ to 3. Recall that due to steadily growing audiovisual data resolution, this step sharply reduces the criterion of computational time complexity. From now, takes an illustrative example in Figure 6.2 of 10 computed crucial frequencies (horizontal axis) with magnitudes from 0 to 255 (vertical axis) from original and synthesized dynamic texture.

Whereas the harmonics set expresses the behavior of one spacial frequency alongside temporal dimension, it is essential to use only those harmonics that carry essential visual information. If the spatial frequency temporal behavior can

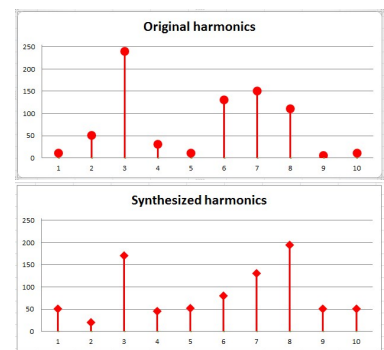


Figure 6.2: *Harmonics example:* Original and synthesized frequencies illustration. The vertical axis denotes magnitudes; the horizontal axis picked crucial harmonics.

be considered significant, it is used in the criterion. Note that the significant frequencies designation characterizes the given frequencies through all the compared DTs (i.e., Y_s and Y_c). This detection ensures that the missing or modified frequencies in the synthesized texture are detected and evaluated.

Now with the rough knowing of $\tilde{\xi}$ characterisation let us denote general function with a measure L between two DTs Y_s and Y_c as:

$$\uparrow \Theta_{Y_s, Y_c}^L = \sum_{\forall \delta} \sum_{\forall \xi_1 \in Y_s^\delta \cap Y_c^\delta} \sum_{\forall \xi_2 \in Y_s^\delta \cap Y_c^\delta} \theta_{Y_s^\delta, Y_c^\delta}^{\xi_1, \xi_2}, \in \langle -NM\delta\beta; 0 \rangle, \quad (6.7)$$

where \uparrow is the desired value orientation, β is the Fourier transformation maximum, $\theta_{Y_s^\delta, Y_c^\delta}^{\xi_1, \xi_2}$ for a δ spectral band is:

$$\theta_{Y_s^\delta, Y_c^\delta}^{\xi_1, \xi_2} = -M_{x_3}^\delta(\mathcal{F}\{f_{x_3}(\zeta)\}) |L(\xi_{Y_s^\delta}^{\zeta, \bullet}, \xi_{Y_c^\delta}^{\zeta, \bullet})|, \quad (6.8)$$

where Y_s and Y_c are source and comparison DTs, respectively. $L(u, v)$ is the measure between impulse characteristic u and v (a distance between two harmonics sets) like eq. (6.12), (6.13) or (6.14). $M_{x_3}(\gamma)$ is a function which return the maximal value for the argument vector γ and scale value to the range $\langle 0; 1 \rangle$ over all frame's maxima. Θ_{Y_s, Y_c} is a set of all measure values $\theta_{Y_s, Y_c}^{\bullet, \bullet}$ between DTs Y_s, Y_c across all spatial frequencies. Physical meaning of $\theta_{s, c}^{\xi_1, \xi_2}$ represents different dynamics of the spatial frequency $\xi = (\xi_1, \xi_2)$, i.e., periodicity and magnitude variability between textures Y_s and Y_c . The M function could be substituted by a different function that reproduces human visual sensitivity as i.e. $CSF(\zeta, x_3)$.

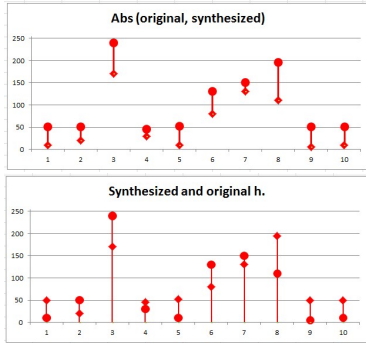


Figure 6.3: *Raw frequencies example*: difference and absolute values (6.10) illustrated. Vertical axis denotes magnitudes, the horizontal axis picked harmonics.

Emphasize that the ${}^L\theta_{i, j}^{Y_s, Y_c}$ is taken as negative to preserve the definition of *similarity* meaning of criterion and therefore the maximization criteria. For the purpose of comparing the effectivity of $\tilde{\xi}$ determination the equation 6.7 for whole $\hat{\Xi}$ set of appropriate harmonics.

The harmonics $\tilde{\xi}$ or more precisely the distance function values between them are then weighted by the function $M()$ which correspond to human perception and further emphasize the substantial difference frequencies.

The M calculation optimization

Understanding what frequencies are crucial for texture comparison, of course, should not be taken simply from a statistical point of view. The order of statistical and human perception criterion enhancements is questionable. The alternative description of equation 6.6, which could look as:

$$\theta_{Y_s, Y_c}^{i, j} = -|L^*(M_\zeta(\xi_{Y_s}^{\zeta, \bullet}), M_\zeta(\xi_{Y_c}^{\zeta, \bullet}))|. \quad (6.9)$$

Clearly the explicit version of the perception enhancement in eq. 6.6 involve less computational time. Again, for now just suppose that M is function takes value from 0 to 1 with relation to human perception of particular time / space frequency. Moreover, the $CSF(\zeta, t)$ (6.1) can be used to emphasize the visually important frequencies.

For the most general case (Figure 6.3) of computing dynamics similarity between two arbitrary textures the most simplest form of metrics has form of:

$$L^\sigma(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) = \sum_{\forall \xi_3 \in Y_s \cap Y_c} l^\sigma(\xi_{Y_s}^{\zeta, \xi_3}, \xi_{Y_c}^{\zeta, \xi_3}) , \quad (6.10)$$

$$l^\sigma(s_i, c_i) = |c_i - s_i| \quad (6.11)$$

where c_i and s_i are i -th values from the given harmonics ξ . This effortless distance has an advantage of simplicity, metrics definition and above all physical significance. Many textural metrics (let us mention them, for example, STSIM) have poor or no exact physical relationship.

The physical meaning of $\theta_{\xi_1, \xi_2}^{s,c}$ without weighting represents pure different dynamics of the spatial frequency $\xi = (\xi_1, \xi_2)$, i.e., periodicity and magnitude variability between textures Y_s and Y_c . The values are directly dependent on image representation and in the most usual and general are from -255 to 0 , or in the little readable form $-\theta_{\xi_1, \xi_2}^{s,c}$ from 0 to 255 , where the value has physical meaning of the sum of differences between harmonics set per impulse. The any given value of the distance can be directly interpreted.

Another clear advantage is the probability distribution of values directly related to the input textures and thus the whole range of values.

Lost frequencies

When the eq. 6.10 interprets the harmonics distance in its purest way as the absolute difference; there are another and more interesting ways to distinguish the dynamics that can occur in the compared dynamic textures.

For the purpose of comparing dynamic textures where one was created by a modification of the other and thus it shares major common part of the criterion function, some additional information can be obtained.

Let us assume that we have two dynamics textures - the synthesized dynamic texture Y_s and original source texture Y_o . The dynamic texture Y_s was created by modifying the original DT Y_o .

The loss of information between the synthesized dynamic texture Y_s and the original dynamic texture Y_o can be expressed concerning dynamics as the absence of some spatial/temporal frequencies (see Figure 6.4) which thy synthesizing / modeling approach is not able to articulate. This

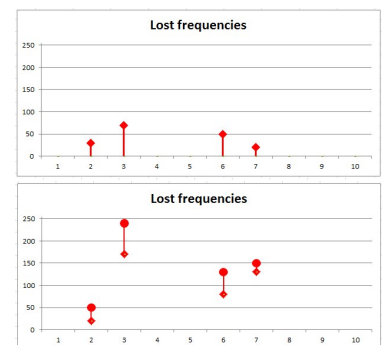


Figure 6.4: *Lost frequencies example*: Difference of absolute values (6.12)

particular case can be caused, for example, by inappropriate sample or insufficiently learned the model and is very common. The HPS consequence of this is a loss of structure, missing sharp edges and details, as well as low frequencies that DT synthesizing methods are unable to represent.

The lost frequencies, both spatial and temporal, term is:

$$L^{lost}(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) = \sum_{\forall \xi_3 \in Y_s \cap Y_c} l^l(\xi_{Y_s}^{\zeta, \xi_3}, \xi_{Y_c}^{\zeta, \xi_3}), \quad (6.12)$$

$$l^l(s_i, c_i) = \begin{cases} s_i - c_i & \text{if } s_i \geq c_i \\ 0, & \text{otherwise} \end{cases},$$

where s_i (original) and c_i (compared) are i -th values from the given harmonics ξ .

False frequencies

However, due to synthesis or inpainting methods, some other phenomena that reduce visual quality of the synthesized DTs exist. The problem is not the missing information, but false frequencies introduced by these methods and not related to the original texture. False synthetic DT periodicity is visually disturbing the perception of the quality of DTs. This is the apparent repetition of the same samples as well as the visible boundaries between data samples from which the new texture composed.

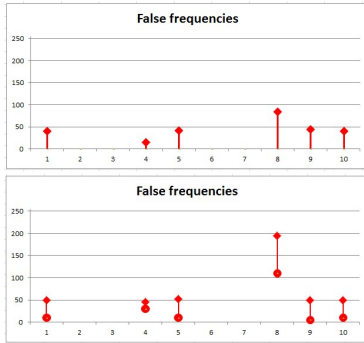


Figure 6.5: *False frequencies example*: difference and absolute values illustrated. Vertical axis denotes magnitudes, the horizontal axis picked harmonics.

Problem is not always the frequency variation made by the method. The missing information computed as $L^{lost}(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet})$ is not the only problem that can occur.

The missing information problem is often directly caused by the used method and not by the input dynamic texture. That approach determined phenomena reduce visual quality of the synthesized dynamic textures and are manifested as false edges, step color changes, etc. Often these phenomena may be described as a violation of G^0 or C^0 continuity in both spatial and temporal dimensions.

The problem is therefore not the missing information, but false frequencies (see Figure 6.5) introduced by these methods and not related to the original texture. Usually, this method determined unfaithfulness information are manifest themselves as false periodicity and homogeneity violation.

In relation to the name of the equation (6.12), it can be named as *false frequencies of approach determined frequencies*. For the sake of simplicity (and as a complement to eq. (6.12) the first term will be used here.

The false frequencies, both spatial and temporal, term is:

$$L^{false}(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) = \sum_{\forall \xi_3 \in Y_s \cap Y_c} l^f(\xi_{Y_s}^{\zeta, \xi_3}, \xi_{Y_c}^{\zeta, \xi_3}) , \quad (6.13)$$

$$l^f(s_i, c_i) = \begin{cases} c_i - s_i & \text{if } c_i \geq s_i \\ 0, & \text{otherwise} \end{cases} .$$

Here it is good to remind that the newly created false frequencies are harmonics $e^{\pm 2\pi i x \xi}$ with normal $\xi = (\xi_1, \xi_2, \xi_3)$ where $\xi_{(1,2,3)}$ varies to any combination as the patches are R^n space with noticeable error (which are usually minimized by the synthesizing method). Thus $L^{false}(s, c)$ can be efficiently computed on any harmonics $\xi_Y^{x_1, x_2, x_3}$ from which we, for simplicity and to streamline the computation process, pick $\xi_Y^{\zeta, \bullet}$ and $\xi_Y^{x_1, x_2, x_3}$ where $x_{(1,2,3)}$ are maximized mutual value difference (i.e., ξ yielding an oblique harmonics).

In the most frequent case the false repetition is caused by an inappropriate patch (which are usually easy to recognizable), unstable or inadequately learned synthesis model, etc.

The inadequate tile size or using of only one patch is the most obvious example: If there is only one patch (even tileable one), the repeating pattern could be easily noticeable even if there are no visually recognizable patch border which could which would reduce the visual quality of the result. Even when the repetition is due to texture stochasticity hard to observe in image space, the peaks are in Fourier frequency space are - comparing to original texture - noticeably easier to observe.

Combined measure

Both methods (6.12), (6.13) complement the absolute difference in texture behavior and allow more accurate behavior comparison of different DT synthesis approaches and more detailed analysis of their weaknesses and strengths.

In the proposed criterion we use their combination:

$$L^\alpha(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) = \alpha_l L^{lost}(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) + \alpha_f L^{false}(\xi_{Y_s}^{\zeta, \bullet}, \xi_{Y_c}^{\zeta, \bullet}) , \quad (6.14)$$

where α_l and α_f are weight parameters between 0 and 1. After our extensive tests, it seems that for the purpose of computed similarity of DTs the values of $\alpha_l = \alpha_f = 1$ are sufficient. For applications comparing the original and modified (synthesized or inpainted) textures the parameter α_l seems to be more important and determining, thus $\alpha_l \leq \alpha_f$.

6.4 Psycho-Physical Test

Whereas quality criteria try to emulate human perceptions, it is necessary to verify them. Thus any computed similarity criteria must be compared to human perception. Unfortunately, this requires tedious psycho-physical experiments.

The psycho-physical test should be ideally performed in a controlled environment under the supervision of the presenter. Of course, the other form of testing could be done - i.e. testing on participants own PC in their similarly as the test performed by Haindl and Kudělka[58]¹. In these cases, of course, the different variants of the environment (surrounding objects, lighting, etc.), hardware, and much more needs to be taken into account.

Because the DT compression is one of the critical factors for DT inpainting test, it is optimal to perform a test on uncompressed data. Unfortunately, the data amount produced by current resolution in uncompress form are difficult to handle in a remote test. The use of compression methods leads to blurring of sharp edges (both edges in the image and especially edges between the original and synthetic parts of DT). The similar effect occurs in downsampling the resolution. The using of compressed or downsampled dynamic texture for test leads to biased evaluations, often deflected in favour of modified data, in which the adjustment is less pronounced.

For our tests, participants should always compare the data in maximum quality and resolution undiminished. For comparing with another method, dynamic textures resolution² will be downsampled from full resolution to reproduce the similar artifacts and texture quality or alternative methods.

For a proper comparison of inpainting methods, two test with a different setup is suitable. First suited to evaluate the ability to determine the border and the second test with a goal to rank the inpainted result:

Blind test For blind tests, participants should rate the displayed dynamic texture by its visual quality and whether it contains or does not contain obvious interference, retouch or artifacts. The used scale from 0 to 5, is suitable and sufficiently detailed. The maximal values 5 denotes the highest quality of the presented DTs. As a control mechanism, the original dynamic textures without any editing or inpainting is appropriate to be added to the evaluating process.

Informed test In the informed tests, the same set of dynamic textures should be used. Unlike the blind test, additional information should be presented to participants. This information consists of original dynamic textures and the corresponding masks thus participants can directly compare changes in the presented results.

¹Texture fidelity benchmark: (<http://tfa.utia.cas.cz>)

²which are very often downsampled - i.e. from 720×576 to 174×130 or published in loss formats such as MPEG



Figure 6.6: *Informed psycho-physical tests arrangement* for DT BOAT: The source data - used masks and original textures - are in the left column in the top to down order. The two possible output of an inpainting algorithm is showed in the middle and right column.

To obtain a direct comparing ranking, two dynamic textures result can be simultaneously presented. If results from more inpainting methods exist, the all pair combinations should be presented and evaluated. To ensure the blindness of the test, the presented dynamic texture pairs should be randomly placed on the left / right sides. The designed arrangement can be seen in Figure 6.6.

Dynamic BTF

The bidirectional texture function is currently the best visual texture representation of various textured materials which can be simultaneously modeled and acquired. The process of BTF measuring is time consumable, but the results are currently the most high-end and physically correct surface materials appearance modeling. Above all, the adaptability of the model to the illumination conditions is extremely wide. This short chapter consists of a BTF and DT model combination demonstration (see Figure 7.1).

The short BTF model property overview is in the first section 7.1, while the dynamic bidirectional texture function and using of a proposed toroidal model editing capabilities to synthesize is discussed in the second section 7.2.



Figure 7.1: *DBTF*: Examples of a DBTF texture, different frames from the same sequence with varying illumination presented.

7.1 Bidirectional Texture Function

The BTF model is a 7-dimensional model, subspace of a more general BSSRDF model. Remind, that the 7-dimensional multispectral function is defined as:

$$Y_r^{BTF} = BTF(\varsigma, x, y, \theta_i, \phi_i, \theta_v, \phi_v) , \quad (7.1)$$

where ς is spectral index, x and y are planar coordinates. $\omega_i = [\theta_i, \phi_i]$ is illumination direction and $\omega_v = [\theta_v, \phi_v]$ is viewing direction. Y denote a random field (true unobservable image) for pixel r .

The BTF data acquiring is a lengthy, complex and complicated process resulting in a huge data set. For example, single texture measurement on the University of Bonn setup consists of 6561 (a combination of 81 view (ω_v) and 81 illumination (ω_i) source position) multispectral images.

7.2 Dynamic Bidirectional Texture Function

For the desired DBTF model the BTF can be then taken as generalised DBTF model with the assumption of time dimension as a constant. The whole BTF image space is then subspace of DBTF model:

$$Y^{BTF} = Y_{\bullet, \bullet, \bullet, \bullet, \bullet, \bullet, \bullet}^{DBTF} \quad (7.2)$$

where t denotes an observing time. The whole dynamic BTF model is then defined as:

$$DBTF(\varsigma_i, x, y, \theta_i, \phi_i, \theta_v, \phi_v, t), \quad (7.3)$$

with a clear relationship to BTF model:

$$BTF(\varsigma, x, y, \theta_i, \phi_i, \theta_v, \phi_v) = DBTF(\varsigma_i, x, y, \theta_i, \phi_i, \theta_v, \phi_v, \emptyset). \quad (7.4)$$

where \emptyset denote missing index. Recall, that the dynamic texture is in its simplest form defined as:

$$Y_r^{DT} = DT(\varsigma_i, x, y, t) , \quad (7.5)$$

Similarly, the direct extension of a BTF model (1.5) to temporal domain as combination of BTF and DT model leads, due to compliance parameters ς, x, y that are shared by both models in the exact same meaning, to the resulting DBTF 8-dimensional model which can be suited as:

$$DBTF(\varsigma_i, x, y, \theta_i, \phi_i, \theta_v, \phi_v, t) , \quad (7.6)$$

where ς is spectral index, x and y are planar coordinates. $\omega_i = [\theta_i, \phi_i]$ is illumination direction and $\omega_v = [\theta_v, \phi_v]$ is viewing direction. The t parameter is again a newly added temporal dimension index.

The measures are then stored as the set of classical rectified original measurement (frames) with possibility to convert them into set of apparent BRDF: $ABRDF_{[r,t]}(\theta_i, \phi_i, \theta_v, \phi_v)$.

As this dissertation thesis recommends to address dynamic texture as 5-dimensional function with specifically designated dynamic dimension as a representation of high-order texture inner dynamics, the function 7.9 becomes to:

$$DBTF(\varsigma, x, y, \theta_i, \phi_i, \theta_v, \phi_v, t, \varrho_d) , \quad (7.7)$$

where vector ϱ denotes the appropriate dynamic, namely, i.e. optical flow vectors.

DBTF Measuring

The crucial problem of acquiring a suitable DBTF data is, of course, demanding measurement. The BTF measuring itself is a very complex process which is furthermore complicated due to an assumption of a homogenous dynamic process.

The inability of a natural dynamic texture deterministic behaviour through all measurements complicates the whole data obtaining process as the dynamics is usually supplied artificially by any sort of a proper device (of course, the artificial dynamic texture with deterministic behaviour is also possible, but usually is not our object of interest).

Inasmuch as the measurement is an iterative process, the sample is *different* for every measure. Moreover, every measurement takes a significant time. The possibility of obtaining more measurement simultaneously is theoretically possible, but extremely complicated (currently, BRDF simultaneous measuring is possible).

The problem of varying subject form, however, is only apparent as the texture is defined as the random realisation of a stochastic field with unvarying parameters. So, with the textural assumptions applied to dynamic texture temporal dimension the samples of a dynamic texture (and thus even dynamic BTF) are defined as the same texture (even different realisation) as the dynamic texture are seen as generative video model defined by a random processed with some observed variable and hidden variables[27].

Therefore, if an (artificial) dynamics has constant properties through whole measuring process, all samples, even obtained in the different time, can be considered as one dynamic texture.

Interestingly, the constant properties of a dynamic process source can be defined as a hidden \mathcal{U} process, which defines the texture inner dynamics. The general multidimensional function \mathcal{U} thus generate the texture dynamics:

$$BTF(\varsigma_i, x, y, \theta_i, \phi_i, \theta_v, \phi_v) \xrightarrow{\mathcal{U}(\kappa)} DBTF(\varsigma_i, x, y, \theta_i, \phi_i, \theta_v, \phi_v, t, \varrho_i) , \quad (7.8)$$

The dynamic BTF whose dynamic properties vary but has the same structural data can be therefore defined in a more general way as varying dynamic DBTF:

$$VDBTF(\varsigma, x, y, \theta_i, \phi_i, \theta_v, \phi_v, t, \varrho_i, \mathcal{U}(\delta_i)) . \quad (7.9)$$

The general dynamic generative function \mathcal{U} with hidden vector parameter κ_i is then, i.e. fan wind and δ the direction, wind strength etc. The \mathcal{U} function can then define the high order and long-term dynamics property and can be understood as generative and determining function for dynamic texture transition function in temporal meaning. For example, temporal transition textures between two stable states (e.g. strong wind and weak wing, or raining process) can be modeled this way.

Approximate DBTF modeling

Whereas a moderate precision measurement of one single material made by controlled robotic arms with camera and light source can take around one terabyte, the amount of a dynamic material represented by a dynamic BTF, therefore, increases significantly. Even if the temporal dimension length can usually be noticeably smaller than a spatial resolution, the amount of data is considerably tremendous.

The amount of data and problematic comprehensive measurement leads to use an approximate model, in which only noticeably smaller subset of a ω_i and ω_v are used.

Moreover, the BTF measuring problem caused by self-occluding is even more significant in dynamic materials. Especially in the case of extreme low angles of view, the distortion of the given dynamic texture is due to a non-planar material plane problematic and excludes the simple approximation of values.

The using of additional information as a measured and registered depth map can be used to produce a slightly better result, i.e. by utilizing the displace mapping algorithm. The depth map could be used as an additional spectral plane to archive better suited toroidal tile and patch. The methods that blur the sharp edge can be utilized.

DBTF modeling

The utilizing of a proposed toroidal model to a DBTF material is similar to a BTF synthesizing process and DT editing process. The set of measured dynamic textures are labelled by with appropriate ω_i and ω_v . The optimal triple toroidal tile is found in a primary dynamic texture (say sample with $\omega_v = [0^\circ, 0^\circ]$ and $\omega_i = [45^\circ, 0^\circ]$) and propagated to secondary measures or can be computed as a compromise solution. The appropriate sample is then applied to a desired location in the rendered scene.

The problem occurs when the scene illuminance r observing direction is varying. The temporal length of a used triple toroidal patches limits the angular speed of the light source and even the viewing direction. The angular speed

$$\omega = \frac{d\phi}{dt} \quad (7.10)$$

with respect to the texture plane is directly limited by a sample density and patches length, t denotes time and ϕ an angle. As the only sparse measurements are performed the maximal illumination and viewing angle changes are defined as:

$$\omega \approx g \cdot [\mathcal{P}]_t, \quad g \in \mathbb{N}. \quad (7.11)$$

If the ω are not linear dependent to a $[\mathcal{P}]_t$ value, the dynamic texture transition is inaccurate. Obviously, the visualization expected error measure can be described as:

$$\epsilon_\omega = ||\omega| - g \cdot [\mathcal{P}]_t|, \quad g \cdot [\mathcal{P}]_t < \omega < (g + 1) \cdot [\mathcal{P}]_t, \quad g \in \mathbb{N} \quad (7.12)$$

and should be minimized.

The simple example of modeling a dynamic texture measured under varying illumination condition and arrangement from measured texture patches is demonstrated in Figure 7.2.

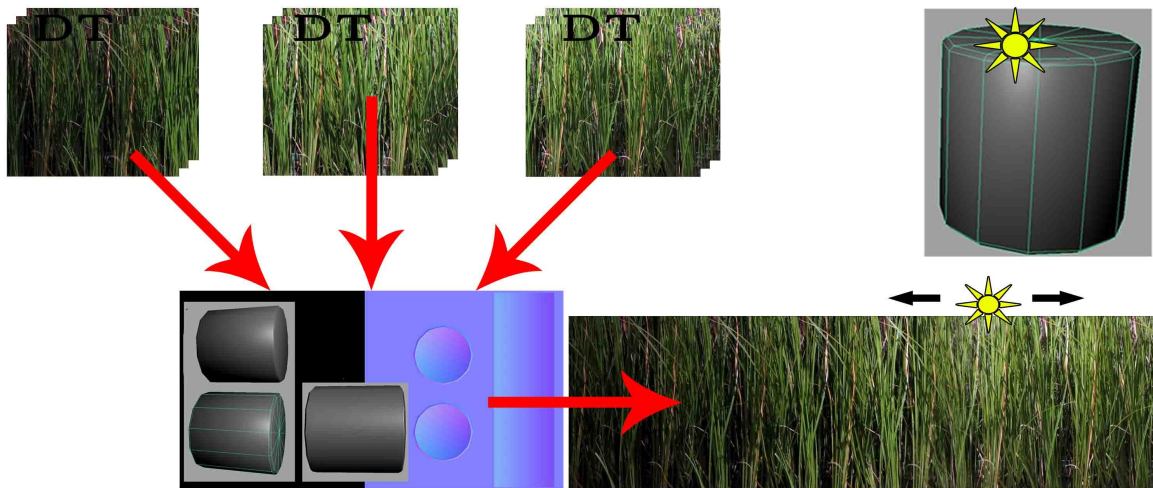


Figure 7.2: *DBTF assembly illustration*: Illustration of applying a dynamic texture to a cylindrical object. The three dynamic texture samples obtained in the different illumination are put together to cover the object. The normal maps illustrate the angle to the illumination source.

Dynamic Texture Database and Criterion Validation

This chapter consist of two main parts both closely links to previously presented inpainting method and dynamic texture similarity criterion. At first section 8.1, the created dynamic texture database dismantled with a comparison to existing one, namely and primary DynTex database is presented. Next, in section 8.2, the database is used to compare results and validate the Spatiotemporal Fourier Transformation Criterion which was presented in the previous chapter.

8.1 Dynamic Texture Database Subset

As an excellent, inspiring material, the widespread and often used DynTex[104] database class labels could be used. The database offers three classification benchmarks dataset of dynamic textures with various length measured under varying conditions including camera span, zoom, and different weather but unfortunately only in PAL resolution. Most DynTex textures are in three occurrences per each class, and current database contains over 650 sequences while covering the wide texture classes as flowers, water, fountain, foliage, flags and many others. For used datasets the classes can be seen in Table 8.1:

Table 8.1: *DynTex database classes*

Dataset	Classes				
Alpha	SEA	GRASS	TREES		
Beta	SEA	VEGETATION	TREES	FLAGS	CALM WATER
	FOUNTAIN	SMOKE	ESCALATOR	TRAFFIC	ROTATION
Gamma	FLOWERS	SEA	NAKED TREE	FOLIAGE	ESCALATOR
	CALM WATER	FLAGS	GRASS	TRAFFIC	FOUTAINS



Figure 8.1: *Class* CALM WATER



Figure 8.2: *Class* FLOWER

The illustration of DT types CALM WATER and FLOWER can be seen in Figure 8.1 and Figure 8.2. Usually, every class contains tens of textures. All three datasets consist of almost 650 textures with various lengths, camera moves, or levels of detail.

Database Description For our database, we consider a setting which is considered optimal for the presented method test but can be used in many other approaches. The basic technical settings parameters are:

- resolution at least FullHD (1920×1080),
- using of a static camera (no handshake),
- no camera move,
- manually fixed focus,
- manually fixed exposure and ISO,

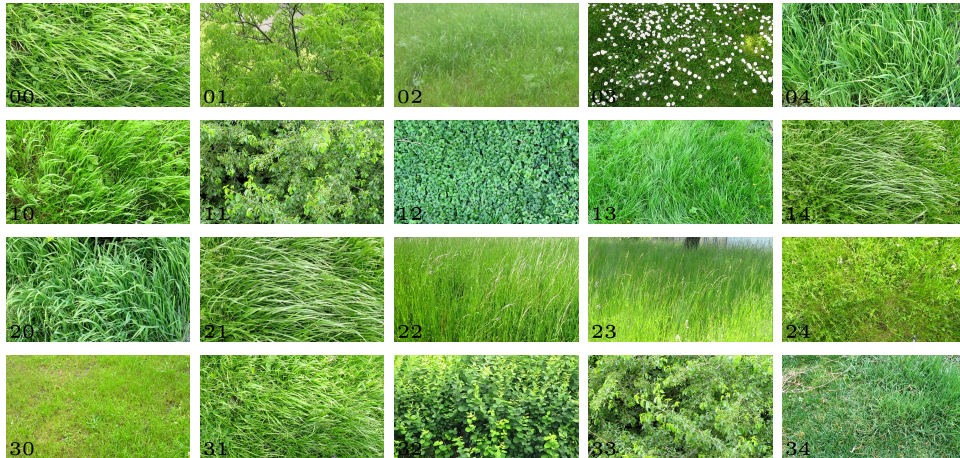


Figure 8.3: *Representative selected frames from several GRASS*

- manual color balance,

while the main reason is to measure the maximal amount of data with the same parameters.

For the validating purpose and to test of presented inpainting and synthesizing method it is optimal to satisfy not only given technical parameters but to focus on parameters of dynamic textures themselves. The more shots of the same dynamic textures, more measures with different dynamics and luminance etc. is then needed. Varying parameters allow for the extensive test of various modeling methods and its robustness.

The other appropriate parameters for build database should consist of:

- more measures of one particular dynamic texture,
- sufficient distance to take into account the specific texture dynamics,
- the dataset should consist of visually similar textures, yet with different dynamics
- the database should contain textures with pronounced (strong wind, rain) and low dynamics as well as textures with similar dynamics (directional wind) but very different structure.

Figure 8.3 illustrates representative measured subset of types GRASS and FOLIAGE DTs. Note i.e. DT 11 and 33 as a textures with similar structure, but very different dynamics. Our whole database consist of 1751 dynamic textures with duration of from twenty second at least one minute.

8.2 Spatio-Temporal Fourier Transformation Criterion Validation

The dynamic texture database and its classes were used to spatio-temporal Fourier transformation criterion presented in the previous chapter. It is clear that just like a slightly edited dynamic texture (i.e. by inpainting) should be very similar to the original dynamic texture, the value of similarity of textures belonging to the same class should be noticeably higher than for textures outside this class. So the ratio between inter-class and intra-class similarities could be used to cross-validate the criterion definition (or alternatively to determine the class consistency). Again note, that for the purpose of this dissertation thesis the primary and crucial property of a dynamic texture is its dynamics.

Lets remind that the similarity based on spatio-temporal Fourier transformation criterion between two given dynamic texture Y_s and Y_c with specific metric L is denoted as:

$$\Theta_{Y_s, Y_c}^L ,$$

and similarly, with notation of any given dynamic texture class (i.e. FOUNTAIN) as $^F Y$, the similarity between two dynamic texture class A and B expressed as simple sum of all particular dynamic texture similarities:

$$\Theta_{A\mathcal{Y}, B\mathcal{Y}}^L = \frac{1}{|A\mathcal{Y}||B\mathcal{Y}|} \sum_{\forall i \in A\mathcal{Y}} \sum_{\forall j \in B\mathcal{Y}} L\theta^{A\mathcal{Y}_i, B\mathcal{Y}_j} , \quad (8.1)$$

where $|A\mathcal{Y}|$ and $|B\mathcal{Y}|$ denotes number of a texture in classes A and B.

Recall, that the scrips notation are organised for any function α as

$$\underset{space}{method\ label} \alpha_{key}^{metric} ,$$

where *metric* is used metrics between harmonics sets, *space* is set of all used DT classes, *method label* usually denotes what is measured / computed and *key* is the particular relevant dynamic texture. The small greek case and upper greek case note if a key is one texture or set o textures.

More comprehensive with using of \bullet notation the similarity between classes could be written down simply as:

$$\Theta_{A\mathcal{Y}, B\mathcal{Y}}^L = \frac{\sum_{\bullet} \theta_{A\mathcal{Y}_{\bullet}, B\mathcal{Y}_{\bullet}}^L}{|A\mathcal{Y}||B\mathcal{Y}|} . \quad (8.2)$$

The weighted value of the intra-class similarity is then

$$\overset{intra}{\mathcal{Y}} \Theta_{A\mathcal{Y}_s}^L = \Theta_{A\mathcal{Y}_s, A\mathcal{Y}_{\bullet \setminus s}}^L = \frac{1}{|A\mathcal{Y}|} \sum_{\bullet} \theta_{A\mathcal{Y}_s, A\mathcal{Y}_{\bullet \setminus s}}^L . \quad (8.3)$$

Since \bullet denote all possible value of given variable or all elements of a set, $\bullet Y$ denotes a set of all DT classes and $\bullet \setminus A Y$ a set of all DTs classes other than A. Consequently $\bullet \setminus A Y_{\bullet}$.

denotes all DTs from all classes excluding A. The sum of similarities of one DT $^A Y_s$ from given class A to all DTs in all other classes is then the inter-class similarities is:

$${}^{inter}_y \Theta_{^A Y_s}^L = \Theta_{^A Y_s, \bullet \setminus ^A Y_\bullet}^L = \frac{1}{|\bullet \setminus ^A Y_\bullet|} \sum_{\bullet} \theta_{^A Y_s, \bullet \setminus ^A Y_\bullet}^L . \quad (8.4)$$

The ratio of similarity values between one particular DT $^A Y_x$ from a given class labeled by A to all other DTs in every class (for simplicity suppose the same class size) is then ratio between intra-DT class similarity values and inter-DT class similarity values:

$${}^{ratio}_{\bullet Y} \psi_{^A Y_s}^L = \psi_{^A Y_s, \bullet \setminus ^A Y_\bullet}^L = \frac{|^A Y_\bullet| - 1}{(|\bullet Y| - 1)|^A Y_\bullet|} \frac{\sum_{\bullet} \Theta_{^A Y_s, \bullet \setminus ^A Y_\bullet}^L}{\sum_{\bullet} \Theta_{^A Y_s, ^A Y_\bullet}^L} . \quad (8.5)$$

The overall inter-class average ratio can be subsequently express as:

$${}^{overall} \Psi_{^A Y} = \dot{\Psi}_{^A Y} = \Psi_{^A Y, \bullet \setminus ^A Y_\bullet} = \frac{1}{|^A Y|} \sum_{\forall s \in ^A Y} {}^{ratio}_y \psi_{^A Y_s}^L = \frac{1}{|^A Y|} \sum_{\bullet} {}^{ratio}_y \psi_{^A Y_\bullet}^L , \quad (8.6)$$

or precisely, but more cryptic with two-level $\bullet\bullet$ notation as:

$$\dot{\Psi}_{^A Y} = \Psi_{^A Y}^{overall} = \sum_{\bullet} {}^L_y \psi_{^A Y_\bullet}^{ratio} = \frac{1}{^A Y} \sum_{\bullet\bullet} \frac{|^A Y_\bullet| - 1}{(|\bullet Y| - 1)|^A Y_\bullet|} \frac{\sum_{\bullet} {}^L \theta_{^A Y_\bullet, \bullet \setminus ^A Y_\bullet}}{\sum_{\bullet} {}^L \theta_{^A Y_\bullet, ^A Y_\bullet}} . \quad (8.7)$$

$$(8.8)$$

The total mean ratio can be therefore express as :

$$\ddot{\Psi} = \Psi_{\bullet Y}^{total} = \sum_{\bullet} \Psi_{\bullet Y}^{overall} = \sum_{\bullet\bullet} {}^L_y \psi_{\bullet Y}^{ratio} = \frac{1}{\bullet Y} \sum_{\bullet\bullet\bullet} \frac{|\bullet Y_\bullet| - 1}{(|\bullet Y| - 1)|\bullet Y_\bullet|} \frac{\sum_{\bullet} {}^L \theta_{\bullet Y_\bullet, \bullet \setminus \bullet Y_\bullet}}{\sum_{\bullet} {}^L \theta_{\bullet Y_\bullet, \bullet Y_\bullet}} . \quad (8.9)$$

$$(8.10)$$

The ratio values $\psi = 1$ represent extremely similar dynamics or dynamics that is unrecognizable by our criterion.

Since our criterion does not acquire absolute values smaller than 1, the ratio $\Psi_{Y_s, Y_\bullet} > 1$ determines significantly higher mutual similarities of the intra DTs than similarity to DTs from other classes. Thus the overall $\Psi_{Y_s, Y_\bullet} > 1$ should be maximized to achieve the highest classes discriminability.

Let us mention that in some significant cases, the ratio of the distance of values may be less than 1. These cases are usually caused either by different dynamics (e.g., leaves versus branches, lee, etc.) within the DT or by the texture being composed of multiple DT textures.

8.2.1 Subclasses criterion validation

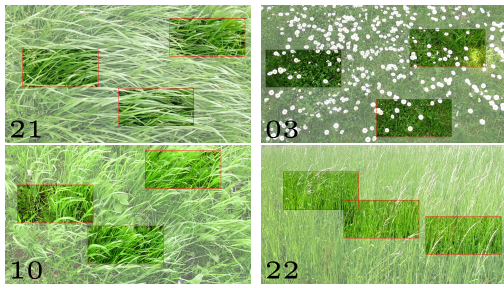


Figure 8.4: Illustration of randomly chosen sub-parts of four DTs. Three sub-parts of each DT are marked and used to the validation experiment.

Suppose, that every texture is random field realisation with given parameters. Then even vaguely defined dynamic texture classes should have similar parameters. Hence the proposed criterion is defined as similarity it is obviously maximized for identity.

Since each texture is a random field realisation with some given parameters, which is homogeneous in both spatial and temporal domain, the texture property from any texture subparts should be remarkably similar but not identical. With respect of this fact, any given dynamic texture can be seen as its concrete dynamic texture class and its sub-parts as textures belonging into (note that the sub-parts should have similar size due to due to possible variations in texture scale).

So for the validation test, the given amount of small randomly placed dynamic texture sub-parts were extracted from every used dynamic texture. Every dynamic texture is then taken as its own dynamic texture type and texture sub-parts as a concrete dynamic texture. As the input DTs (or classes for this purpose) are homogenous texture, the requirement of a consisting property is trivially satisfied.

As a source for used dynamic textures, the class GRASS was chosen as an example and typical class with the most dynamic textures and the most differing dynamics realisations. The several sup-parts from chosen high-resolution textures can be seen in Figure 8.4.

The illustration of the sub-parts similarity values and used sup-parts is on Figure 8.5. The single values inside textures represent the inter-texture similarities and intra-texture similarities ratio and then express the similarity of every particular texture to all other textures. Clearly, the DAISIES texture (on the left) is different from most another DTs.

On the other side, the GRASS textures are more similar to all other. The paired values then represent direct ration between two textures (exactly the all-to-all sub-part similarities ratio) with an order of top-left and left.right. Then the similarity from DAISIES DT to bottom GRASS DT is 1.590 and subsequently the ratio from bottom to right GRASS texture is 1.093.

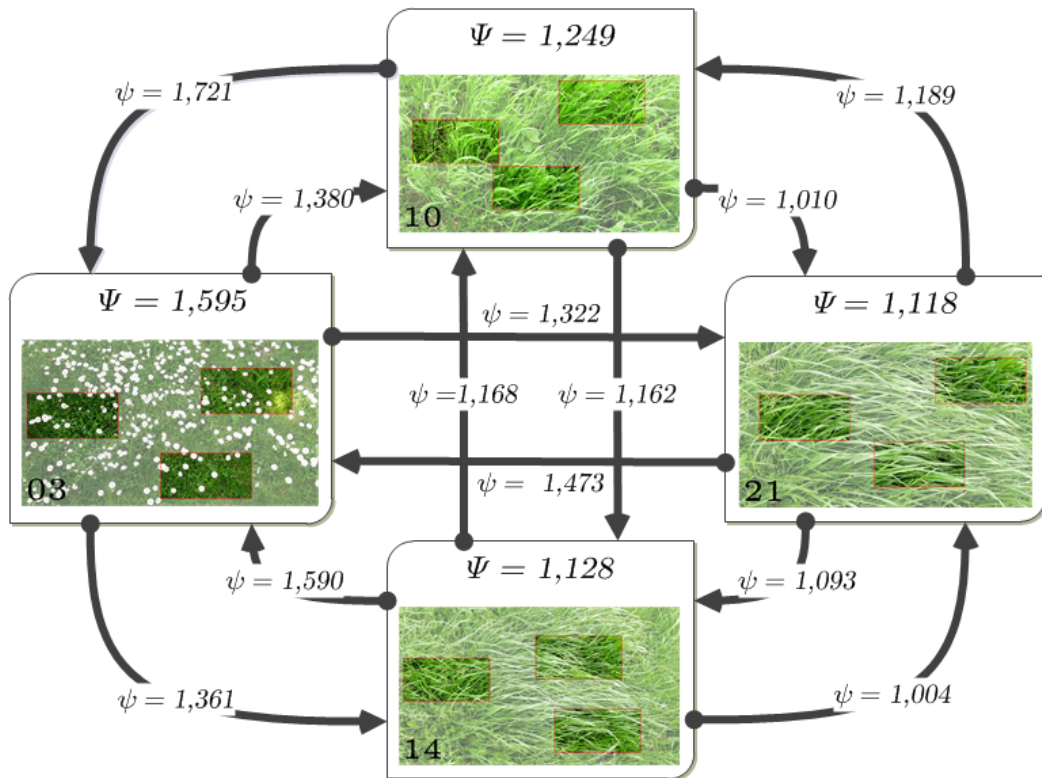


Figure 8.5: Visualization of similarity ratios between four grass DTs: Values of Ψ_{Y_s, Y_c}^j are indicated by arrows. The upper inter-value ratios are registered in top to down and left to right order while the bottom numbers are the opposite directions. Values near circular arrows indicates overall $\Psi_{Y_s, \bullet}$ ratio between inter-DT and all intra-DT similarities. Total ratio $\Psi^{\bullet\bullet} = 1.2725$.

Main Results

9.1 Dynamic Texture Model

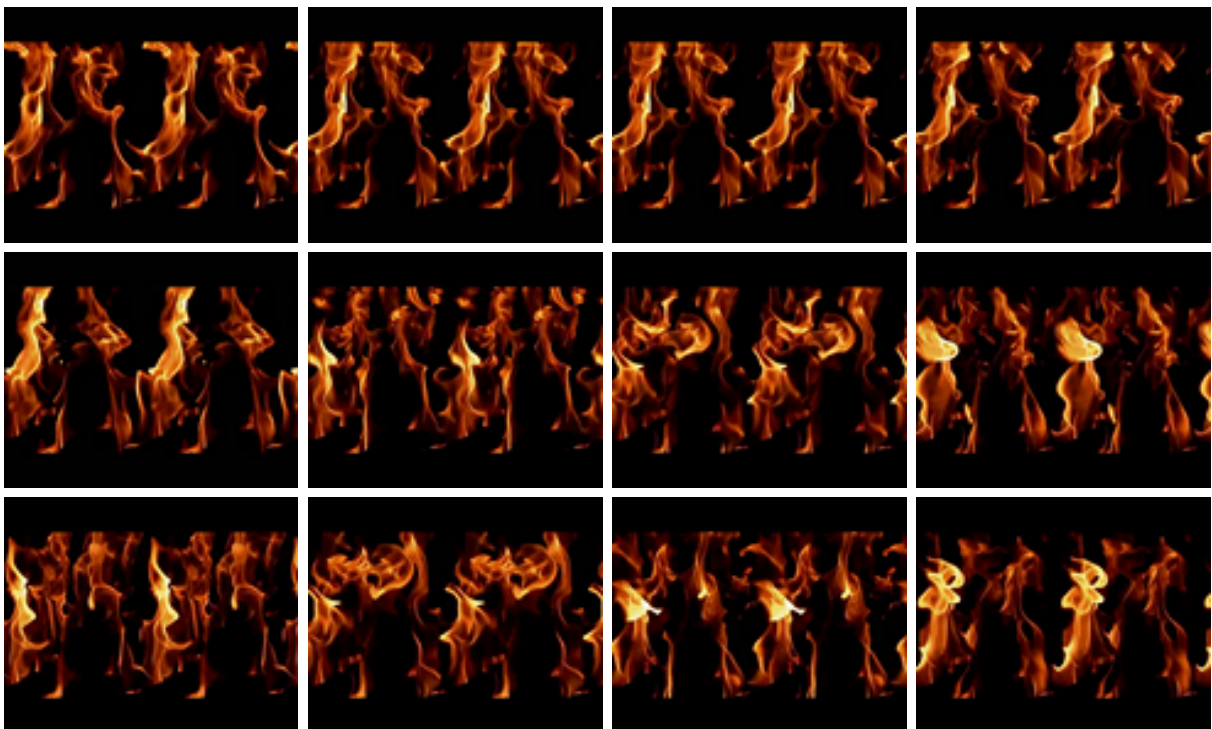


Figure 9.1: *Synthesised FIREA examples*: The set-of-cuts method is used.



Figure 9.2: *Original of DTs FIREA, FIREB and FIREC*

The proposed dynamic texture model shows the ability to model suitable and visually plausible dynamic textures. Here, representative frames from synthesized results are showed. Step between showed frames is at least 10 frames in every sequence. Examples of representative frames of a natural dynamic textures are showed in Table 9.1. The last column of a table demonstrates that method could synthesize result from input consisting of more DTs. Figure demonstrates model ability to adapt to optical flow and enlarge DT in all dimensions. Moreover, two patches are used (here without any editing enhancements). As textures consisting of fire and water phenomena are one of the hardest to synthesise,

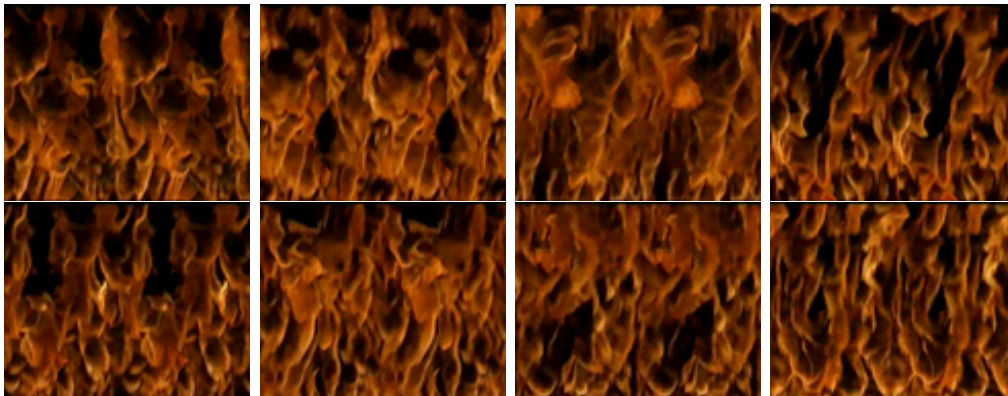


Figure 9.3: *Synthesized texture of FIREB*: The cyclic degrading of cuts can be seen. The set-of-cuts method is used.

some examples of representative frames are showed. Figure 9.1, Figure 9.3 and Figure 9.4 shows the synthesized (time and spatial enlargement) fire sequences. The input sequences

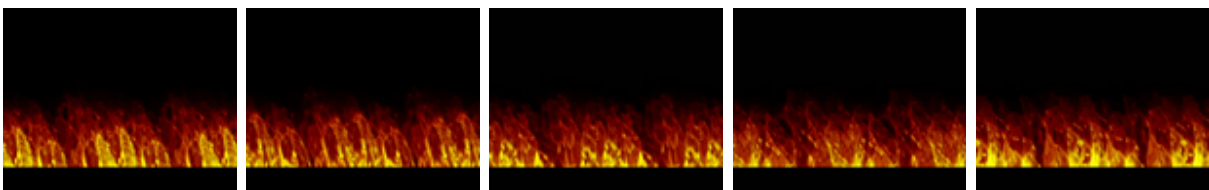


Figure 9.4: *Synthesized texture of FIREC*: The cyclic degrading of cuts can be seen. The set-of-cuts method is used.



Figure 9.5: *Synthesized texture of DT WaterA*

representative frames are in Figure 9.2.

All synthesized FIRE textures were synthesized with method set-of-cuts to handle very erratic flame behaviour. Note, that after many iterations and when flames drastically change their location or even disappear the cuts quality can degrade for a small amount of time.

Another dynamic texture type which is difficult to model is water. Three different dynamic texture enlargement with the various property is shown. The various temporal and horizontal enlargement is demonstrated in Figure 9.5 and various vertical and temporal enlargement in Figure 9.6. The Figure 9.7 demonstrate problematic case as the strong visible content (the reflection of the tree on the surface) is presented. The method set-of-cuts is used in the second case. Let's mention, that all dynamic textures are recommended to see as a video.

Limitations Presented toroidal editing approach can work with many input DTs types, but there are some restrictions. If some perspective distortion is present, the quality

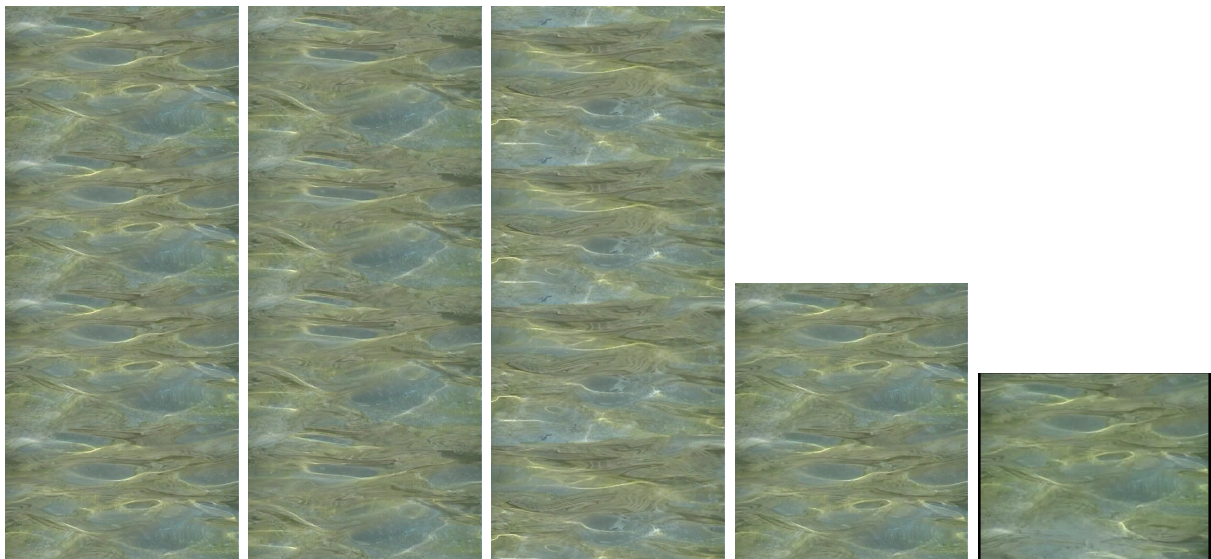


Figure 9.6: *Synthesized texture of WATERB*: The set-of-cuts method is used. Note, that the cut-planes in this case are very dynamics.

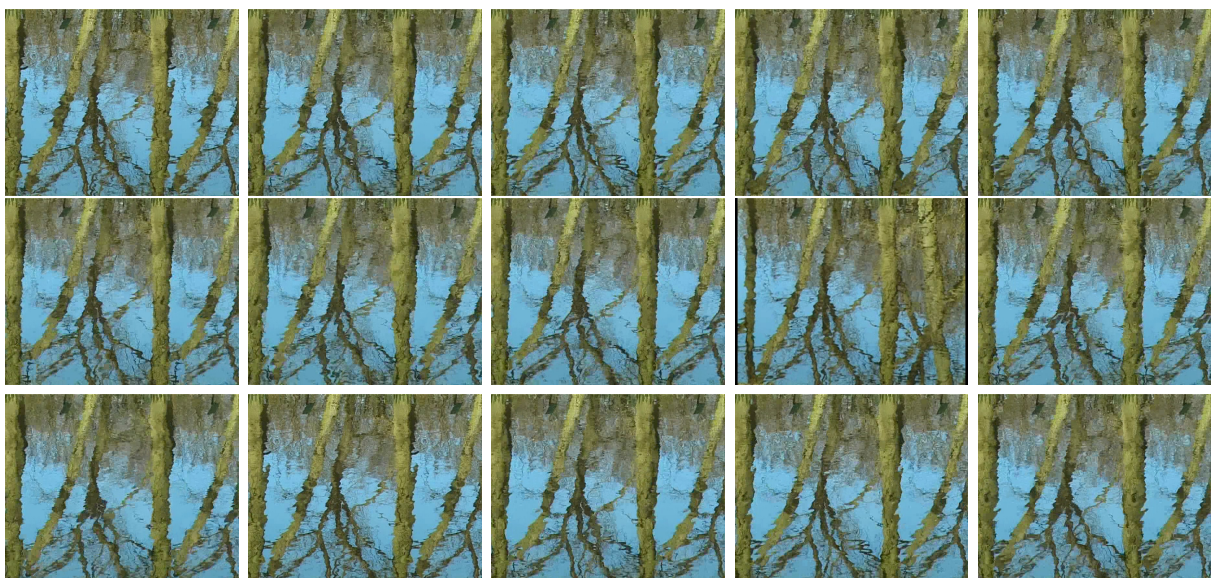
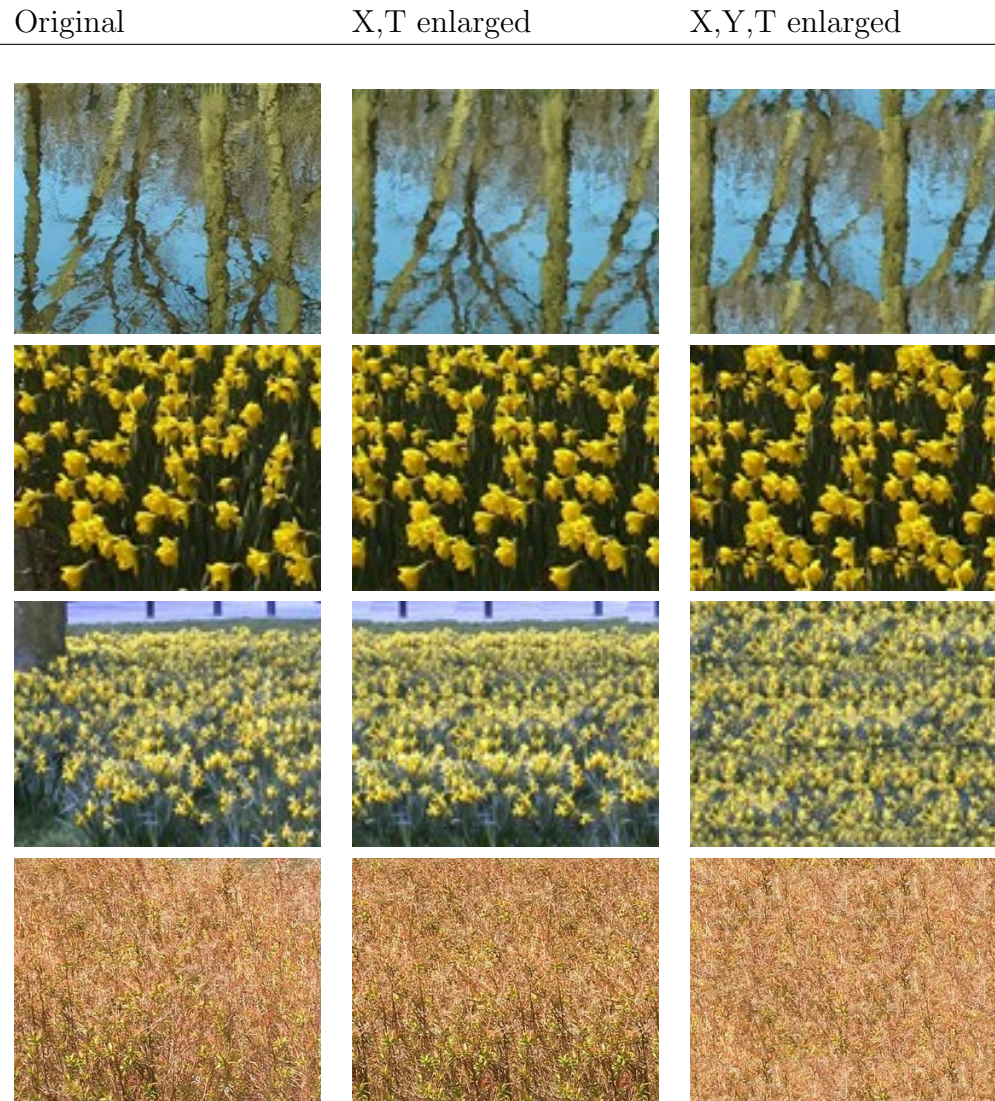


Figure 9.7: *Synthesized texture of WATERC*: One of synthesized frame is original. Rest are synthesized with spatial and temporal dimension to fit the original size.

of the result can vary widely depending on the severity of distortion (but still can be synthesized, see Tab. 9.1, third row), dynamic textures with small non periodical color gradient generate optically visible transition on seams because the are not any repetitive parts to find. The strong unpredicted local dynamics with fast disappearing of small objects can create artifacts too.

Table 9.1: *Demonstration of different enlargements combination*; First column: the original texture; Second column: texture enlarged in temporal and one spatial (horizontal) dimension; Third column: texture enlarged in both spatial and temporal dimensions. Note the WATER texture - due to large structure the vertical dimension is problematic. The SMALL IRISES and WITHERED demonstrate synthesizing from input mix-of-DTs, where result consist from only one of them.

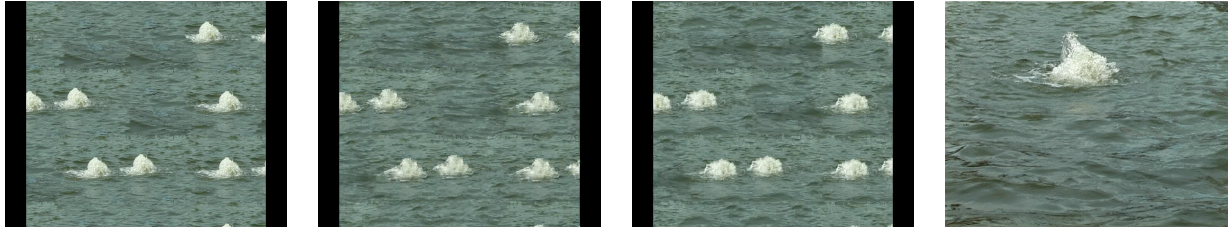


The result with noticeably spatial enlargement can be seen in Figure 9.2. More patched used in every sample. The original texture is on the right, the three textures on the left are downsampled. Figure 9.8 demonstrates noticeable changes in the visual quality if more patches are used.

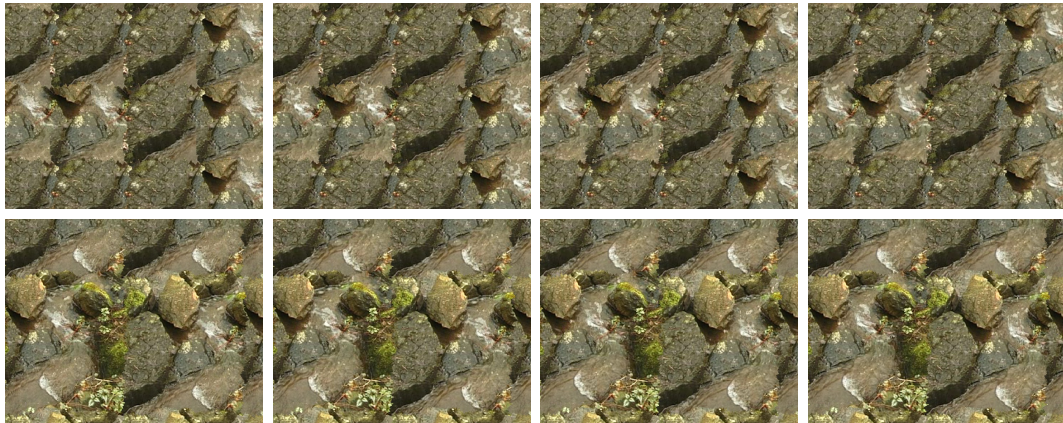
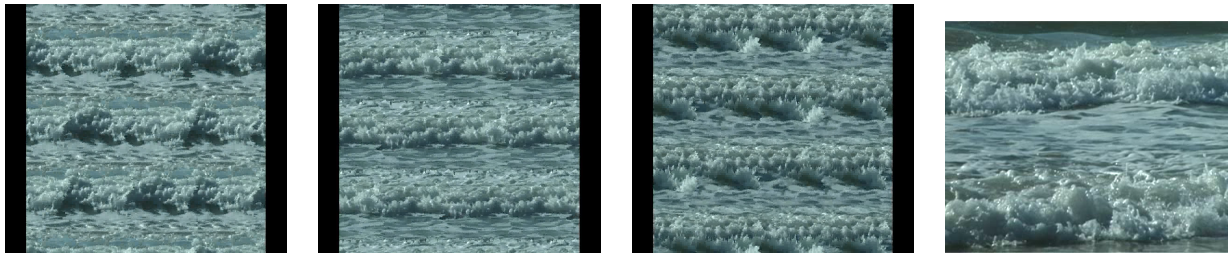
9. MAIN RESULTS

Table 9.2: *Synthesized water texture examples*: The first three columns show examples of synthesized textures, the fourth column of the original sample texture.

WATERD



WATERE



(a) Synthesized textures - first row 2 patches, second row 3 patches;



(b) original

Figure 9.8: *Different number of KAMVOD patches used.*

9.2 Dynamic Texture Editing



Figure 9.9: *Mix-of-DT*: FENCE and SHRUBS are synthesized as mix-of-DT with transition textures. Smaller (scaled) figures are input DTs, larger figures are synthesized output.

A set of individual synthesis algorithm enhancements that serve to increase the limited number of results and, above all, to consider their visual quality was proposed. The enhancement utilized the previously proposed 4D dynamic multispectral textures synthesis and editing based on toroid-shaped tiles.

The synthesized examples illustrate good performance of our approach (many types of DT can be seen on the video). Representative frames of a result subset are shown in figures in this chapter.

The mix-of-DTs are showed in Figure 9.10, Figure 9.9 and Figure 9.11. The Figure 9.10 is pure mix-of-DTs with slight colortone enhancement, the Figure 9.9 demonstrates border patches approach and and Figure 9.11 demonstrate border tile approach.

Moreover, dynamic texture editing approaches are suitable to synthesize dynamic BTF textures. The example of results can be seen in section 9.6.



Figure 9.10: *Mix-of-DT*: GRASS synthesized from two inputs. Smaller (scaled) figures are input DT, larger figure is synthesized output.

The figure 9.13 illustrate zoomed detail of a synthesized texture that consists of only one used patch where the time shift of $\Lambda = 5$ and $\Lambda = 10$ are used. Notice that extremely small value of ($\frac{1}{6}sec$) is sufficient and safe from disturbing the global Φ values.

The temporal dimension editing is showed in Figure 9.13 and scheme of an interactive texture with multiple temporal jump in Figure 9.12. Note small zoomed detail that helps to recognize slightly different dynamics.



Figure 9.11: *Mix-of-DTs*, TREES IN THE RAIN (*multiple tiles*): The resulting dynamic texture is mix-of-DTs with more border tiles. Smaller (scaled) figures are input DT; larger figure is synthesized output.



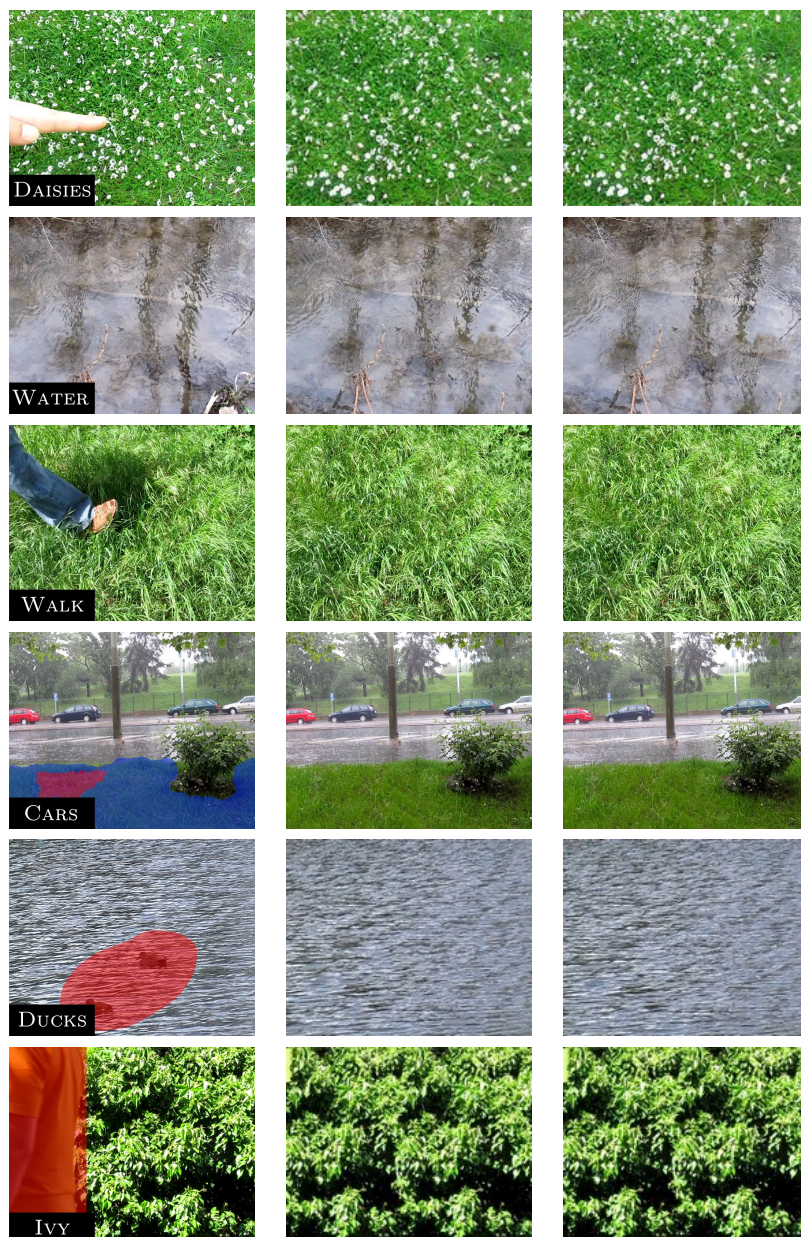
Figure 9.12: *Interactive texture THUJA in the rain.*



Figure 9.13: *Temporary shifted texture* with three identical, time-shifted patches details are zoomed. The texture is consistent in spatial and temporal dimensions but vary in local dynamics.

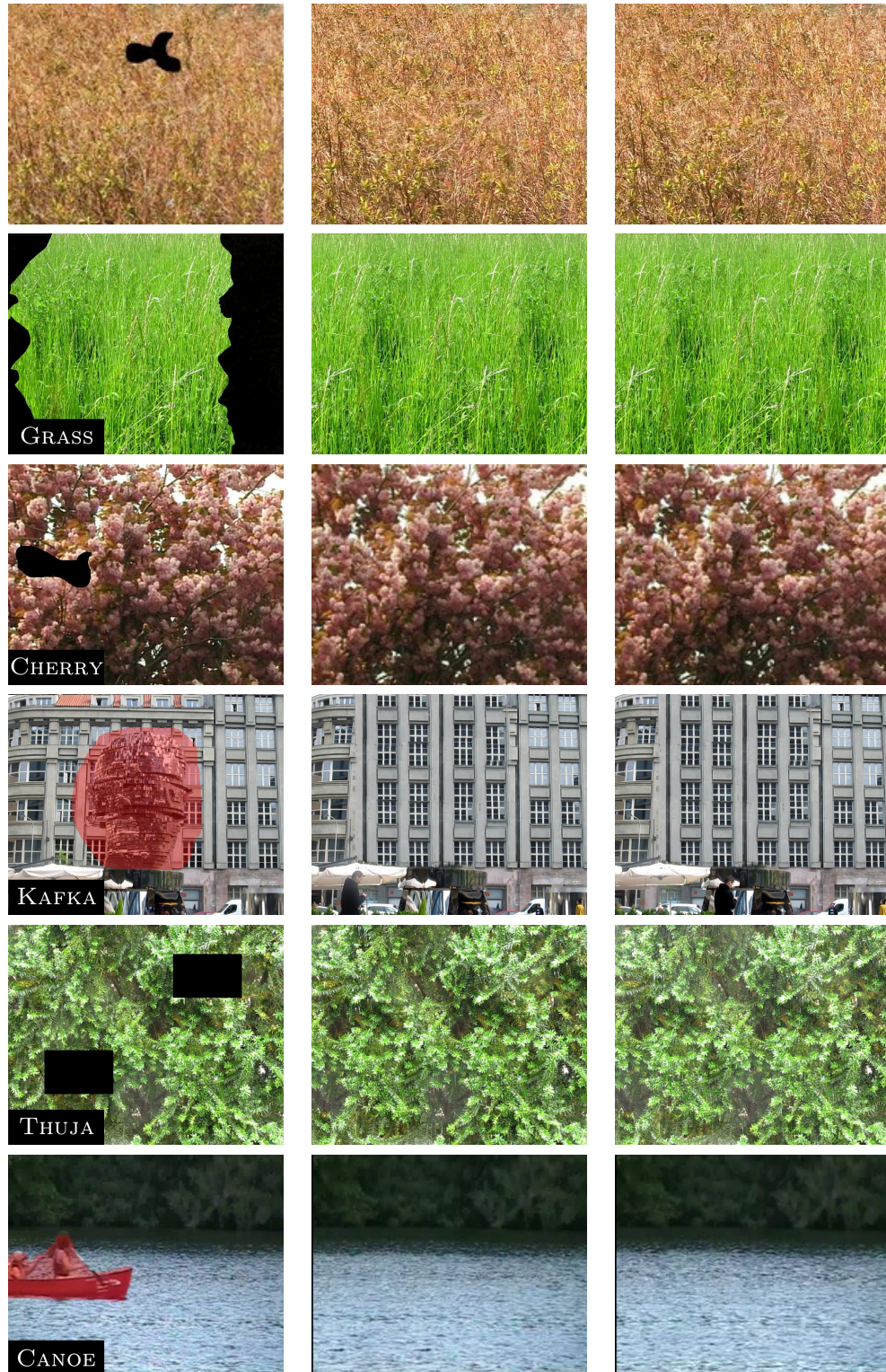
9.3 Dynamic Texture Inpainting

Table 9.3: *Measured textures (left column) and inpainted results, part A*: Measured textures (left column) and inpainted results (first frame - middle column, last frame - right column). DT DAISIES, WATER, and WALK are inpainted automatically. DT CARS, DUCKS, IVY are inpainted with a rough mask (red) for undesirable (CARS) or occluded objects. DT



9. MAIN RESULTS

Table 9.4: *Measured textures (left column) and inpainted results, part A*: Measured textures (left column) and inpainted results (first frame - middle column, last frame - right column). DT CANOE and KAFKA are inpainted with a rough mask (red) for undesirable objects. DT GRASS, WITHERED, THUJA and CHERRY have artificially created masks (black).



The proposed method was extensively tested on high-resolution dynamic textures from the DynTex database and our database of DTs. All used dynamic textures were color with resolution up to horizontal dimension 720 pixels and varying length (from seconds to minutes). Both natural and artificially added occlusions and holes were used. The analysis usually takes several minutes (see Table 9.5) and it strongly depends on the length and the self-similarity of a texture (due to the branch and bound method). The synthesis time is negligible, and it is only limited by memory operation and the video storing/coding operations. Because the inpainting time of our method consists of strictly separated analytical and synthesis parts, our method allows adjustment of the mask or patch usage (picking object to remove - i.e. chose which duck to remove in the DT DUCKS). Presented inpainting approach can work with video texture as well, but dynamic texture must be present. Presented dissimilarity criterion was tested on numerous DTs with various, but usually satisfying results. Non-periodical color gradient, strong handshake or chaotic local dynamics might degrade the resulting inpainted DT quality but can be efficiently solved by a various known method.

9.3.1 Time complexity

The inpainting time of our method consists of strictly separated analytical and synthesis parts. Moreover, hence the patches are found in the analytical part and computation of the minimizing borders is negligible or can be skipped. Our solution is strongly resilient to the hole size or the number of holes - i.e., on DT CHERRY the analysis part takes 37 minutes for one hole, 38 minutes for two holes, and DT WHITERED where the analysis part takes 23 minutes for two holes and 26 minutes for one hole, respectively. Similarly, the time complexity of analysis of WATER DT with one or two holes are almost the same (see Table 9.5). The comparison of our non-optimized C++ implementation with the inpainting algorithm of Newson [97] can be checked on Table 9.5 (measured on a laptop with 2.3Ghz Intel i7 CPU and 8GB memory). Note that although we can inpaint much longer sequences, here due to the time complexity of the Newson method [97] we use much shorter ones. No other calculations than the analysis and synthesis time are required.

Note that the alternative Newson methods are published as Matlab code, so the time comparison should be interpreted with this on the mind. Even so, it is necessary to state that the presented method is (by taking advantage of toroidal approach) an order of magnitude faster. Emphasize, however, the principal drawback of combined synthesis and analysis in alternative approaches. The presented method does not need a significant additional amount of time with increasing output dimension resolution. Comparison to this efficient algorithm with typically inpainting approach can demonstrate the strengths of our approach - strong resilience to the error area size and the number of error regions (error concealment) in contrast to other approaches, i.e. [33].

The other state-of-the-art method like [33] and [142] typically takes at least hours, i.e. [66] process short video with 854×480 pixels, 90 frames and 18% missing took around 3 hours. DTs with a similar property like THUJA or KAFKA took around minutes in our solution. The only time-consuming part of our approach is to find proper time jumps,

Table 9.5: TIME COMPLEXITY COMPARISON.

Scene, hole size [%]	Newson A+S. [<i>min</i>]	Our Anal. [<i>min</i>]	Our Syn. [<i>min</i>]	Size of DT
DAISIES, 7	146	89	<1	720×576×20
DUCKES, 5	113	24	<2	720×576×37
BOATS, 9	203	16	<2	720×576×37
CANOE, 15	384	21	<2	720×576×37
CARS, 3	36	24	<1	720×576×30
WATER, 13	251	45	<1	720 ×576×30
GRASS, 27	320	25	<1	720×560×25
IVY2, 28.5	356	42	<1	720×560×25
WITHERED1, 1.2	103	33	<1	720×560×25
WITHERED2, 2.4	133	39	<1	720×560×25
CHERRY, 1.3	126	16	<1	720×560×25
CHERRY, 2×1.3	151	16	<1	720×560×25
KAFKA, 24	178	17	<1	1920×1080×25
THUJA, 2	58	23	<1	720 ×576×25
THUJA, 2×2	87	22	<1	720 ×576×25
THUJA, 5	70	21	<1	720 ×576×25
THUJA, 3×2	104	22	<1	720 ×576×25
THUJA, 10	86	22	<1	720 ×576×25
THUJA, 2×5	140	22	<1	720 ×576×25
THUJA, 10+5	340	20	<1	720 ×576×25
WALK-L, 63	2442	245	<3	720×576×167
WALK-S, 63	797	72	<1	720 ×576×25

but this can be extremely shorted by user input or setting desired loop length. Usually, our solution is extremely fast due to triple toroid-shaped approach while other approaches repeatedly computed borders and new patches inside the hole which is extremely time-consuming for larger videos and holes.

Long and short DT inpainting Typically, the overlay error reaches a minimum if the source DT contains similar or exactly the same data like \mathcal{H} with the same or different internal structure (δ_r^* contains only temporal shift). This type of DT we called long DT. Usually, inpainting of the long DT creates a visually much better result (for comparison see Figure 9.14). Note that even in the (very) short DT type our solution creates visually seamless results (see Figure 9.15, detailed images of DT WALK) in comparison with alternative methods. Recall that all other input DTs are the short type.

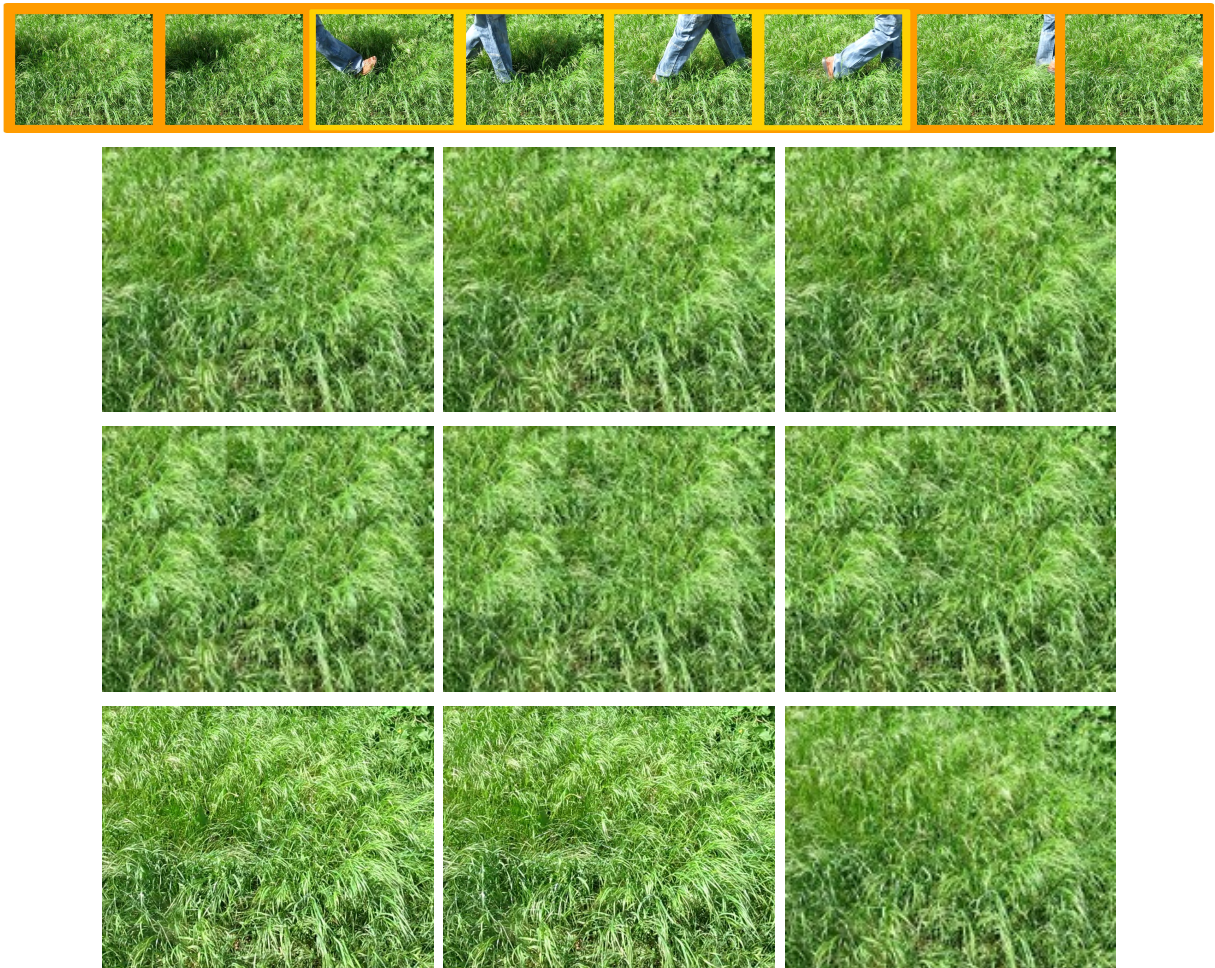


Figure 9.14: *Detail of a WALK synthesis, method comparison* Top: Original measured DT WALK, yellow highlighted short DT, orange highlighted long scene; Second row: Long scene inpaintings, patches found mainly in temporal dimension (note that the temporal cut from the patch to the original DT is presented); Third row: Short scene inpaintings, \mathcal{H} occurs only occluded in the DT; Fourth row: Another measurement of the same scene.

Dissimilarity A simple but sufficient criterion for optical flow dissimilarity was presented. This criterion was shown to be sufficient for most DTs used. The ability to substitute segmentation of inpainted area, or using additional masks have been demonstrated. Due to the placement of area-to-inpaint inside the set of patches and thus no need of tight mask the presented measure is mostly sufficient.

Synthesis-based inpainting We presented DT inpainting not as an iterative greedy-like process but by synthesizing a sufficient large DT and insertion this newly synthesized DT to the inpainted area by the energy minimization criterion. The synthesizing process preserves all essential textural properties like local dynamics and both high and low frequencies. No additional computing or editing of the inpainted data are needed to reach visually satisfying results.

Video inpainting We demonstrated the ability of our method to work properly with video textures containing DTs. A rough mask indicating the source of dynamic textures is useful but not always necessary - information from the dynamic textures surrounding the repaired area is mostly sufficient to determine the correct sample through an iterative search for the optimal filling samples.

Psycho-Physical Validation The presented method results were tested on four groups of almost 150 participants in two test containing blind quality rating and informed pair evaluation. The result (Figure9.19 and Fig 9.18) has shown that the presented inpainting results dominated in most cases and their cases the results were of comparable quality.

Spatio-Temporal Criterion Validation The presented method results were evaluated by spatio-temporal Fourier transformation based criterion and compared to the results of three alternative approach. Proposed method dominates over most of the comparisons with the exception of two cases where the alternative method of Newson showed less lost data values (however, mention here a noticeably higher rate of false frequencies).

Structure preserving Our approach works well on various types of DTs but if the structure of the DT lack high homogeneity diminished visual quality can follow. If the input DT does not contain sufficiently similar (or another sample DT) structures, the presence of a spatial or temporal cut can be visually recognizable (e.g., WALK-SHORT taken from 10 frames). In these particular cases, another DTs can be used as an input, or a similar patch must be adjusted (e.g., by color-tone modeling, manually editing, etc.). In the case that the structural problem is given by the temporal cut, it can be easily solved by reusing a time-consistent toroid sample and preserving the inpainted patches throughout all DT duration.

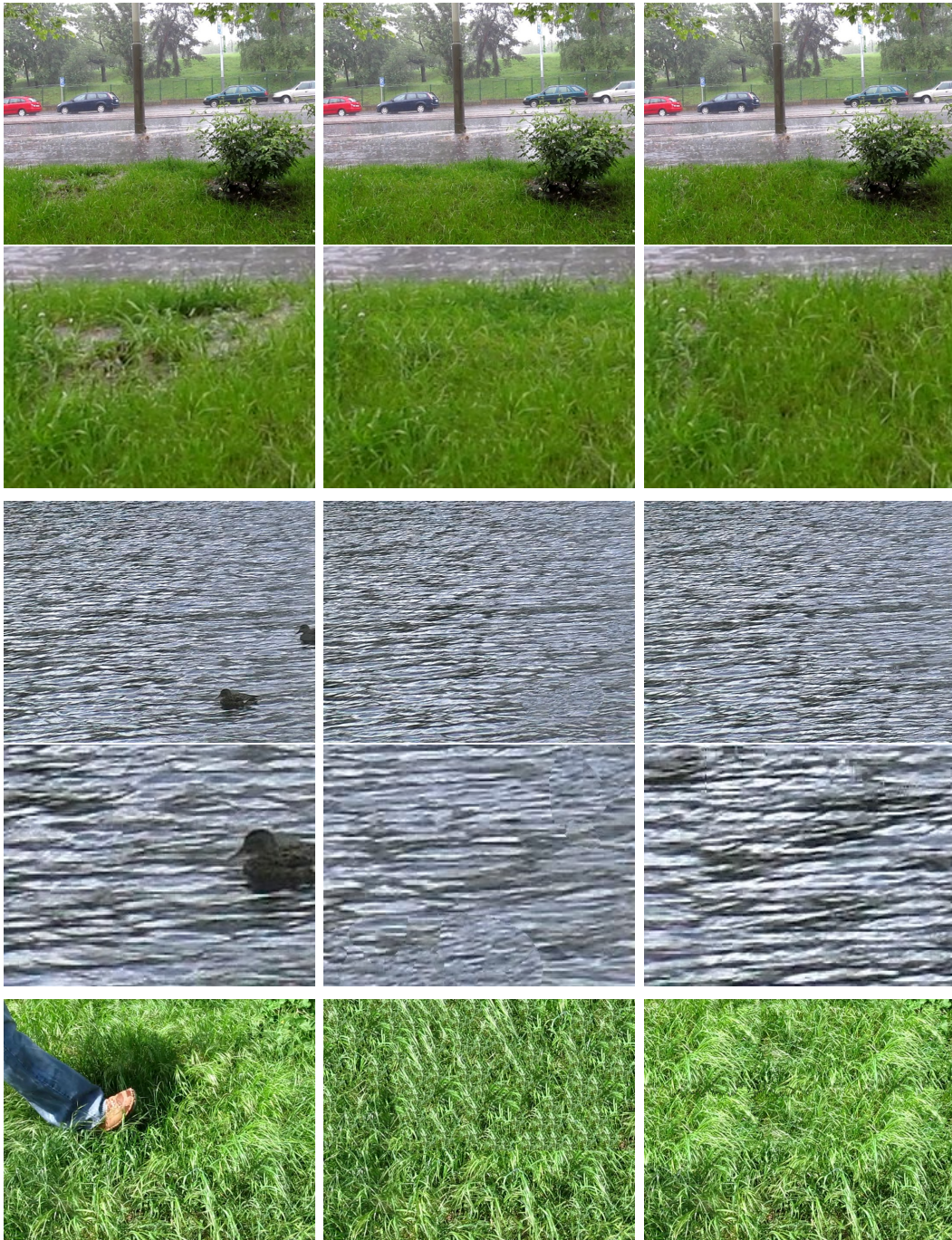


Figure 9.15: *The comparison between our and the Newson method [97] results, part A. Odd rows: left column - original DT, middle column - Newson's inpainted result, right column - our inpainted result; Even rows: details from the images above.*



Figure 9.16: *The comparison between our and the Newson method [97] results, part B. Odd rows: left column - original DT, middle column - Newson's inpainted result, right column - our inpainted result; Even rows: details from the images above.*

Frequency preserving We demonstrate an extremely good efficiency of our solution in the case of preserving both low and high frequencies from the original DTs. Many alternative methods have problems, due to random patch sizes, with preserving some frequencies - typically high on very homogeneous DT (WHITERED or CHERRY and WALK), and low on more video texture-like ones (like KAFKA or WALK). DT WALK is a good example, where due to the inadequate patch size the inpainted alternative methods results can not synthesize larger structures (see Figure9.15, fifth row, middle column). The parameter of the patch size is in our solution driven by the most crucial frequency, and so our solution can preserve them. Different problems can be seen on the KAFKA input where a patch based solution cannot recognize extremely strong and important periodicity (and to obtain satisfying result parameters need to be manually changed). Similar to the spatial frequency problem is a problem with temporal frequencies and optical flow as it can be seen on the inpainted WALK DT. The optical flow (moving of grass in the wind) of the scene is in our approach well preserved but can be insufficiently reproduced by patches with the wrong size.

The illustration of an $L\phi^{Y_s, Y_c}$ behavior and typical changes in the DTs FT is shown in Table 9.7 for original undamaged frames (taken at a different time) and inpainted frames. Note that false and lost frequencies behavior are method given, strongly similar throughout all frames, not only visualized ones.

9. MAIN RESULTS

Table 9.6: *Changes in FTs between original and inpainted DTs frames, part B.* First column DTs, second column FT (power spectrum), third column 3D FT surface detail. The DT BOAT (original, proposed solution, Xie, Newson). Note the false low frequencies in both alternative methods.

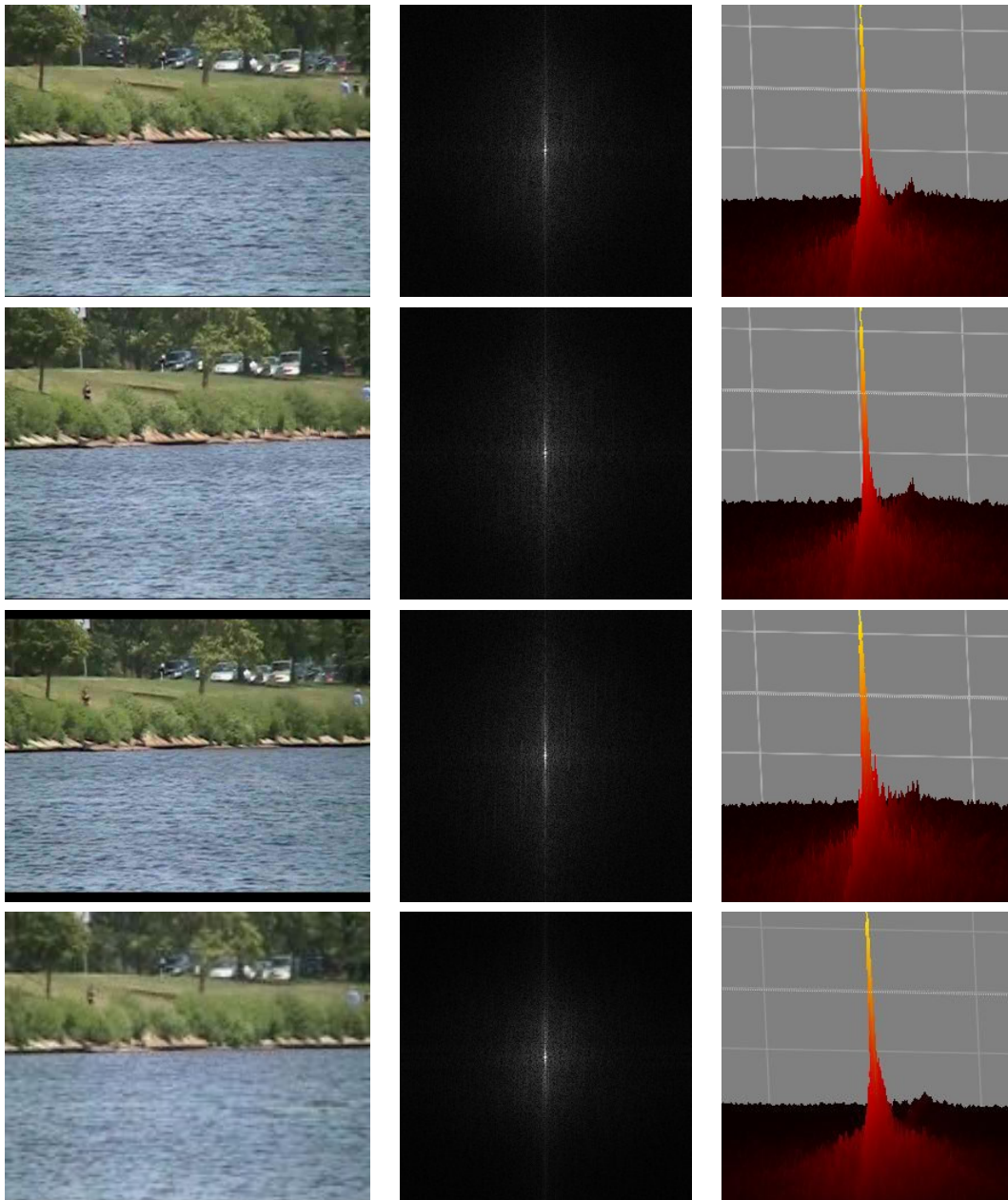
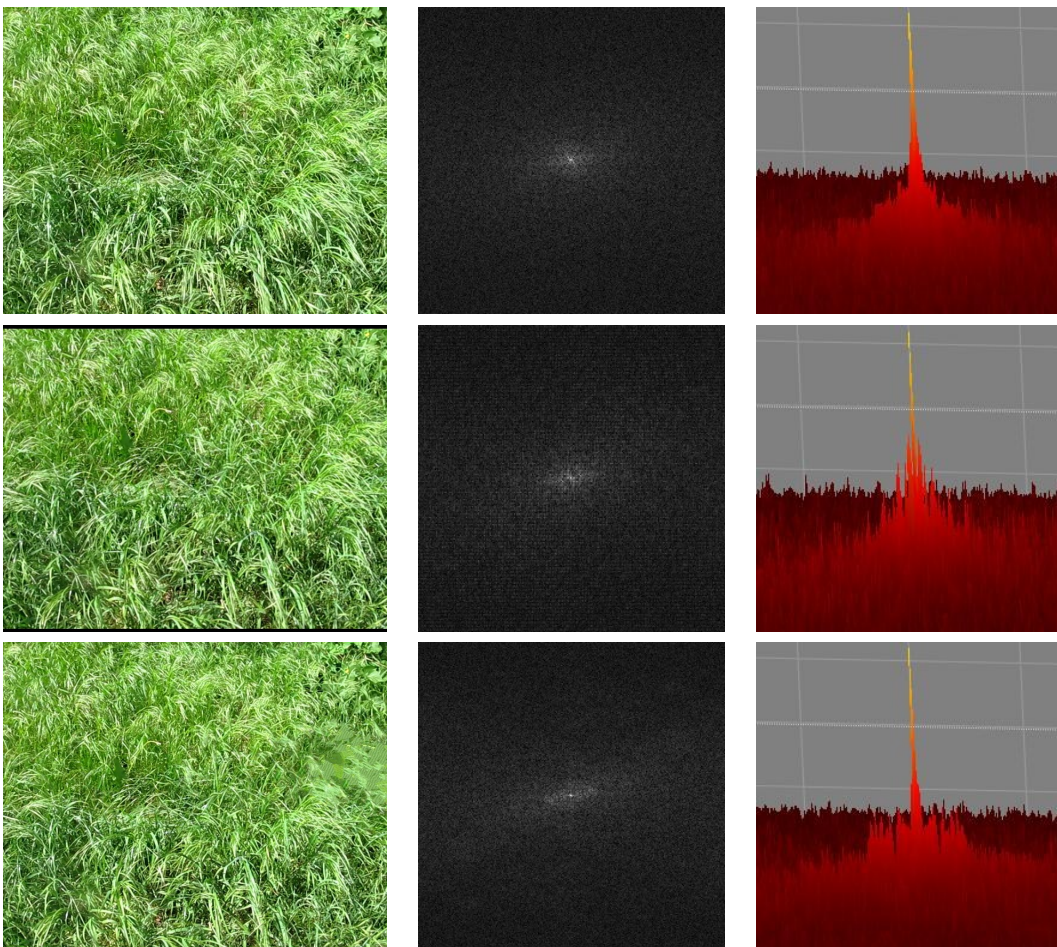


Table 9.7: *Changes in FTs between original and inpainted DTs frames, part B.* First column DTs, second column FT (power spectrum), third column 3D FT surface detail. The DT WALK (original, proposed, Newson). Note the false low frequencies in both alternative methods and a distinct oblique ellipse of false frequencies in DT. WALK.



9.4 Psycho-physical Tests

The psycho-physical tests were performed as a double-blind test on four groups of participants (48, 48, 24, and 30 participants). Note that participants were students unfamiliar with results as well as presenters. As environments for test the schoolroom were used. In first two (more significant) test dynamic textures were presented by the high-resolution projector. For third and fourth test the Scalable Amplified Group Environment with the resolution of 9600x4320 was used. Whereas the results of both types did not show any observable variations, they are presented as one. All participants in every group and environment setting were subjected to the two designed test - blind and informed.

The psycho-physical test environment All tests were performed in Faculty of Information Technology background. For the first and second test of dynamic texture similarity criterion, the schoolroom with high-resolution was chosen. For the third and fourth test the background of SAGElab ¹. All dynamic textures were presented on the Scalable Amplified Group Environment with the resolution of 9600x4320.

The psycho-physical test property All performed psycho-physical tests were performed as a double-blind test thus all participants including presenters were unfamiliar with results. Of course, only authors provided / publicated results and codes were used. Four groups of participants were subjected to tests in the number of 48, 48, 24 and 31 participants. The used scale for the rating was used values from 0 to 5, where 5 denotes higher quality of presented DTs.

Blind psycho-physical test arrangement The first psycho-physical test arrangement consists of presenting one inpainted dynamic textures after another while participant should write down the presumed *quality* of a presented dynamic texture (except low-resolution caused artifacts). Participants should rate the displayed dynamic texture by its visual quality and whether it contains or does not contain apparent interference, retouch or artifacts. As a control mechanism, the original dynamic textures (if exists) was added to test set.

Informed psycho-physical test arrangement Participants should rate the two displayed dynamic textures by its visual quality and whether it contains or does not contain apparent interference, retouch or artifacts. In the informed test the additional information was presented to participants: The original dynamic texture and mask (if available). For used arrangement see Figure 9.17. Thus participants could directly compare the inpainted texture with original and moreover compare more results at one time. In case of more than two methods result exist, all possible combination was used. The presented dynamic texture pairs were randomly placed on the left / right side and both participants and presenters were unfamiliar with the right order.

¹<https://sagelab.cesnet.cz>

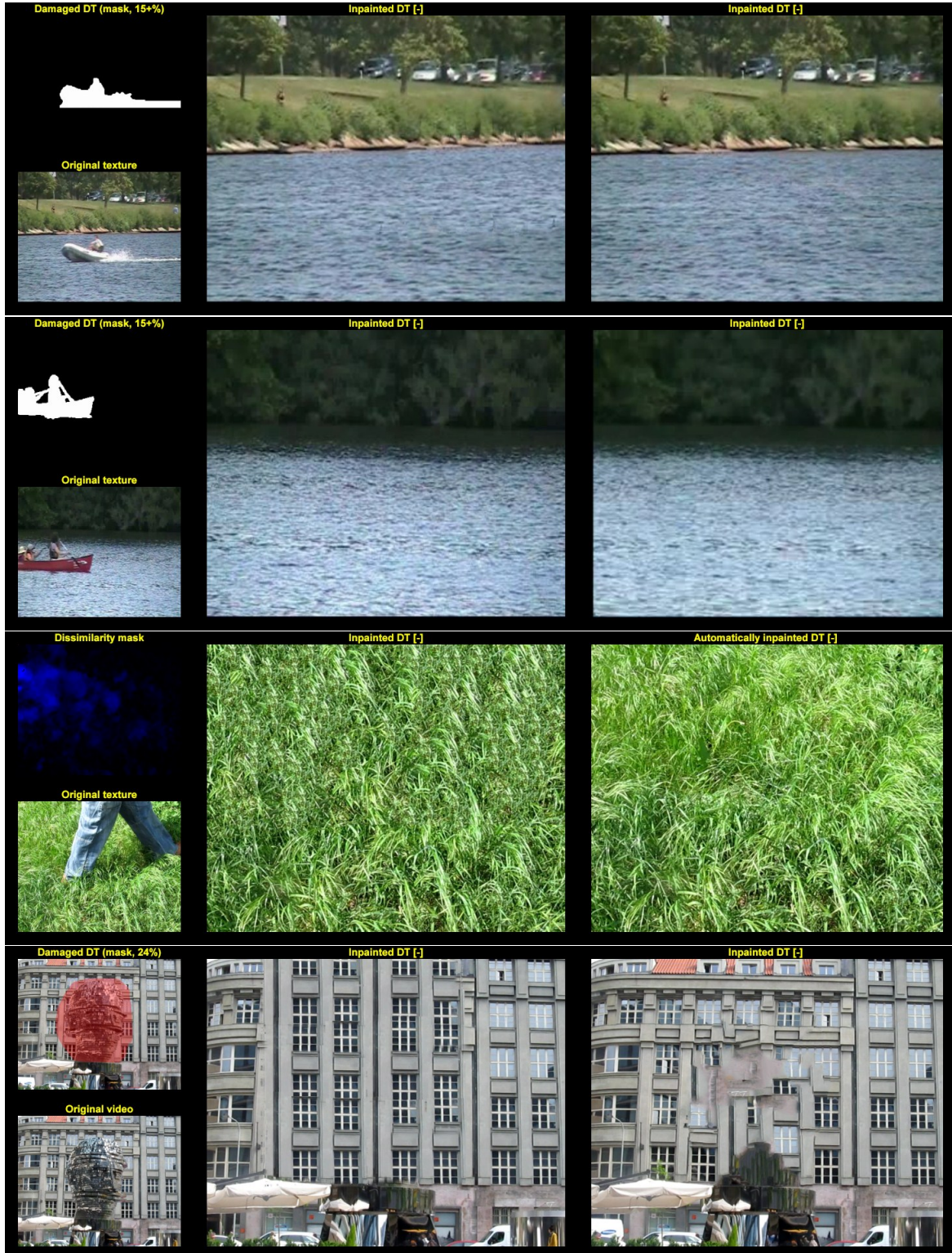


Figure 9.17: *Informed psycho-physical tests arrangement*: Arrangement of DTs BOAT, CANOE, WALK and KAFKA showed. DTs BOAT and CANOE has exact mask. DT WALK has mask (not shown) for alternative solution only, DT KAFKA has rough mask.

9. MAIN RESULTS

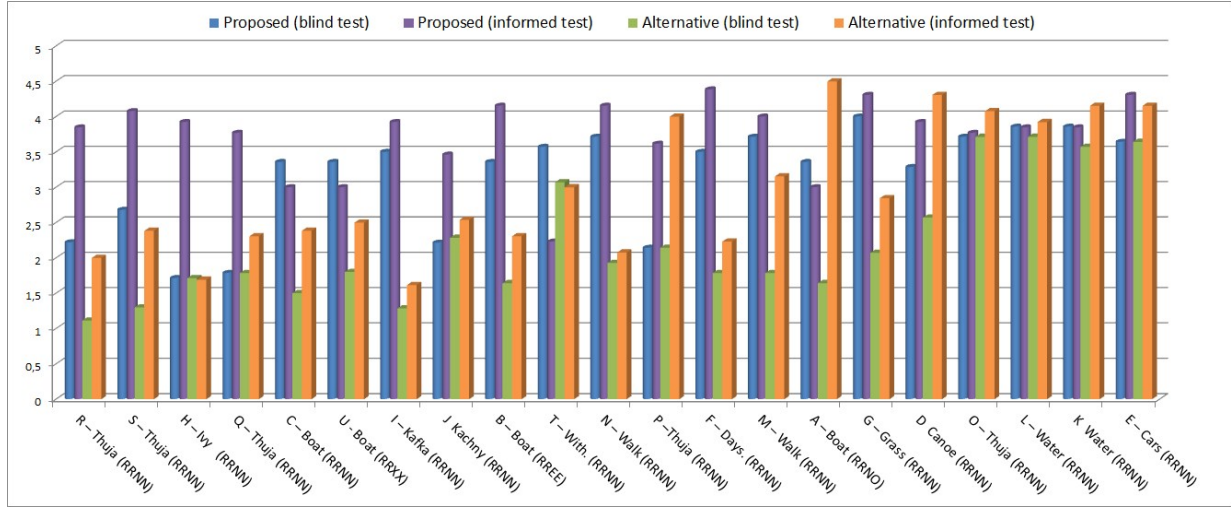


Figure 9.18: *Overall comprehensive results of psycho-physical tests.* For every DT and mask the four values are shown: **Blue** - our method blind test quality rating, **violet** our method informed test rating, **green** - alternative method blind test rating and **orange** for informed rating of an alternative method. The letters in subscribe denotes used alternative method: E - Ebdeli method, N - Newson method, X - Xie method and R - proposed method. The scale (vertical axis) is from 0 to 5, where 5 denotes higher *quality* and thus values should be maximized.

The human ratings (see Appendix B) of over 150 participants was used to validate the proposed criterion. The observers evaluated similarity using a rating between 0 to 5. Tab. C.1 shows the comparison between our criterion values and values obtained from testing. Since the difference may also have negative values, the goal is to minimize the absolute value. A positive value means an underestimation of the criterion due to values obtained from testing; a negative value means overestimation of the criterion.

The complete results of both psycho-physical tests are in Figure 9.18. Note that results are organized from left to right with declining predominance of our method. For comparing inpainting results the more comprehensive graph is shown in Figure 9.19. Emphasize that pairs labeled as $R \times O$ compared our inpainting results with original unedited data (taken from the slightly different time or without the artificial mask). Labels R denote proposed method, N denotes alternative method by Newson[97], X denotes Xie [142] method and label E denotes method by Ebdeli[33]. Label O denotes using of original dynamic texture. The values are grouped by used dynamic texture *and* mask thus DT THUJA has more results.

From the tests carried out, it can be clearly seen that the present method dominates in the vast majority of the comparisons performed. Very interesting is the apparent dominance when comparing more complex textures and the difference between the blind and the informed test. The distinction between the blind and informed test values is expected. The apparent inability to determine (without additional information) the texture edited

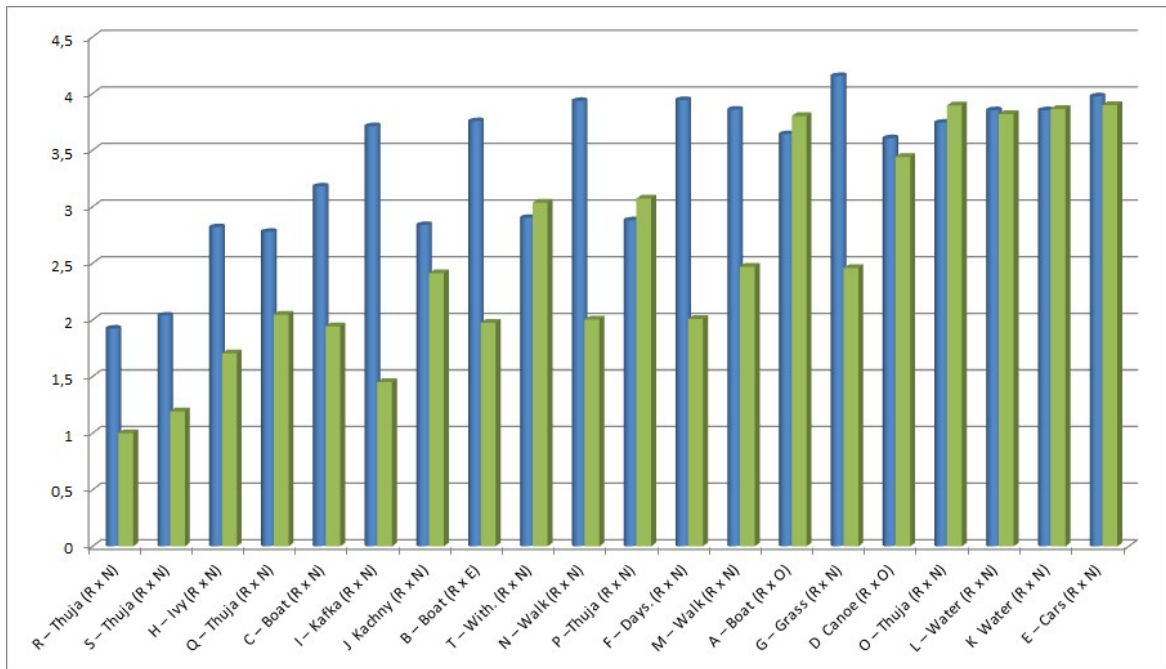


Figure 9.19: *Overall results of psycho-physical tests*: The scale is from 0 to 5, where 5 denotes higher *quality* and thus values should be maximized. **Blue** cones denotes our results rating and **green** bars alternative method ratings. The letters in brackets denotes alternative method: E - Ebdeli method, N - Newson method, X - Xie method, R - proposed method and O denotes original DTs.

area and therefore the original removed texture data if of course inpainting goal. Note also the low variance of the informed test values for our method and therefore the relatively consistent quality of our results.

It seems surprising that the values of the informed test in most cases exceeds blind solution values. This indicates high-quality results of all methods that stand up to the participants' knowledge of the mask. Moreover, the knowledge of the mask and thus ability to focus at exact inpainted locations with a combination of great blind-informed quality jump suggest that used method is very effective. Usually, that means that participants cannot find an artifact in the blind test or anything that suggest what was inpainted. Note the great leap in H-IVY, Q-THUJA or J-KACHNY. The decreasing results C-BOAT and T-WITHERED suggest that with the knowledge of mask, the participant were able to (or they thought so) identify artifacts / exact mask boundary.

Note that presented inpainting solution is considered as lower in both cases only in one THUJA version but with a difference of 0.07 in informed test and thus is considered as the same. Proposed solution is overcome only in P-THUJA, D-CANOE and K-WATER with difference of 0.36, 0.34 and 0.3. Interestingly, the difference between our inpainted solution and undamaged original (case A-BOAT) is only 1,17. From that point of view, the difference in order of $< 0, 0.5 >$ can be considered as extremely small. Moreover, this

stressed the clear domination of proposed method in a major part of tested DT sequences.

Note that some other tests were performed on small groups (tens) of participants and different DTs and arrangements.

9.5 Dynamic Texture Similarity Criterion Results

The behavior of a proposed similarity criterion with varying w_f is showed in Table 9.8. It can be clearly seen that with the decreasing w_f and thus with a smaller number of used frequencies (which must be subsequently stronger in the spatial frequency magnitude) the value ratio is consistent. Thus for test included in this article the value of w_f is set to 3.

Table 9.8: *The average difference of spatiotemporal frequencies criterion* (\downarrow): based on the different w_f value (and thus spatial frequencies with different minimal magnitudes). The difference between Newson (marked as N) and presented method (marked as P) are shown. Results are shown on DTs WALK and IVY. The minimal magnitude was then in order of 140, 160, 180 and 200 for WALK and 100, 160, 180, 200 and 250 for IVY respectively.

WALK	7	5	4	3	IVY	7	5	4	3	2	
α_n	20,30	16,51	15,08	13,42	α_n	20,01	18,37	15,14	13,73	12,41	
α_p	19,54	15,79	14,25	13,16	α_p	17,32	15,55	12,68	11,10	8,88	
false_n	10,56	9,23	8,60	7,81	false_n	13,09	11,63	9,04	8,13	6,46	
false_p	10,13	8,48	7,83	7,89	false_p	8,94	7,64	5,93	4,99	3,27	
lost_n	11,17	8,64	7,89	7,15	lost_n	7,95	7,90	7,68	7,48	8,21	
lost_p	11,19	8,99	8,22	7,55	lost_p	9,07	8,74	7,84	7,36	6,63	
BOAT	7	5	4	3	GRASS	10	7	5	4	3	2
α_n	20,64	19,27	17,89	16,52	α_n	33,15	36,73	41,98	44,81	48,76	58,27
α_p	20,01	18,21	25,45	26,37	α_p	19,04	16,73	14,18	13,32	12,50	10,75
false_n	10,33	10,42	10,50	10,59	false_n	29,41	34,30	40,51	43,58	47,79	57,55
false_p	10,28	9,37	1,11	0,99	false_p	11,07	9,74	8,23	7,63	7,05	5,79
lost_n	12,44	11,38	10,32	9,25	lost_n	5,00	3,63	2,68	2,49	2,35	3,13
lost_p	11,00	11,41	25,21	26,18	lost_o	8,83	8,00	7,23	7,13	7,04	6,51

The overview of all three L methods and comparison of the corresponding $L\phi^{Y_s, Y_c}$ with all used frequencies and the crucial only (w_f is set to 3) is showed in Table 9.9 as a difference between proposed and alternative methods criterion value. The values for DTs BOATS, WALK and DAYSIES is shown. Used resolution is 720×576 for the first DT and 1920×1080 for the second and third one.

Table 9.9: *A spatial Fourier transformation criterion values difference comparison (\uparrow) as a difference between proposed and alternative methods criterion value. $w_f = 3$ for crucial frequencies.*

DT and L	all f.	crucial f.	DT and L	all f.	crucial f.
boats_abs_ebdeli	21421	5895	daysies_abs_newson	-13749	-625
boats_abs_newson	19433	7354	daysies_false_newson	-69884	-891
boats_false_ebdeli	27042	394	daysies_lost_newson	56134	266,2
boats_false_newson	34651	11445	walk_abs_newson	-34364	26,68
boats_lost_ebdeli	-552	5501	walk_false_newson	-8918	825,4
boats_lost_newson	2514	3921	walk_lost_newson	-110354	-1488

The overall comparison of the spatio-temporal Fourier transformation criterion values is shown in Table 9.10 as the difference between alternative method values and our method values comparison for DTs BOATS, CANOE, GRASS, IVY and WALK. The values of $Our^L \phi^{Y_s, Y_c} - Alt^L \phi^{Y_s, Y_c}$ are thus maximal and positive if presented method result is rated as more similar and negative if the alternative method result is rated as better (more similar). Note that only in two cases - GRASS_{lost} and IVY_{lost} the alternative methods are rated as better and in case of BOAT-EBDELI_{lost}, CANOE-NEWSON_{lost} and WALK-NEWSON_{lost} the results are very similar.

Table 9.10: *Overall comparison of frequency criterion values as the difference between proposed method and alternative methods (\uparrow): DTs BOATS, CANOE, GRASS, IVY, and WALK used. Note the proposed method takes negative (and thus is worse) in only two cases - GRASS_{lost} and IVY_{lost}. Used $w_f = 3$.*

texture	criterion value	texture	criterion value
boats_abs_ebdeli	15,9509	canoe_abs_newson	1,0533
boats_abs_newson	1,3543	canoe_false_newson	1,0474
boats_abs_xie	0,8504	canoe_lost_newson	-0,0324
boats_false_ebdeli	8,7607	ivy_abs_newson	2,8218
boats_false_newson	0,6590	ivy_false_newson	3,9903
boats_false_xie	2,7462	ivy_lost_newson	-0,8381
boats_lost_ebdeli	0,1556	walk_abs_newson	0,7557
boats_lost_newson	0,9712	walk_false_newson	0,4336
boats_lost_xie	0,8948	walk_lost_newson	-0,0184

The time complexity comparison of our method with a very effective approach of Newson[97] can be seen in Table 9.5, the quality comparison in Figure 9.15, and in the accompanying video. We chose the Newson's method over newer one as [66] as a new and very good inpainting algorithm with open Matlab code, existing comparison with another

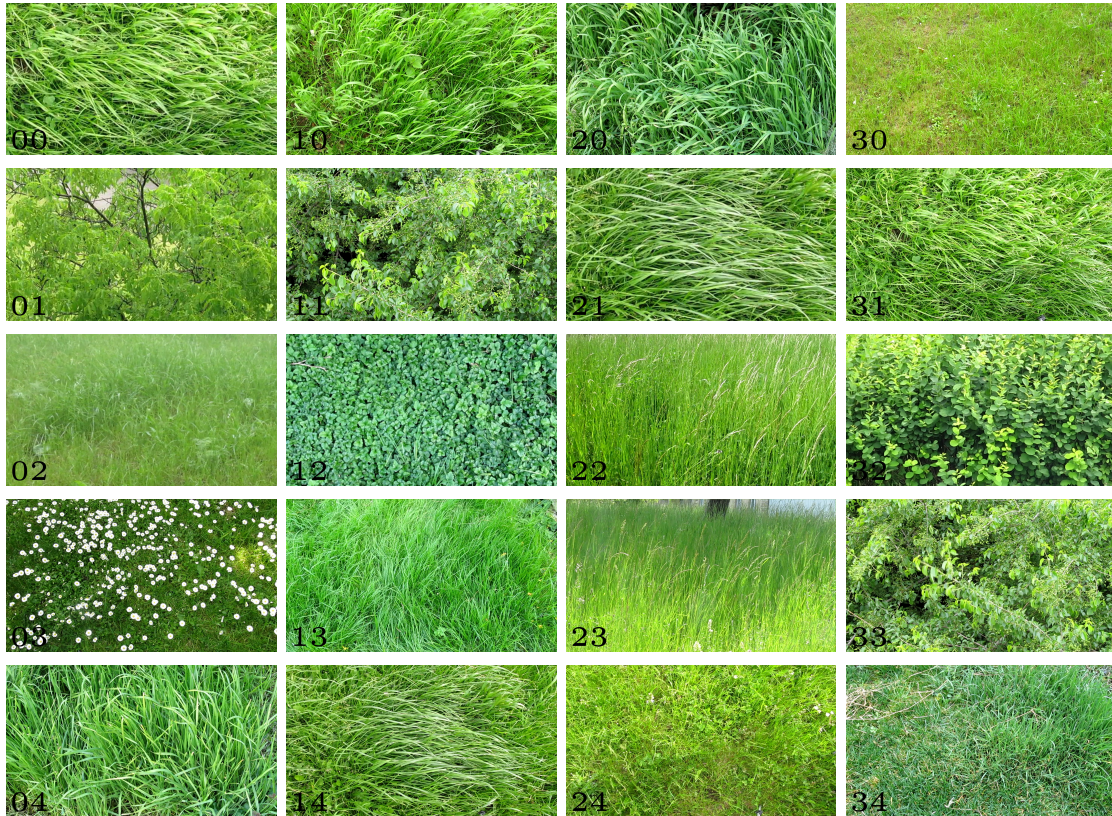


Figure 9.20: *Representative DTs subset for psycho-physical tests*: Representative selected frames from several GRASS and FOLIAGE dynamic textures marked with labels.

inpainting method[66] with known strengths and limitations, and as typical patch based inpainting approach.

Experimental Dynamic Textures Subset

The proposed similarity criterion was tested on our dynamic texture database and the DynTex[104] database. To determine the texture class, the DynTex class labels (SEA, FLOWERS, FOLIAGE, etc.) were used also for our own database.

Our DTs have a noticeably higher resolution (including Full HD and higher) and time duration in the order of tens of seconds or minutes. Figure 9.20 illustrates representative GRASS and FOLIAGE Full HD DTs with all to all computed similarity criterion (only a representative part is shown for explanation purposes).

The dataset consists of visually similar textures, yet with different dynamics. The database contains textures with pronounced (strong wind, rain) and low dynamics as well as textures with similar dynamics (directional wind) but very different structure.

Due to the high variability of dynamics and structure, while maintaining a similar class, this is an interesting and representative dataset allowing useful comparison and demonstration of strong and weak characteristics of the criterion. The criterion was

also tested on other datasets with similar results. The furthermore criterion values can be found in the next chapter.

The submitted dataset has been subjected to detailed psycho-physical testing with more than 150 users who evaluated the likeness of individual DTs. This user evaluation (Tab. B.1) is therefore used for the proposed criterion validation.

These results show both pairs rated as extremely similar (21-14), and couples that have been rated as highly different (11-12, 10-12). For example, DTs 23-22 and 02-12 have similar structures and similar dynamics but very different color, but these couples were evaluated as relatively similar.

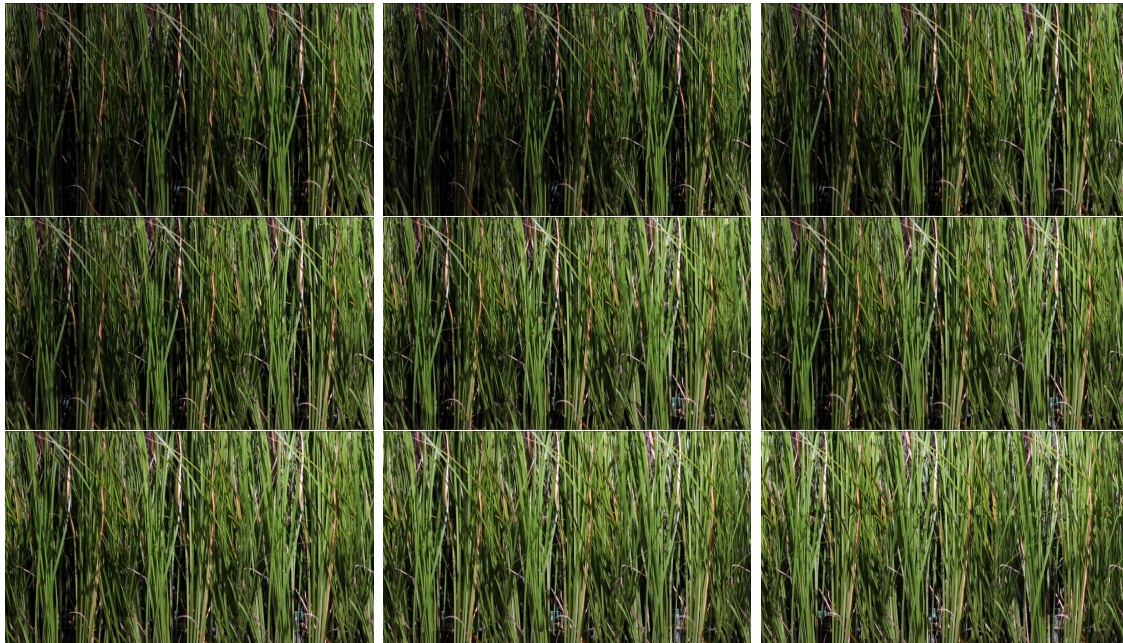


Figure 9.21: *Several frames from composed DBTF*: Several frames from the synthesized video in top-down and left-right order. The source of illumination is moving from right (outside frames) to middle.

9.6 DBTF

The resulting synthesis of an approximative DBTF shows that the DBTF as a medium is possible and valid. The incredibly complex natural object consisting of a high number of self-occluding and self-shadowing could be very problematic to model in a classical way.

Of course, proposed DBTF and DVBTf model are topics, which will be in focus in further works.

The presented result shows GRASS texture in Figure 9.21 composed of many samples under a varying illuminance. Note that only one patch is used and thus temporal shift editing method is applied to suppress the dynamics repetition. Larger arrangement is showed in Figure 9.22 with stronger time shift ($\delta = 7$).

More complex results which consist of the samples acquired under different viewing angles can be seen in Figure 9.23. The measured object (plant), has a strongly varying property under different view directions (top three row, all consist of a one spatial and one temporal dimension enlarged texture). The result is then composed together to one resulting texture which covers the desired object texture from $\theta_v = 90$ to 10 .

The presented results (Figure 9.21) show GRASS texture composed of many samples under a different illuminance. Note that only one patch is used and thus again editing method is applied to suppress the dynamics repetition.



Figure 9.22: *Several frames from composed DBTF, detail:* Several frames from the synthesized video in top-down order. The source of illumination is moving from right (outside frames) to middle. Note the temporary edited samples and its varying dynamics (see a curved blade of grass).



Figure 9.23: *composed DBT example*: Resulting DBTF (bottom) composed of three textures with different viewing angle.

Conclusions

10.1 Summary

The new approach for fast 4D multispectral dynamic textures modeling was presented. The proposed dynamic texture model is based on a triple toroid-shaped tile, has separated analysis and synthesis and is able to create an infinitely long dynamic texture in both spatial and temporal domain. Therefore the synthesis can be done very quickly and effectively. The synthesized high-resolution examples illustrate good performance of a proposed approach for many types of dynamic textures. Our approach can edit temporal and spatial properties of DT and can easily create a mix of dynamic textures.

A set of specific synthesis algorithm enhancements that increase the limited number of results and, above all, to enhance their visual quality was proposed. The editing ability and creating of mix-of-DTs was subsequently applied to the synthesis of approximative dynamic BTF textures.

The proposed dynamic texture model ability to inpainting problem was demonstrated. The model utilizing allows very efficient filling of an error area with maintaining the visual plausibility. Moreover, due to toroidal principle, the process takes several minutes instead of hours, days, or more. The inpainted examples illustrate good performance of our approach for many types of DTs. We have demonstrated our solution on several DT inpainting problems and validated its efficiency through performed psycho-physical tests.

A novel entirely spectral dynamic textural similarity criterion was presented and successfully validated with several state-of-the-art criteria, which were straightforwardly extended to temporal domain. Due to the absence of reliable DT benchmarking extensive psycho-physical measurements of dynamic texture similarities with more than 150 participants was performed. The presented criterion significantly corresponds with similarity values obtained from psycho-physical tests. The validation of presented criterion has clearly shown its discriminatory and sufficiently diversifying capabilities within a given dynamic texture class.

10.2 Contributions of the Dissertation Thesis

A) DYNAMIC TEXTURE SYNTHESIS MODEL

A sufficiently general and descriptive model that offers a more in-depth insight into dynamic in the dynamic texture, its regularity and many others properties of a dynamic texture was proposed. The creation (synthesis) of the new dynamic texture which is visually similar to the original one is the fundamental contribution of this dissertation thesis (see Chapter 3).

Spatial Domain For only spatial domain enlargement, the double toroidal-shaped patch approach was presented. The sufficiently representative toroidal-shaped patches allowed extremely fast yet effective synthesis with the possibility to handle various types of DT.

Temporal Hence the perceptual property between visual and temporal dimension differs, the temporal domain must be handled differently. The presented model temporal synthesis utilize spatial dimensions synthesis and enhance it with an ability to adapt to the dynamics of an every particular DT.

B) DYNAMIC TEXTURE EDITING APPROACH

Pure synthesis approach is useful but not satisfactory enough in many specific cases. The editing of patch arrangements, suppression of visual disturbances, editing of the dynamics or specific color balancing that increases the amount of synthesis results and field of uses was demonstrated (see Chapter 4).

Interactive texture Due to the presented general temporal synthesis approach, the transition between more DT types can be created. Synthesized results have varying visual and dynamics circumstances like illumination or weather conditions. The transition order can be due to separated analysis and synthesis easily controlled.

Mix-of-DTs The general video data usually contains more dynamic textures. Presented approach can analyze mix-of-textures and subsequently synthesize great amount of specific arrangements containing many different dynamic texture types.

Temporal editing One of the intelligent sampling method disadvantages is the strictly limited amount of source data patch which can lead to visually repetitive effect. The slight temporal changes of the dynamics and structural factors which can distinctly reduce this undesirable behaviour was shown.

C) INPAINTING AND ERROR CONCEALMENT

One well-known image processing goal is to remove error areas, artifact or unwanted parts of the dynamic texture. The proposed dynamic texture inpainting approach allows to focus on cases with the dynamic

background, which are usually extremely difficult, and handle it with the advantage of fast toroidal-shaped patch approach. The proposed method focuses on the perceptual quality in term of human perception and produces visually high-quality results (see Chapter 5).

Object removal Assuming that unwanted object distorts surrounding area, which is supposed to have dynamic texture property, the need of some exact mask can be, in some cases, bypassed. Distorting parts in given dynamic texture can be roughly marked automatically by proposed dissimilarity measure based on optical flow and color information.

- D) **DYNAMIC TEXTURE PERCEPTUAL SIMILARITY** Dynamic texture similarity ranking is a challenging and still unsolved problem. Evaluation of how well are various dynamic textures similar in the way of human perception is challenging even for static textures and requires tedious psycho-physical experiments. Characteristics of human perception and a different way of perceiving spatial and temporal dimension complicate the definition of similarity criterion in an analogous way as for the dynamic texture synthesis (see Chapter 6).

Multidimensional Frequency Similarity Criterion A novel dynamic texture criterion based on the Fourier transformation and properties of spatiotemporal frequencies was presented. The proposed criterion allows for a qualitative comparison of both synthesis and inpainting results. Presented criterion results correlate well with a performed psycho-physical test (see Chapter 8).

Synthesis validation The multidimensional frequency similarity criterion has made it possible to detect and measure crucial frequencies in the temporal and spectral domain and thus be used to synthesis results quality verification.

Inpainting validation A typical inpainting problem is the neglect of fundamental spatial frequencies in the newly created data patch or the creation of frequencies that are not to be found in a given source dynamic texture. The proposed criterion has made it possible to measure such frequencies and thus compared two inpainting results in quality in this way.

- E) **DYNAMIC BTF** The proposed dynamic texture model editing abilities were utilised to synthesize novel dynamic BTF textures. The demonstrated promising result suggests that the dynamic bidirectional textural function model is a suitable model for modeling dynamic processes under varying illumination and observing conditions (see Chapter 7).

10.3 Future Work

The author of the dissertation thesis suggests to explore the following:

- The presented criterion evaluates the spatial frequencies temporal behaviour but not take into account possible color variations, nor other suitable features (for example PCA features or AR model features). Further research will also cover this aspect together with possible scale and illumination invariance.
- In the following work, a developed solution should handle perspective translation or handshake to overcome typical problems with moving scenes.
- A more efficient way of solving and find the (sub)optimal solution for the tiling problem with using a mix-of-DT is also possible.
- It is possible and desired to expand proposed criterion to more texture fidelity and human visual system problems.
- The promising DBTF result, although only the start, suggest that furthermore measuring of a suitable data and better handling the approximative results, i.e. with novel state-of-the-art, BTF methods could lead to a useful dynamic objects data model usable, i.e. in the VR.

Bibliography

- [1] Aseem Agarwala, Ke Colin Zheng, Chris Pal, Maneesh Agrawala, Michael Cohen, Brian Curless, David Salesin, and Richard Szeliski. Panoramic video textures. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 821–827, New York, NY, USA, 2005. ACM.
- [2] Mark Allmen and Charles Dyer. Toward human action recognition using dynamic perceptual organisation. *Looking at People Workshop*, 1993.
- [3] C. H. Anderson, P. J. Burt, and G. S. van der Wal. Change detection and tracking using pyramid transform techniques. 579:72–79, December 1985.
- [4] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.
- [5] Ronen Barzel and Alan H. Barr. *Physically-Based Modeling for Computer Graphics, First Edition : A Structured Approach*. Morgan Kaufmann, 2013.
- [6] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–466, September 1995.
- [7] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. Super-helices for predicting the dynamics of natural hair. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1180–1187. ACM, 2006.
- [8] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *Trans. Img. Proc.*, 12(8):882–889, August 2003.
- [9] Marcelo Bertalmío, Simon Masnou Caselles, and Guillermo Sapiro. Encyclopedia of computer vision, ch. inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2011. Springer.

- [10] Marcelo Bertalmío, Vicent Caselles, Simon Masnou, and Guillermo Sapiro. Inpainting. In *Computer Vision, A Reference Guide*, pages 401–416. 2014.
- [11] Marcelo Bertalmío, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00*, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [12] J. Bigun and J.M.H. du Buf. N-folded symmetries by complex moments in gabor space and their application to unsupervised texture segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(1):80–87, January 1994.
- [13] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005.
- [14] Christina A Burbeck and DH Kelly. Spatiotemporal characteristics of visual mechanisms: Excitatory-inhibitory model. *JOSA*, 70(9):1121–1126, 1980.
- [15] A. B. Chan and N. Vasconcelos. Mixtures of dynamic textures. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 641–647 Vol. 1, October 2005.
- [16] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):909–926, May 2008.
- [17] Antoni B. Chan and Nuno Vasconcelos. Layered dynamic textures. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2006.
- [18] Damon M Chandler and Sheila S Hemami. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE transactions on image processing*, 16(9):2284–2298, 2007.
- [19] R. Costantini, L. Sbaiz, and S. Susstrunk. Higher order svd analysis for dynamic texture synthesis. *IEEE Transactions on Image Processing*, 17(1):42–52, January 2008.
- [20] Roberto Costantini, Luciano Sbaiz, and Sabine Süsstrunk. Dynamic texture analysis and synthesis using tensor decomposition. In *International Symposium on Visual Computing*, pages 245–254. Springer, 2006.
- [21] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Object removal by exemplar-based inpainting. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2003.

-
- [22] George R. Cross and Anil K. Jain. Markov random field texture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-5(1):25–39, January 1983.
- [23] K.J. Dana, B.V. Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and Texture of Real World Surfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 151–157, Jun 1997.
- [24] K.J. Dana, B. Van-Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and Texture of Real World Surfaces. *ACM Transactions on Graphics (TOG)*, 18(1):1–34, Jan 1999.
- [25] Jeremy S De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368. ACM Press/Addison-Wesley Publishing Co., 1997.
- [26] T. Ding, M. Sznaiier, and O. I. Camps. A rank minimization approach to video inpainting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, October 2007.
- [27] G. Doretto. Dynamic texture modeling. Master’s thesis, University of California, Los Angeles, CA, June 2002.
- [28] G. Doretto. Dynamic texture modeling. Master’s thesis, University of California, Los Angeles, CA, June 2002.
- [29] G. Doretto. *DYNAMIC TEXTURES: modeling, learning, synthesis, animation, segmentation, and recognition*. PhD thesis, University of California, Los Angeles, CA, March 2005. Committee: Adnan Darwiche, Petros Faloutsos, Demetri Terzopoulos, Ying Nian Wu, Stefano Soatto (Chair).
- [30] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1236–1242 vol.2, October 2003.
- [31] G. Doretto, E. Jones, and S. Soatto. Spatially homogeneous dynamic textures. In *eccv*, volume 2, pages 591–602, Prague, Czech Republic, May 2004.
- [32] Frédo Durand and Julie Dorsey. Interactive tone mapping. In *Rendering Techniques 2000*, pages 219–230. Springer, 2000.
- [33] M. Ebdelli, O. Le Meur, and C. Guillemot. Video inpainting with short-term windows: Application to object removal and error concealment. *IEEE Trans. on Image Proc.*, 24(10):3034–3047, October 2015.

- [34] D. S. Ebert and Richard E. Parent. Rendering and animation of gaseous phenomena by combining fast volume and scanline A-buffer techniques. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 357–366, New York, NY, USA, 1990. ACM Press.
- [35] Shimon Edelman. Visual perception. *Encyclopedia of Artificial Intelligence*, pages 2:1655–1663, 1992.
- [36] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038 vol.2, 1999.
- [37] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.
- [38] Gabriel Eilertsen, Jonas Unger, and Rafal K Mantiuk. Evaluation of tone mapping operators for hdr video. In *High Dynamic Range Video*, pages 185–207. Elsevier, 2016.
- [39] Douglas Enright, Stephen R. Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph.*, 21(3):736–744, 2002.
- [40] M Fahle and T Poggio. Visual hyperacuity: spatiotemporal interpolation in human vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 213(1193):451–477, 1981.
- [41] Petros Faloutsos, Michiel Van de Panne, and Demetri Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 251–260. ACM, 2001.
- [42] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 15–22, August 2001.
- [43] James A Ferwerda, Sumanta N Pattanaik, Peter Shirley, and Donald P Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 249–258. ACM, 1996.
- [44] J. Filip, Michal Haindl, and D. Chetverikov. Fast synthesis of dynamic colour textures. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 25–28, 0-0 2006.
- [45] James D Foley, Andries Van Dam, et al. *Fundamentals of interactive computer graphics*, volume 2. Addison-Wesley Reading, MA, 1982.

-
- [46] John M Foley. The size-distance relation and intrinsic geometry of visual space: Implications for processing. *Vision Research*, 12(2):323–332, 1972.
- [47] John M Foley and Mark E McCourt. Visual grating induction. *JOSA A*, 2(7):1220–1230, 1985.
- [48] Nick Foster and Dimitris Metaxas. Modeling the motion of a hot, turbulent gas. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 181–188. ACM Press/Addison-Wesley Publishing Co., 1997.
- [49] Alain Fournier and William T. Reeves. A simple model of ocean waves. In *Annual Conference on Computer Graphics*, volume 20, pages 75–84, 1986.
- [50] Joseph M Francos, A Zvi Meiri, and Boaz Porat. A unified texture model based on a 2-d wold-like decomposition. *IEEE transactions on signal processing*, 41(8):2665–2678, 1993.
- [51] W.T. Freeman and E.H. Adelson. Steerable filters for image processing and analysis. In *Multidimensional Signal Processing, 1991., Proceedings of the Seventh Workshop on*, page 98, September 1991.
- [52] Arthur Gill et al. Introduction to the theory of finite-state machines. 1962.
- [53] K. Gould and M. Shah. The trajectory primal sketch: a multi-scale scheme for representing motion characteristics. In *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on*, pages 79–85, June 1989.
- [54] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How not to be seen — object removal from videos of crowded scenes. *Comput. Graph. Forum*, 31(2pt1):219–228, May 2012.
- [55] Miguel Granados, Kwang In Kim, James Tompkin, Jan Kautz, and Christian Theobalt. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I*, chapter Background Inpainting for Videos with Dynamic Objects and a Free-Moving Camera, pages 682–695. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [56] M. Haindl and J. Filip. *Visual Texture*. Advances in Computer Vision and Pattern Recognition. Springer-Verlag, London, 2013.
- [57] Michal Haindl and Jiri Filip. *Visual texture: Accurate material appearance measurement, representation and modeling*. Springer Science & Business Media, 2013.
- [58] Michal Haindl and Miloš Kudělka. Texture fidelity benchmark. In *Computational Intelligence for Multimedia Understanding (IWCIM), 2014 International Workshop on*, pages 1–5. IEEE, 2014.

- [59] Robert M Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [60] Robert M Haralick, Karthikeyan Shanmugam, et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.
- [61] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings., International Conference on Image Processing*, volume 3, pages 648–651 vol.3, October 1995.
- [62] D. J. Heeger and A. P. Pentland. Seeing structure through chaos . *IEEE Workshop on Motion*, 1986.
- [63] David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, January 1988.
- [64] Jessica K Hodgins, Wayne L Wooten, David C Brogan, and James F O’Brien. Animating human athletics. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 71–78. ACM, 1995.
- [65] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.
- [66] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Trans. Graph.*, 35(6):196:1–196:11, November 2016.
- [67] ZhiYong Huang, JingJing Zhang, and QiuHong Zhang. A hybrid method for video stabilization and retargeting. In *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service, ICIMCS ’13*, pages 46–51, New York, NY, USA, 2013. ACM.
- [68] Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Lazyfluids: Appearance transfer for fluid animations. *ACM Trans. Graph.*, 34(4):92:1–92:10, July 2015.
- [69] Xiaogang Jin, Shaochun Chen, and Xiaoyang Mao. Computer-generated marbling textures: A gpu-based design system. *IEEE Computer Graphics and Applications*, 27(2):78–84, March 2007.
- [70] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Interpolation of missing data in image sequences. *IEEE Transactions on Image Processing*, 4(11):1509–1519, November 1995.
- [71] Miloš Kudělka and Michal Haindl. Texture fidelity criterion. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2062–2066. IEEE, 2016.

-
- [72] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics, SIGGRAPH*, 24:795–802, 2005.
- [73] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graph-cut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, July 2003.
- [74] Guy Le Besnerais and Frédéric Champagnat. Dense optical flow by iterative local window registration. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–137. IEEE, 2005.
- [75] Redouane LGUENSAT, Pierre TANDEO, Ronan FABLET, and Pierre Ailliot. Non-parametric ensemble kalman methods for the inpainting of noisy dynamic textures. In *ICIP 2015 : IEEE International Conference on Image Processing*, pages 2488–2492. IEEE, September 2015.
- [76] Jianghong Li, Liang Chen, and Yuanhu Cai. Dynamic texture segmentation using 3-d fourier transform. In *Image and Graphics, 2009. ICIG'09. Fifth International Conference on*, pages 293–298. IEEE, 2009.
- [77] Zicheng Liao, Neel Joshi, and Hugues Hoppe. Automated video looping with progressive dynamism. *ACM Trans. Graph.*, 32(4):77:1–77:10, July 2013.
- [78] Chia-Wen Lin and Nai-Chia Cheng. Video bckground inpainting using dynamic texture synthesis. In *ISCAS*, pages 1559–1562. IEEE, 2010.
- [79] W. C. Lin and Y. Liu. A lattice-based mrf model for dynamic near-regular texture tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):777–792, May 2007.
- [80] Wen-Chieh Lin, James Hays, Chenyu Wu, Yanxi Liu, and Vivek Kwatra. Quantitative evaluation of near regular texture synthesis algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 427–434. IEEE, 2006.
- [81] Wen-Chieh Lin, James Hays, Chenyu Wu, Yanxi Liu, and Vivek Kwatra. Quantitative evaluation of near regular texture synthesis algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 427–434. IEEE, 2006.
- [82] Wen-Chieh Lin, Yanxi Liu, et al. A lattice-based mrf model for dynamic near-regular texture tracking and manipulation. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Citeseer, 2005.
- [83] Aristid Lindenmayer. Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of theoretical biology*, 18(3):280–299, 1968.

- [84] Che-Bin Liu, Ruei-Sung Lin, and Narendra Ahuja. Modeling dynamic textures using subspace mixtures. In *ICME*, pages 1378–1381. IEEE, 2005.
- [85] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Trans. Graph.*, 30(1):4:1–4:10, February 2011.
- [86] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.
- [87] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI’81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.
- [88] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.
- [89] Thomas Luhmann, Stuart Robson, SA Kyle, and IA Harley. *Close range photogrammetry: principles, techniques and applications*. Whittles, 2006.
- [90] Simon Masnou and Jean-Michel Morel. Level lines based disocclusion. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 259–263. IEEE, 1998.
- [91] Josh H McDermott, Michael Schemitsch, and Eero P Simoncelli. Summary statistics in auditory perception. *Nature neuroscience*, 16(4):493, 2013.
- [92] Josh H McDermott and Eero P Simoncelli. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron*, 71(5):926–940, 2011.
- [93] Chris McGlone, Edward Mikhail, and Jim Bethel. *Manual of photogrammetry*. 1980.
- [94] Edward M Mikhail, James S Bethel, and J Chris McGlone. *Introduction to modern photogrammetry*. New York, 2001.
- [95] Andrew Nealen and Marc Alexa. *Hybrid texture synthesis*. Techn. Univ., Fachbereich Informatik, Fachgebiet Graphisch-Interaktive Systeme, 2003.
- [96] Randal C. Nelson and Ramprasad Polana. Qualitative recognition of motion using temporal texture. *CVGIP: Image Underst.*, 56(1):78–89, July 1992.
- [97] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Video inpainting of complex scenes. *SIAM Journal on Imaging Sciences*, 7(4):1993–2019, 2014.

-
- [98] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. Physically based modeling and animation of fire. 21(3):721–728, 2002.
- [99] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. *Geometric Considerations and Nomenclature for Reflectance*. 1977.
- [100] Supot Nitsuwat. Optical flow estimation. Technical report, Technical Report from Department of Artificial Intelligence of School of Computer Science and Engineering of the University of New South Wales, 2006.
- [101] Kedar A. Patwardhan, Guillermo Sapiro, and Marcelo Bertalmío. Video inpainting under constrained camera motion. *IEEE Trans. Image Processing*, 16(2):545–553, 2007.
- [102] C. H. Perry and R. W. Picard. Synthesizing flames and their spreading. In *Proceedings of the Fifth Eurographics Workshop on Animation and Simulation*, pages 1–14, 1994.
- [103] Renaud Péteri. Tracking dynamic textures using a particle filter driven by intrinsic motion information. *Mach. Vis. Appl.*, 22(5):781–789, 2011.
- [104] Renaud Péteri, Sándor Fazekas, and Mark J. Huiskes. DynTex : a Comprehensive Database of Dynamic Textures. *Pattern Recognition Letters*, 31(12):1627–1632, 2010. <http://projects.cwi.nl/dyntex/>.
- [105] R. Polana and R.C. Nelson. Recognition of motion from temporal texture. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 129–134, June 1992.
- [106] Ramprasad Polana and Randal Nelson. Low level recognition of human motion (or how to get your man without finding his body parts). In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 77–82. IEEE, 1994.
- [107] Jovan Popović, Steven M Seitz, Michael Erdmann, Zoran Popović, and Andrew Witkin. Interactive manipulation of rigid body simulations. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 209–217. ACM Press/Addison-Wesley Publishing Co., 2000.
- [108] William T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2:359–376, 1983.
- [109] William T Reeves and Ricki Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *ACM Siggraph Computer Graphics*, volume 19, pages 313–322. ACM, 1985.

- [110] Allan G Rempel, Matthew Trentacoste, Helge Seetzen, H David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs. In *ACM transactions on graphics (TOG)*, volume 26, page 39. ACM, 2007.
- [111] Yossi Rubner, Jan Puzicha, Carlo Tomasi, and Joachim M Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer vision and image understanding*, 84(1):25–43, 2001.
- [112] P. Saisan, G. Doretto, Ying Nian Wu, and S. Soatto. Dynamic texture recognition. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–58–II–63 vol.2, 2001.
- [113] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. pages 489–498, 2000.
- [114] Steven Seitz Seitz and Charles R. Dyer. Detecting irregularities in cyclic motion. In *in Proc. Workshop on Motion of Non-Rigid and Articulated Objects*, pages 178–185, 1994.
- [115] Yaser Sheikh, Niels C. Haering, and Mubarak Shah. Shape from dynamic texture for planes. In *Computer Vision and Pattern Recognition*, pages 2285–2292, 2006.
- [116] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, March 1992.
- [117] Karl Sims. Particle animation and rendering using data parallel computation. In *Computer Graphics*, pages 405–413, 1990.
- [118] Stefano Soatto, Gianfranco Doretto, and Ying Nian Wu. Dynamic textures. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 439–446. IEEE, 2001.
- [119] Jos Stam and Eugene Fiume. Turbulent wind fields for gaseous phenomena. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93*, pages 369–376, New York, NY, USA, 1993. ACM.
- [120] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 129–136. ACM, 1995.
- [121] Ondrej Stava, Sören Pirk, Julian Kratt, Baoquan Chen, Radomír Měch, Oliver Deussen, and Bedrich Benes. Inverse procedural modelling of trees. In *Computer Graphics Forum*, volume 33, pages 118–131. Wiley Online Library, 2014.

-
- [122] M. Strobel, J. Diebold, and D. Cremers. Flow and color inpainting for video completion. In *German Conference on Pattern Recognition (GCPR)*, Münster, Germany, September 2014.
- [123] Richard Szeliski, Richard Szeliski, David Tonnesen, and David Tonnesen. Surface modeling with oriented particle systems. In *Computer Graphics*, pages 185–194, 1991.
- [124] Martin Szummer. Temporal texture modeling. Master’s thesis, MIT Media Lab Perceptual Computing, Cambridge, MA, USA, September 1995. AAI0804614.
- [125] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *IEEE Intl. Conf. Image Processing*, volume 3, pages 823–826, September 1996.
- [126] Martin Szummer and Rosalind W. Picard. Temporal texture modeling. In *IEEE Intl. Conf. Image Processing*, volume 3, pages 823–826, September 1996.
- [127] Richard P Taylor, Adam P Micolich, and David Jonas. Fractal expressionism. *Physics World*, 12(10):25, 1999.
- [128] Demetri Terzopoulos. Artificial life for computer graphics. *Communications of the ACM*, 42(8):32–42, August 1999.
- [129] Gui Yun Tian, D. Gledhill, D. Taylor, and D. Clarke. Colour correction for panoramic imaging. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 483–488, July 2002.
- [130] Jack Tumblin and Greg Turk. Lcis: A boundary hierarchy for detail-preserving contrast reduction. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90. ACM Press/Addison-Wesley Publishing Co., 1999.
- [131] P. M. B. van Roosmalen, A. C. Kokaram, and J. Biemond. Fast high quality interpolation of missing data in image sequences using a controlled pasting scheme. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3105–3108 vol.6, March 1999.
- [132] René Vidal and Avinash Ravich. Optical flow estimation and segmentation of multiple moving dynamic textures. In *In CVPR*, pages 516–521, 2005.
- [133] Yizhou Wang and Song chun Zhu. Analysis and synthesis of textured motion: Particles and waves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1348–1363, 2004.
- [134] Yizhou Wang and Song Chun Zhu. A generative method for textured motion: Analysis and synthesis. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter

- Johansen, editors, *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part I*, volume 2350 of *Lecture Notes in Computer Science*, pages 583–598. Springer, 2002.
- [135] Zhou Wang and Alan C Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.
- [136] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [137] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [138] Zhou Wang and Eero P Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–573. IEEE, 2005.
- [139] Greg Ward. A contrast-based scalefactor for luminance display. *Graphics gems IV*, pages 415–421, 1994.
- [140] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488. ACM Press/Addison-Wesley Publishing Co., 2000.
- [141] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007.
- [142] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7101, 2017.
- [143] G. Xu and S. Ma. Robust tile-based texture synthesis using texture element. In *2010 International Conference on Electronics and Information Engineering*, volume 2, pages V2–179–V2–183, August 2010.
- [144] M. Yokota and Y. Yokota. A hypothesis of perceptual filling-in process for spatio-temporal frequency of dynamic textures. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, volume 3, pages 2784–2787 Vol.3, September 2003.
- [145] Dongxiao Zhou. *Texture analysis and synthesis using a generic Markov-Gibbs image model*. PhD thesis, ResearchSpace@ Auckland, 2006.

- [146] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame) – towards a unified theory for texture modeling. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 27(2):1–20, 1998.
- [147] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame) – towards a unified theory for texture modeling. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 27(2):1–20, 1998.
- [148] Jana Zujovic, Thrasyvoulos N Pappas, and David L Neuhoff. Structural texture similarity metrics for image analysis and retrieval. *IEEE Transactions on Image Processing*, 22(7):2545–2558, 2013.
- [149] Jana Zujovic, Thrasyvoulos N Pappas, David L Neuhoff, René Van Egmond, and Huib De Ridder. Effective and efficient subjective testing of texture similarity metrics. *JOSA A*, 32(2):329–342, 2015.

Reviewed Publications of the Author Relevant to the Thesis

- [A.1] Richtr, R. and Haindl, M. Dynamic Texture Similarity Criterion, In *the 24th International Conference on Pattern Recognition (ICPR)*. IEEE, August 2018.
- [A.2] Haindl, M. and Richtr, R. Dynamic Texture Enlargement, In *Proceedings of the 29th Spring Conference on Computer Graphics*. SCCG '13, pages 5-12, New York, NY, USA, 2013. ACM.

The paper has been cited in:

- Havlíček, M. et al. Dynamic textures modeling using temporal mixing coefficients reduction. In: *Computational Intelligence for Multimedia Understanding (IWCIM)*, 2014 International Workshop on. IEEE, 2014. p. 1-5.
- [A.3] Richtr, R. and Haindl, M. Dynamic Texture Editing, In *Proceedings of the 31th Spring Conference on Computer Graphics*. SCCG '15, pages 133-140, New York, NY, USA, 2015. ACM.

Journal papers under review:

- [A.4] Richtr, R. and Haindl, M. Dynamic Texture Inpainting, In: *IEEE Transactions on Image Processing*.

Remaining Publications of the Author Relevant to the Thesis

- [A.5] Radek Richtr *Dynamic Textures Modeling*. Ph.D. Minimum Thesis, Faculty of Information Technology, Prague, Czech Republic, 2013.
- [A.6] Richtr, R. and Haindl M. *Dynamic Texture Editing*. Poster at 7th International Workshop Data - Algorithms - Decision Making, Mariánská, Czech Republic, 2011.

Inpaintig Method Comparison

Table A.1: *Comparisons with some state-of-the-art video and DT inpainting algorithms based on [66]* (drawbacks are denoted by \ominus , benefits by \oplus and \odot refer to an alternative, but not a drawback, method). The visibility assumption refers to each missing pixel needing to be visible in at least one frame. Warping implicates adjustment to the original DT. Note that due larger patches, we do not conserve our spatiotemporal synthesis unit like limitations due to preserving of the local OF and thus is marked as \odot . Moreover, non-preservation could lead to local inconsistencies (i.e. in strongly structured DTs with self-shading, self-occluding etc.)

Method	Granados[55]	Granados[54]	Newson[97]	Strobel[122]	Huang[66]	Ebnel[33]	Ours	Xie[142]
Synthesis unit	Spatial \oplus	S.-T. \ominus	S.-T. \ominus	Spatial \oplus	Spatial \oplus	Spatial \oplus	S.-T. \odot	S.-T. \ominus
Optimization	Graph-cut \oplus	Graph-cut \oplus	Hard-EM \oplus	Greedy \ominus	Hard-EM \oplus	Short t. w. \ominus	Dyn. prog. \oplus	ConvNet \ominus
Dynamic backgr.	No \ominus	Yes \oplus	Yes \oplus	No \ominus	Yes \oplus	Yes \oplus	Yes \oplus	Yes \oplus
Visibility asump.	Yes \ominus	No \oplus	No \oplus	Yes \ominus	Yes \ominus	No \oplus	No \oplus	No \oplus
Warping	Homography \ominus	N/A \oplus	Affine \ominus	N/A \oplus	Dense F.F. \ominus	Homography \ominus	N/A \oplus	N/A \oplus
Temporal consist.	No \ominus	No \ominus	No \ominus	Yes \oplus	Yes \oplus	Yes \oplus	Yes \oplus	Yes \oplus

Psycho-physical tests values

Table B.1: *Results of psycho-physical test on pairs of DTs GRASS and FOLIAGE. Range from 0 (different) to 1 (similar).*

	0	1	2	3	4	10	11	12	13	14	20	21	22	24	30	31	34
0	0,46	0,68	0,65	0,4	0,26	0,44	0,45	0,39	0,38	0,65	0,61	0,18	0,45	0,53	0,48	0,48	
1		0,48	0,48	0,31	0,17	0,28	0,48	0,44	0,22	0,46	0,22	0,2	0,26	0,73	0,31	0,55	
2			0,61	0,55	0,19	0,44	0,4	0,41	0,62	0,5	0,41	0,14	0,4	0,4	0,21	0,5	
3				0,35	0,17	0,17	0,75	0,4	0,24	0,51	0,31	0,24	0,28	0,55	0,28	0,66	
4					0,46	0,35	0,4	0,77	0,53	0,84	0,55	0,35	0,33	0,6	0,77	0,51	
10						0,4	0,2	0,28	0,66	0,28	0,46	0,66	0,46	0,24	0,46	0,2	
11							0,17	0,31	0,48	0,2	0,31	0,44	0,4	0,26	0,4	0,24	
12								0,48	0,22	0,6	0,26	0,17	0,28	0,46	0,26	0,66	
13									0,46	0,6	0,48	0,26	0,33	0,6	0,48	0,68	
14										0,37	0,88	0,51	0,31	0,77	0,26		
20											0,44	0,26	0,24	0,57	0,44	0,64	
21												0,35	0,35	0,44	0,77	0,37	
22													0,71	0,28	0,33	0,17	
24														0,57	0,48	0,42	
30															0,4	0,68	
31																0,35	

Similarity criterion values

Table C.1: *Presented dynamic texture similarity criterion values*; Bottom triangle: The proposed texture similarity criterion values scaled to range (0;1); Upper triangle: Difference between test results (Tab. B.1) and our criterion. Positive values suggest an underestimation of the criterion and subsequently negative values suggest overstatement. The total mean variation over the values is 0,1431.

	0	1	2	3	4	10	11	12	13	14	20	21	22	24	30	31	34
0	X	0,1	0,35	0,32	0,07	0,08	0,18	0,1	0,08	0,11	0,29	0,32	-0,06	0,15	0,21	0,18	0,15
1	0,38	X	0,05	0,14	-0,04	0,02	0,01	0,13	-0,02	0,06	-0,08	-0,08	-0,04	-0,06	0,39	-0,01	0,2
2	0,35	0,45	X	0,29	0,19	0,01	0,17	0,09	0,08	0,22	0,12	0,16	-0,09	0,07	0,06	-0,11	0,14
3	0,34	0,35	0,33	X	-0,15	0,1	-0,02	0,13	-0,02	0,02	-0,04	0,13	0,07	-0,13	0,09	-0,08	0,16
4	0,34	0,36	0,37	0,51	X	0,33	0,11	-0,11	0,39	0,28	0,35	0,3	0,12	-0,04	0,19	0,42	0,06
10	0,19	0,19	0,19	0,08	0,15	X	0,2	0,13	0,12	0,46	0,15	0,28	0,49	0,3	0,08	0,29	0,07
11	0,28	0,28	0,29	0,21	0,25	0,21	X	-0,03	0,06	0,23	-0,04	0,06	0,22	0,13	0,02	0,13	0
12	0,37	0,49	0,32	0,63	0,52	0,08	0,22	X	0,1	-0,05	0	0,01	0,01	-0,13	-0,01	-0,12	0,15
13	0,33	0,33	0,34	0,43	0,4	0,18	0,26	0,4	X	0,19	0,18	0,23	0,01	0	0,21	0,15	0,29
14	0,29	0,25	0,42	0,23	0,27	0,22	0,27	0,28	0,28	X	0,1	0,54	0,28	0,24	0,05	0,5	-0,01
20	0,38	0,41	0,39	0,56	0,51	0,15	0,25	0,61	0,43	0,28	X	0,21	0,05	-0,18	0,11	-0,02	0,15
21	0,3	0,31	0,26	0,19	0,27	0,2	0,26	0,27	0,27	0,36	0,25	X	0,13	0,1	0,2	0,5	0,13
22	0,25	0,25	0,25	0,19	0,24	0,19	0,23	0,18	0,26	0,24	0,23	0,24	X	0,48	0,06	0,1	-0,04
24	0,32	0,33	0,34	0,43	0,38	0,18	0,28	0,43	0,34	0,29	0,43	0,26	0,24	X	0,2	0,15	0,03
30	0,39	0,35	0,35	0,48	0,42	0,17	0,26	0,49	0,4	0,27	0,47	0,25	0,24	0,39	X	0,04	0,24
31	0,32	0,34	0,34	0,38	0,37	0,19	0,28	0,4	0,35	0,29	0,47	0,29	0,24	0,35	0,37	X	-0,02
34	0,35	0,37	0,37	0,51	0,46	0,14	0,26	0,53	0,41	0,28	0,51	0,25	0,23	0,4	0,46	0,38	X

Table C.2: *STSIM-I and STSIM-II criteria values* (average of the first 60 frames of each DT) of the representative DT subsets; Upper triangle: STSIM-I values; Bottom triangle: STSIM-II values; Note low deviation of values.

	0	1	2	3	4	10	11	12	13	14	20	21	22	24	30	31	34
0	X	0,77	0,65	0,79	0,76	0,78	0,79	0,8	0,74	0,78	0,78	0,77	0,73	0,79	0,76	0,8	0,78
1	0,75	X	0,74	0,76	0,74	0,75	0,76	0,79	0,74	0,73	0,74	0,74	0,76	0,77	0,77	0,73	0,75
2	0,71	0,74	X	0,63	0,62	0,63	0,64	0,66	0,64	0,61	0,62	0,65	0,69	0,65	0,71	0,6	0,61
3	0,76	0,75	0,71	X	0,77	0,78	0,8	0,8	0,77	0,77	0,77	0,74	0,75	0,81	0,77	0,79	0,79
4	0,74	0,74	0,7	0,75	X	0,75	0,78	0,77	0,76	0,72	0,76	0,69	0,75	0,77	0,75	0,75	0,78
10	0,75	0,74	0,7	0,76	0,74	X	0,79	0,78	0,77	0,78	0,76	0,75	0,75	0,79	0,77	0,78	0,79
11	0,76	0,75	0,71	0,77	0,75	0,76	X	0,8	0,79	0,77	0,78	0,74	0,77	0,81	0,79	0,79	0,81
12	0,76	0,76	0,71	0,76	0,75	0,75	0,76	X	0,76	0,76	0,79	0,76	0,75	0,79	0,76	0,77	0,78
13	0,74	0,74	0,71	0,75	0,74	0,75	0,76	0,75	X	0,73	0,75	0,68	0,75	0,8	0,8	0,75	0,8
14	0,75	0,73	0,69	0,75	0,73	0,75	0,75	0,74	0,73	X	0,75	0,79	0,7	0,78	0,75	0,8	0,77
20	0,75	0,74	0,7	0,75	0,74	0,74	0,75	0,75	0,74	0,74	X	0,73	0,73	0,78	0,75	0,77	0,77
21	0,75	0,74	0,7	0,74	0,71	0,74	0,73	0,74	0,71	0,75	0,73	X	0,66	0,73	0,71	0,76	0,72
22	0,74	0,75	0,73	0,75	0,74	0,74	0,75	0,74	0,74	0,72	0,73	0,71	X	0,79	0,81	0,72	0,78
24	0,76	0,75	0,71	0,77	0,75	0,76	0,77	0,76	0,76	0,75	0,75	0,74	0,76	X	0,81	0,8	0,78
30	0,75	0,75	0,73	0,76	0,74	0,75	0,76	0,75	0,77	0,74	0,74	0,73	0,77	0,77	X	0,76	0,79
31	0,76	0,74	0,69	0,76	0,74	0,75	0,76	0,75	0,74	0,76	0,74	0,74	0,73	0,76	0,75	X	0,79
34	0,75	0,74	0,7	0,76	0,75	0,76	0,77	0,76	0,76	0,75	0,75	0,73	0,76	0,77	0,76	0,76	X

Table C.3: *STSIM-I and STSIM-II comparison to psychophysical test*: Difference of psycho-physical test to between STSIM-I (upper triangle) and STSIM-II (bottom triangle); Positive values suggest an underestimation of the criterion and subsequently negative values suggest overstatement. The average difference of absolute values are 0,349 for STSIM-I and 0,332 for STSIM-II. Note that both criteria often seem overvalued due to DTs color similarities.

	0	1	2	3	4	10	11	12	13	14	20	21	22	24	30	31	34
0	0	-0,31	0,03	-0,14	-0,36	-0,52	-0,35	-0,35	-0,35	-0,4	-0,13	-0,16	-0,55	-0,34	-0,23	-0,32	-0,3
1	-0,29	0	-0,26	-0,28	-0,43	-0,58	-0,48	-0,31	-0,3	-0,51	-0,28	-0,52	-0,56	-0,51	-0,04	-0,42	-0,2
2	-0,03	-0,26	0	-0,02	-0,07	-0,44	-0,2	-0,26	-0,23	0,01	-0,12	-0,24	-0,55	-0,25	-0,31	-0,39	-0,11
3	-0,11	-0,27	-0,1	0	-0,42	-0,61	-0,63	-0,05	-0,37	-0,53	-0,26	-0,43	-0,51	-0,53	-0,22	-0,51	-0,13
4	-0,34	-0,43	-0,15	-0,4	0	-0,29	-0,43	-0,37	0,01	-0,19	0,08	-0,14	-0,4	-0,44	-0,15	0,02	-0,27
10	-0,49	-0,57	-0,51	-0,59	-0,28	0	-0,39	-0,58	-0,49	-0,12	-0,48	-0,29	-0,09	-0,33	-0,53	-0,32	-0,59
11	-0,32	-0,47	-0,27	-0,6	-0,4	-0,36	0	-0,63	-0,48	-0,29	-0,58	-0,43	-0,33	-0,41	-0,53	-0,39	-0,57
12	-0,31	-0,28	-0,31	-0,01	-0,35	-0,55	-0,59	0	-0,28	-0,54	-0,19	-0,5	-0,58	-0,51	-0,3	-0,51	-0,12
13	-0,35	-0,3	-0,3	-0,35	0,03	-0,47	-0,45	-0,27	0	-0,27	-0,15	-0,2	-0,49	-0,47	-0,2	-0,27	-0,12
14	-0,37	-0,51	-0,07	-0,51	-0,2	-0,09	-0,27	-0,52	-0,27	0	-0,38	0,09	-0,19	-0,27	-0,44	-0,03	-0,51
20	-0,1	-0,28	-0,2	-0,24	0,1	-0,46	-0,55	-0,15	-0,14	-0,37	0	-0,29	-0,47	-0,54	-0,18	-0,33	-0,13
21	-0,14	-0,52	-0,29	-0,43	-0,16	-0,28	-0,42	-0,48	-0,23	0,13	-0,29	0	-0,31	-0,38	-0,27	0,01	-0,35
22	-0,56	-0,55	-0,59	-0,51	-0,39	-0,08	-0,31	-0,57	-0,48	-0,21	-0,47	-0,36	0	-0,08	-0,53	-0,39	-0,61
24	-0,31	-0,49	-0,31	-0,49	-0,42	-0,3	-0,37	-0,48	-0,43	-0,24	-0,51	-0,39	-0,05	0	-0,24	-0,32	-0,36
30	-0,22	-0,02	-0,33	-0,21	-0,14	-0,51	-0,5	-0,29	-0,17	-0,43	-0,17	-0,29	-0,49	-0,2	0	-0,36	-0,11
31	-0,28	-0,43	-0,48	-0,48	0,03	-0,29	-0,36	-0,49	-0,26	0,01	-0,3	0,03	-0,4	-0,28	-0,35	0	-0,44
34	-0,27	-0,19	-0,2	-0,1	-0,24	-0,56	-0,53	-0,1	-0,08	-0,49	-0,11	-0,36	-0,59	-0,35	-0,08	-0,41	0