

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kučera** Jméno: **Michal** Osobní číslo: **420953**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**  
Studijní program: **Otevřená informatika**  
Studijní obor: **Počítačová grafika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Kvalitativní srovnání metod pro přenos výtvarného stylu**

Název diplomové práce anglicky:

**Qualitative comparison of methods for example-based style transfer**

Pokyny pro vypracování:

Prostudujte techniky pro přenosu výtvarného stylu, které využívají řízenou syntézu textur [1, 2] a hluboké neuronové sítě [3, 4, 5]. Algoritmy klasifikujte do skupin podle požadavků na vstupní data a popište jejich základní vlastnosti. Vycházejte přitom ze studie [6]. V dalším se zaměřte zejména na techniky, které lze použít v rámci paradigmatu obrazové analogie [7]. Ve spolupráci s vedoucím práce navrhnete sadu vstupních dat, na kterých bude možné provést kvalitativní srovnání formou percepčního experimentu. Tyto experimenty proveďte a na základě jejich výsledků posuďte kvalitu syntézy a pokuste se objektivně charakterizovat hlavní odlišnosti vybraných metod.

Seznam doporučené literatury:

- [1] Fišer et al.: StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings, ACM Transactions on Graphics 35(4):92, 2016.
- [2] Fišer et al.: Example-Based Synthesis of Stylized Facial Animations, ACM Transactions on Graphics 36(4):155, 2017.
- [3] Gatys et al.: Image Style Transfer Using Convolutional Neural Networks, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414-2423, 2016.
- [4] Li & Wand: Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2479-2486, 2016.
- [5] Liao et al.: Visual Attribute Transfer Through Deep Image Analogy, ACM Transactions on Graphics 36(4):120, 2017.
- [6] Semmo et al.: Neural Style Transfer: A Paradigm Shift for Image-based Artistic Rendering? Proceedings International Symposium on Non-Photorealistic Animation and Rendering, 2017.
- [7] Hertzmann et al.: Image Analogies, Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques, pp. 327-340, 2001.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**doc. Ing. Daniel Sýkora, Ph.D., Katedra počítačové grafiky a interakce**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **15.08.2017**

Termín odevzdání diplomové práce: **25.05.2018**

Platnost zadání diplomové práce:

**do konce letního semestru 2018/2019**

\_\_\_\_\_  
doc. Ing. Daniel Sýkora, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Ripka, CSc.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

CZECH TECHNICAL UNIVERSITY IN PRAGUE  
FACULTY OF ELECTRICAL ENGINEERING  
DEPARTMENT OF COMPUTER GRAPHICS  
AND INTERACTION



Master's thesis

# Qualitative comparison of methods for example-based style transfer

*Bc. Michal Kučera*

Supervisor: doc. Ing. Daniel Sýkora, Ph.D.

17th May 2018



---

## **Acknowledgements**

I'd like to thank all the kind people of the DCGI FEE CTU for providing me with all the equipment and help needed to finish this thesis. I'd also like to thank my family and friends for helping and motivating me to spend time doing this thesis, rather than spend time with them.



---

## Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on 17th May 2018

.....

Czech Technical University in Prague

Faculty of Electrical Engineering

© 2018 Michal Kučera. All rights reserved.

*This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Electrical Engineering. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).*

### **Citation of this thesis**

Kučera, Michal. *Qualitative comparison of methods for example-based style transfer*. Master's thesis. Czech Technical University in Prague, Faculty of Electrical Engineering, 2018.



---

## Abstrakt

V této práci se věnuji současnému stavu techniky na poli metod pro přenos výtvarného stylu, popisuji přístup různých vědeckých skupin k řešení této problematiky a porovnávám vlastnosti jejich přístupů. Tuto studii rovněž obohacuji sérií testů a experimentů s cílem porovnat kvality výstupů těchto metod.

**Klíčová slova** nefotorealistic rendering, přenos uměleckého stylu, percepční experiment

---

## Abstract

In this thesis, I provide insight into the current state-of-the-art of example-based style transfer methods, and design and perform a series of tests to compare the differences of output qualities of the studied methods.

**Keywords** non-photorealistic rendering, example-based, style transfer, perceptual experiment



---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>State-of-the-art</b>	<b>5</b>
2.1	Guided texture synthesis . . . . .	5
2.2	Neural networks . . . . .	12
2.3	Summary . . . . .	14
2.4	Deep Image Analogy . . . . .	15
<b>3</b>	<b>Test design</b>	<b>17</b>
3.1	Test Objectives . . . . .	17
3.2	Dataset creation . . . . .	17
3.3	Perceptual experiment . . . . .	24
3.4	Online survey . . . . .	25
<b>4</b>	<b>Test results</b>	<b>29</b>
4.1	Perceptual experiment . . . . .	29
4.2	Online survey . . . . .	34
4.3	Performance measurement . . . . .	37
4.4	Summary . . . . .	40
	<b>Conclusion</b>	<b>43</b>
	<b>Bibliography</b>	<b>45</b>
<b>A</b>	<b>Acronyms</b>	<b>49</b>
<b>B</b>	<b>Contents of enclosed CD</b>	<b>51</b>



---

## List of Figures

1.1	Style transfer example . . . . .	1
1.2	Style transfer to photo . . . . .	3
1.3	Style to style transfer . . . . .	3
1.4	Photographic look transfer . . . . .	3
1.5	Photo to photo . . . . .	3
2.1	Hertzmann et al. – Image analogy . . . . .	5
2.2	The Lit Sphere . . . . .	7
2.3	Comparison of guiding channels . . . . .	9
2.4	LPE guiding channels . . . . .	10
2.5	Guiding channels of Fišer et al. 2017 . . . . .	10
2.6	<i>StyleBlit</i> - chunk transfer . . . . .	12
2.7	Neural-based approach to parametric synthesis example . . . . .	13
3.1	Testing dataset format . . . . .	18
3.2	Source styles . . . . .	19
3.3	Deep Image Analogy results . . . . .	21
3.4	Fast neural style results . . . . .	23
3.5	Survey system layout - Survey choice . . . . .	26
3.6	Survey system layout - Introduction page . . . . .	27
3.7	Survey system layout - Question page . . . . .	27
3.8	Survey system layout - Question answered . . . . .	27
3.9	Survey system layout - Data submission . . . . .	28
4.1	Deep Image Analogy execution times . . . . .	38



---

## List of Tables

4.1	Perceptual experiment - summary . . . . .	30
4.2	Perceptual experiment - $H_0$ rejection probability . . . . .	30
4.3	Perceptual experiment - <i>StyLit</i> vs. <i>StyleBlit</i> results . . . . .	30
4.4	All methods survey – summary . . . . .	34
4.5	All methods survey – comparison results . . . . .	35
4.6	All methods survey – t-test evaluation . . . . .	35
4.7	<i>StyLit</i> <i>StyleBlit</i> survey – summary . . . . .	36
4.8	Deep Image Analogy - Ratio's impact on execution time . . . . .	37
4.9	Deep Image Analogy – Blending Weight's impact on execution time	38





# Introduction

This thesis aims to conduct a series of perceptual experiments that would provide insight into how human observers perceive current state-of-the-art in the field of methods that perform example-based style transfer. Even though similar research already exists [1], it focuses on neural networks only. While those methods have been commercially more successful in the past based on the number of applications (*Prisma* [2], *Deepart.io* [3]), methods using guided patch-based synthesis gained undeniable significance especially in the latest research, which is why a description of their capabilities is necessary along with qualitative comparison of their outputs with their neural network based counterparts. The algorithms of example-based transfer, both those being based on guided patch-based synthesis and neural networks, share the same workflow. The user generally presents the algorithm with 2 inputs: a **source content** and **source style**. The algorithm then returns 1 image as an output, which is an image of the content given in the input, but stylized to look like it was painted in the artistic style represented by the given input style, as presented in the Figure 1.1.

This description is, unfortunately, rather vague as there is no real definition

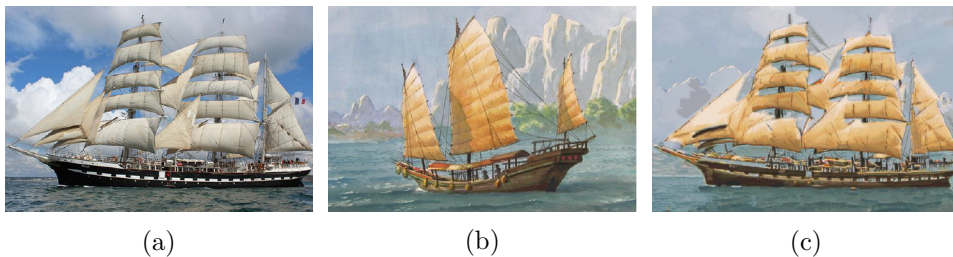


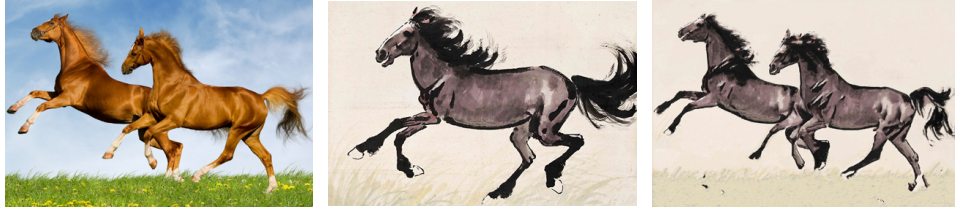
Figure 1.1: Example of style transfer [4]. Image (a) shows source content, image (b) source style and (c) shows the final output, with the content from (a) but painted in the style of (b). The painting used as the style is *A South China Junk* by Chung Chee Kit

what the artistic style really is. It could be the palette of used colors, color structures left by the used medium, the deformation of a person or an object portrayed, the overall composition, or it could be any combination of all the previously mentioned aspects. This is generally decided by the author of the painting and an independent observer cannot unambiguously define it which means that neither can any algorithm. The authors of the style transfer methods usually use a definition „make it look like the given style“ [5], which does not make the meaning any less unambiguous, but it is a simplification that makes sense and it is the definition I will use in this thesis.

All the variations for possible input contents and styles are, in the general case, virtually endless. Their limitations come from the requirements of each individual method or its specific implementation. For example, the method proposed by Liao et al. 2017 [4] requires the two given inputs to contain semantically similar content for the style transfer process to produce meaningful output. Most common variations of the inputs can, however, be categorized into several groups:

- **Style transfer to photo** (Figure 1.2). The most common case, where we present the algorithm with a photo representing the content, and style image (e.g. a painting) representing the style. As an output, we expect an image of the given content, stylized to represent the given style.
- **Style to style** (Figure 1.3), where two style images are presented (e.g. two paintings), one as a content input and the other as a style input. As a result, we expect the content painting to be stylized as the given style image.
- **Photographic look transfer** (Figure 1.4), mostly an experimental usage and by far the most difficult. In this case, we present the algorithm with, for example, a drawing and we provide it with a photo as the input style. This way, the algorithm can produce a realistic looking image from a sketch of what its content should be. This usage can be seen in Herzmann et al. [5], where they managed to produce realistic landscapes from a drawing of its layout and a dissected image of an existing landscape. Liao et al. 2017 [4] were also successful with their algorithm, using it to create human faces just from a sketch and a photo of someone else’s face.
- **Photo to photo** (Figure 1.5). Mostly a color transfer usage, able to change a photo’s spectrum to match the spectrum of another photo, in a semantically meaningful manner.

It should also be said that the style transfer is not limited to just images, but can also be applied to videos. Same categories apply in this case with their video substitutes.



(a)

(b)

(c)

Figure 1.2: Style transfer to photo.



(a)

(b)

(c)

Figure 1.3: Style to style.



(a)

(b)

(c)

Figure 1.4: Photographic look transfer.



(a)

(b)

(c)

Figure 1.5: Photo to photo.

The style transfer algorithms already have several applications on the market, making this field commercially successful. As mentioned before, the smartphone application *Prisma* and the website *Deepart.io* are among those applications. The progress done by Fišer et al. 2017 [6], for example, allows the user to tune minor details in style transfer of videos, such as temporal coherence, which therefore makes the algorithm appealing for movie studios as a visual effect tool. Such a tool can be a vital component in the creation of movies with artistic styles too complex and difficult for modern CGI, for instance the *Loving Vincent*, a fully painted feature film.

The number of methods presented by various researchers lead to the need to conduct research regarding their quality and features in comparison to each other. It is obvious that there is not a clearly superior method solving this problem, as each presents their own advantages and limitations (described and discussed in Chapter 2), as well as requirements on their platform. The goal of this thesis, besides describing the current state-of-the-art, is to design and execute various perceptual experiments to find such limitations and to describe differences in the possible output quality, backed by data collected in the most unbiased way achievable. These tests are described in Chapter 3, and their results are further discussed in Chapter 4.

---

## State-of-the-art

### 2.1 Guided texture synthesis

#### 2.1.1 Image Analogies

In 2001, Hertzmann et al. introduced a new method named „Image Analogies“ [5]. In this method, the authors described a new approach to define both simple and more advanced image filters by finding analogies in presented images by solving the following problem:

Given a pair of images  $A$  and  $A'$  (the unfiltered and filtered source images) along with some additional unfiltered target image  $B$ , synthesize a new filtered target image  $B'$  such that

$$A : A' :: B : B'$$

This means finding an image  $B'$  that relates to  $B$  the same way  $A'$  relates to  $A$ , as shown on example in Figure 2.1. In terms of the definition used in

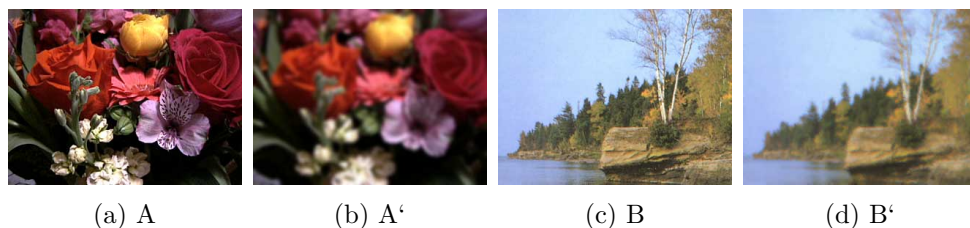


Figure 2.1: Image analogy as defined by Hertzmann et al.:  $A'$  relates to  $A$  the same way  $B'$  relates to  $B$ . In this case,  $A'$  is a blurred version of  $A$ . When these 2 are provided, along with the image  $B$ , image  $B'$  can be computed by using these defined analogies without the need to use the same filter applied to  $A'$ , or even needing to know what filter it is.

the introduction, image  $B$  was the source content,  $A'$  the source style,  $B'$  the final output, and  $A$  was a guiding channel, giving the algorithm information about both source images.

This approach was the first stepping-stone in the field of example-based style transfer, even though the research was not solely focused on artistic styles. The idea was to provide an easy alternative for common filters such as blur, sharpen, emboss, etc. by providing an example of the filter rather than coding each one of them. The resulting framework was not only able to do that, but it was also capable of advanced filters, such as artistic style transfer, super-resolution (resizing an image to a higher resolution with a smarter mean of new element interpolation), texture transfer, improved texture synthesis, and texture-by-numbers.

The method converts the problem into an optimization problem. Let  $x_t$  be a position of a pixel in the target image and  $x_s$  a position of the pixel in the source image.  $A(x_s)$  would then be the value of a pixel in source image  $A$  on position  $x_s$ . While a basic image would, in this case, return a 3-dimensional vector containing each color component, in general the returned value could be an any-dimensional vector containing any supplied channel. Additionally, we require a distance metric  $D$ , which returns a distance value between two given vectors representing the guiding channels values. The whole process then works like this:

```
Data: Images  $A$ ,  $A'$  and  $B$   
Result: Stylized image  $B'$   
foreach  $x_t \in B$  in scan-line order do  
  |  $x_s = \text{BestMatch}(A, A', B, B', q)$   
  |  $B'(x_t) = A'(x_s)$   
end
```

**Algorithm 1:** Image Analogies [5] framework pseudocode

The process uses the function *BestMatch* to find a pixel in  $A$  that is closest to the examined pixel in  $B$  by the given metric  $D$ . The actual framework does some more steps in the process, such as using a coarse to fine approach and iteratively repeating this process from a smaller version of the images up to the full resolution to provide coherence with neighbouring pixels as well.

### 2.1.2 The Lit Sphere

In parallel with the research of Hertzmann et al., Sloan et al. introduced their own contribution to the fields of non-photorealistic rendering and example-based style transfer called *The Lit Sphere* [7]. In their paper, the authors recognize the fact that artists often start with a shading study on a simple sphere, deciding the properties of the light and using it as a guide for later, as shown in Figure ?? . Inspired by this, the authors introduced their framework which used the normals of the sphere and normals of a target scene to guide

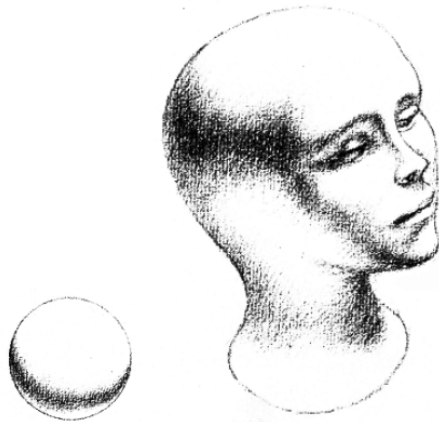


Figure 2.2: The Lit Sphere example. Source [7]

the style transfer in a similar way to the *Image Analogies*, which used RGB as guidance.

Another contribution of *The Lit Sphere* was the feature which allowed the user to create these spherical examples from existing images, storing the texture on a sphere as a bi-product and applying the texture to arbitrary 3D models.

### 2.1.3 Temporally Coherent Image Analogies

In 2013, Bénard et al. [8] introduced their improvement of the method *Image Analogies* by Hertzmann et al. [5], extending the original approach to animations. The idea was that given a video sequence in certain form and given a stylized version of a selection of frames, the frames between the given ones could be computed algorithmically, reducing unwanted noisy artifact such as temporal flickering, and effectively reducing the time required to create the animation while still allowing the artist to have full control over the stylization process by constraining the output by the author’s stylized input.

To achieve this result, Bénard et al. extended the *Image Analogies* by introducing a series of terms that influence an important factor of the style transfer. This way, the user can set the weights of these terms to influence the process and the final output. The algorithm then repeatedly evaluates the goal functions and tries to improve the final solution. The introduced terms are similar to those of *image Analogies* (now taking rotation into consideration), and in addition, the authors added a temporal coherence term preventing unnecessary or sudden changes, and histogram term preventing repeated patterns.

The final implementation is able to produce a visually pleasing result with

variable temporal flickering as required by the user. The authors use a coarse-to-fine version of PatchMatch [9] to optimize their solution and with this acceleration the method is able to compute a FullHD 1080p animation at the rate of 10-12 minutes per frame.

#### 2.1.4 StyLit

In the years after the paper „Image Analogies“, multiple research papers further elaborated on the field of example-based transfer, but unfortunately, many still suffered from many undesirable artifacts such as many wash-out effects - overusage of cheap source patches causing reduction of many important details. As described by Newson et al. [10], these effects were mainly caused by the energy function used which did not restrict excessive usage of the cheapest patch. New approaches aiming to mitigate this problem were introduced, e.g. the papers by Kaspar et al. [11] or Jamriška et al. [12], but both were assuming uniform source patch usage, which is not the general case. In 2016, Fišer et al. [13] introduced an approach called *StyLit* that featured an adaptive mechanism to prevent the observed wash-out effects.

To keep the most of the elements left by the used medium and prevent other artifacts, *StyLit* implements this mechanism to encourage uniform patch usage similar to Kaspar et al. [11] and Jamriška et al. [12], but it has been shown by Fišer et al. that the mechanism provides wrong results if the distribution of lighted areas is different between the source and target scenes. For example, if the source scene would have a large shadowed area, but the target scene would have very small shadow area, the uniform patch usage mechanism would still force the shadowed areas from source style somewhere into the final output, which would produce artifacts. To mitigate this effect, *StyLit* uses the upgraded mechanism with an error budget allowing the uniform patch to restart and reuse some patches instead of forcing them into wrong locations.

Another contribution of the *StyLit* method was the introduction of new guidance channels: **Light path expressions**, or LPEs for short. There are several arguments for using LPEs instead of RGB channels or geometrical normals. The most important one is that LPEs can be used to guide graphical elements that the former guidance channels cannot, such as shadows which cannot be transferred when using normals as guidance, as shown in Figure 2.3. The other is that artists often use light propagation in the painted scene as an important factor in stylizing specific areas, for example, an artist would use a different stroke, brush, or color depicting differently shaded areas. This attention to detail is often washed out during the style transfer process, and methods so far produced synthetic looks which were specific to the used method. This leads to the need to create a mechanism encouraging the algorithm to use larger patches of the source



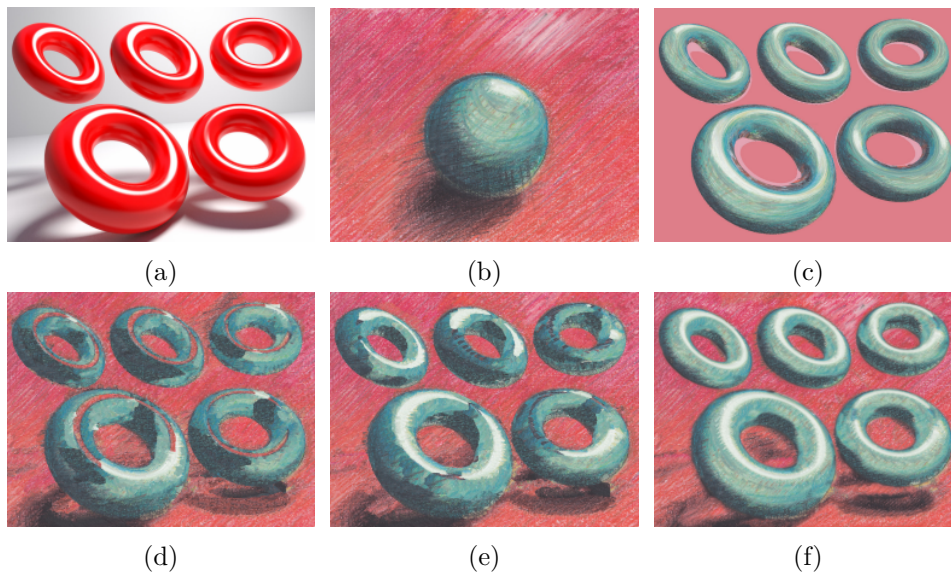


Figure 2.3: Comparison of used guiding channels for the style transfer. (a) shows the source content and (b) the source style. (c) is the algorithm of Sloan et al. [7], which uses the normals as guidance. (d) is the algorithm of Hertzmann et al. [5] which uses RGB as the guiding channel. (e) is the same algorithm, but this time using LPEs as guiding channels. (f) is the optimized algorithm by Fišer et al., which also uses LPEs. All these images were taken from the StyLit paper [13]

style (with all the contained details) while avoiding possible errors of this approach.

The *StyLit* method uses 4 guiding channels for the process: direct diffuse ( $LDE$ )<sup>1</sup>, direct specular ( $LSE$ ), first two diffuse bounces ( $LD\{1,2\}E$ ), and a diffuse interreflection ( $L.*DE$ ). More channels with additional guiding information are also possible. The four described channels are shown in Figure 2.4.

### 2.1.5 FaceStyle

In 2017, the majority of new methods for example-based style transfer were using neural networks as a mean of guidance. Sadly, many of the main disadvantages of these methods were still present. The method by Selim et al. [14] was suffering from artifacts from misalignment of the source and target faces, and along with other popular neural-based methods such as those proposed by Gatys et al. [15] or Johnson et al. [16] suffered from

<sup>1</sup>  $LDE$  and other similar strings used are regular expressions expressing the path of the light in the scene.  $LDE$  describes all rays of light that travel from the light source (L), reflect diffusely exactly once (D) and then immediately end in the eye/camera (E).

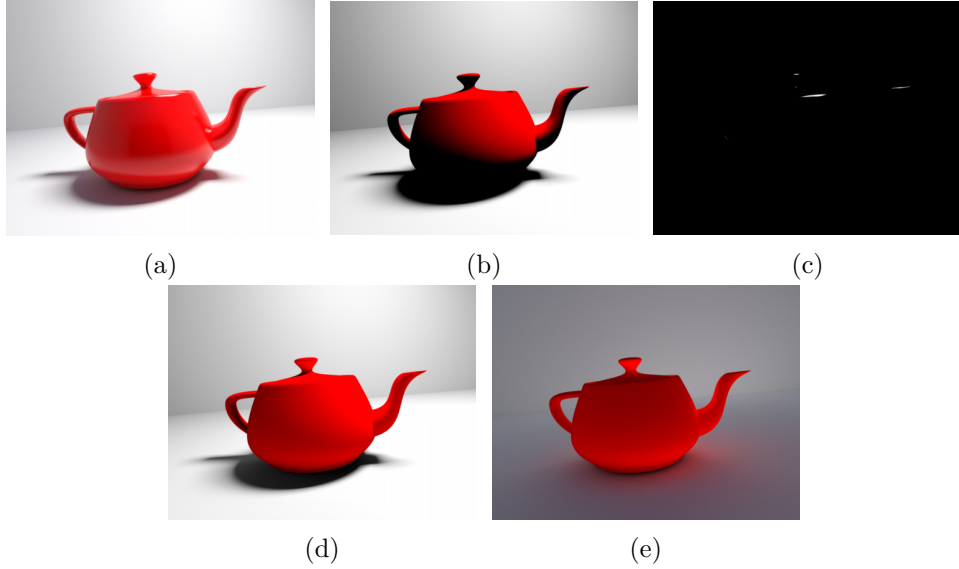


Figure 2.4: The LPE guiding channels used by *StyLit* [13] as shown on the Utah teapot model (a). These are: (b) direct diffuse ( $LDE$ ), (c) direct specular ( $LSE$ ), (d) first two diffuse bounces ( $LD\{1,2\}E$ ) and (e) a diffuse interreflection ( $L.*DE$ )

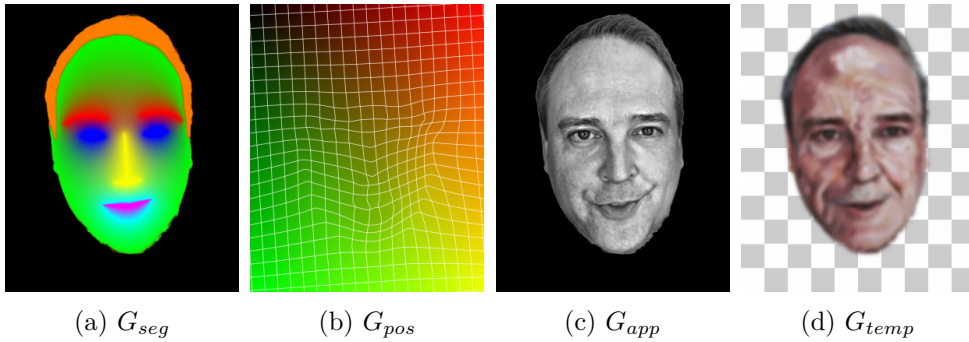


Figure 2.5: The four guiding channels used by the method by *FaceStyle* [6].

serious distortion of detailed textural information. Fišer et al. proposed a new method named „FaceStyle“ [6], following up on their research from the previous year and applying their *StyLit* algorithm to a new set of generated channels. In this research, Fišer et al. address the fact that guided texture synthesis lacks the universal nature of neural-based algorithms. While guided texture synthesis requires the user to provide all the guiding channels, neural-based algorithm constrains the synthesis with information from neural networks trained on object recognition. In the previous research of Fišer et al., they used LPEs to guide the synthesis, but a 3D model of the scene is required for those, and a 3D model is not easy to obtain for a

general scene. *FaceStyle* presents a new set of guidance channels, which can be successfully used to guide the synthesis of human faces, and which all can be generated automatically without the need to provide them separately.

There are a total of 4 channels used for the process, which are shown and described in Figure 2.5. From left to right, they are:  $G_{seg}$ , segmentation guide dividing both the input images into regions containing certain facial features;  $G_{pos}$ , a positional guide computed from the segmentation guide describing the positional distortion from the style face to the content face;  $G_{app}$ , appearance guide helping the process to produce correct facial details, such as shadows, mouth, and eyes; lastly  $G_{temp}$ , the temporal guide containing blurred stylization of the previous frame, helping the algorithm to keep temporal coherence when used on video sequences, and also allowing the user to influence the amount of temporal flickering.

*FaceStyle* also addressed the problem with temporal flickering when running a style transfer on a video sequence. Many methods have fixed temporal coherence that is specific to the implementation, but it is mostly a full temporal coherence, such as the one that can be observed with the method by Selim et al. [14]. While full temporal coherence is not wrong, artists require the option to influence the amount of temporal flickering. Full temporal coherence would make the resulting video sequence look like a texture transfer on a 3D object, but lower temporal coherence means more flicker (change) between frames simulating a hand-painted look since painting each frame separately would produce at least some amount of change. Such effect can be observed for example in a feature movie mentioned in the introduction, the *Loving Vincent*. *FaceStyle* introduces an option to influence the amount of temporal flickering, as the temporal coherence can be influenced by blurring the temporal coherence guide  $G_{temp}$ .

### 2.1.6 StyleBlit

Following their research, Fišer et al. noted that the texture coherence term of the energy function of their optimization-based *StyLit* method, along with the adaptive mechanism that prevents overuse of cheap source patches, leads the algorithm to converge to a solution where large chunks of source style are used and pasted directly into target image, as shown in Figure 2.6. This is based on the fact that within these chunks the textural coherent has no error and contains all the details the source style has, and the textural coherence error can be observed only on the borders of these chunks. This motivated Fišer et al. to create their new method *StyleBlit* [17], which seeks out these chunks by using computationally inexpensive pixel-wise operations rather than expensive patch-based optimization. This approach was meant to work with local guidance channels such as normals, as seen in the approach of Sloan et al. [7] or the guidance channels presented in their previous work *FaceStyle* [6].

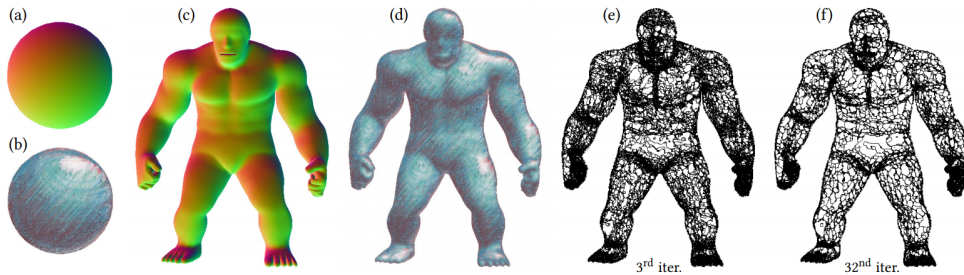


Figure 2.6: *StyleBlit* - chunk transfer. Given a source scene (a), target scene (c) described by their normal values, algorithm *StyLit* transfers the given source style (b) onto the target scene, resulting in (d). As shown in (e) and (f), the algorithm converges into state where it uses larger chunks of source style to maximize textural coherence. Image source [17]

In this method, a random pixel from the target image is chosen and all pixels in the chosen pixel’s area with their guidance values within a given error threshold are added to the selection, which forms the resulting chunk. Once the chunk’s shape is estimated, the pixel values from the source style that are within the chunk are copied into the result. The authors note that it is expected that there would be visible seams around the borders of the pasted chunks, but it appears these seams are either not visible at all (due to the nature of hand-drawn artistic styles), or can be easily suppressed using fast linear blending operations. While an implementation using a brute-force manner is possible, authors further accelerate their implementation by the fully parallel approach, using a hierarchy of spatially distributed seeds and lookup tables (or search trees with more complex guidance channels).

This approach could, in theory, produce results with similar visual quality to those of their previous methods, *StyLit* and *FaceStyle*, but much faster. The prototype of this method is able to transfer style to scenes in 10fps on a single-core CPU and in more than 100fps at a 4K UHD resolution on modern GPUs, which means that this method by far outperforms any other of the current state-of-the-art methods. To find out how this method stands quality-wise against its predecessor, *StyLit*, the output quality of this method is also studied in this thesis.

## 2.2 Neural networks

### 2.2.1 Neural-based approach to parametric synthesis

Before Fišer et al. addressed the issue of the necessity of providing guidance channels in 2017 [6], Gatys et al. proposed a new method of style transfer, named „A Neural Algorithm of Artistic Style“ [15], constrained by the outputs of neural networks trained on object recognition. In the years prior

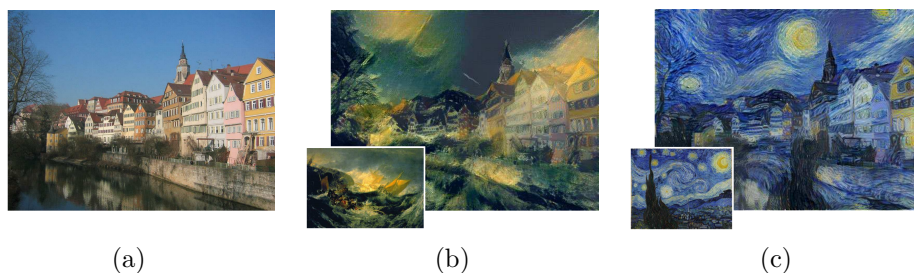


Figure 2.7: Example output of the *Neural-based approach to parametric synthesis* method

to the research paper, neural networks set the state-of-the-art in object recognition and image classification. VGG (Visual Geometry Group) of University of Oxford won several international competitions (such as the ImageNet Challenge in 2014 [18]) and proved that neural networks are capable of advanced object recognition capabilities. Their two most popular and successful networks, a 16-layer VGG-16 and a 19-layer VGG-19, are publicly available under the Creative Commons Attribution License and are widely used in style transfer algorithms such as the one proposed by Gatys et al. [15] in 2015. This method is heavily influenced by the approach of Portilla and Simoncelli from 2000 [19], which used Gabor filters for the parametrical synthesis, instead of which the *Neural Algorithm of Artistic Style* uses the responses from the VGG neural network as a parametric representation of both style and content. The final approach, however, suffers from the same drawbacks as the original approach.

In their work, the authors use the VGG-16 neural network to process both the input style and input content. Afterward, they iteratively try to synthesize an image that matches the high-level representation of the input content and also matches the style of the source style. This approach corresponds with the definition stated in the introduction that only two images are supplied as the input. After the features are extracted from the input images, a total loss (error from desired output) is computed as a linear combination of the content loss (mean squared difference from the feature representation of the content) and the style loss. In the linear combination, two values are given to the process as the weight of the content and the weight of the style, which allows the user to partially influence the look of the stylized output by giving more importance to either content or style. A random image is then generated and iteratively improved by gradient descent until the loss value converges to an acceptably small value. As the authors describe in their paper, the image at the beginning of the process can be arbitrary, but if the image is static the algorithm will be deterministic, which is the reason why a random noise is preferred.

### 2.2.2 Neural-based approach with feed-forward propagation

A year after the work of Gatys et al., Johnson et al. came with a study named „Perceptual Losses for Real-Time Style Transfer and Super-Resolution“ [16]. In this paper, they recognize one of the key disadvantages of the method *A Neural Algorithm of Artistic Style* [15], which is the fact that the method is computationally expensive, as each step of the optimization requires both forward and a backward pass through the pre-trained neural network. Even though the method produced results with reasonable quality, the low speed precluded the method to be used in any real-time applications.

Johnson et al. note the fact that many image transformation tasks can be done using a feed-forward neural network trained for that specific task, which is an approach that has been widely used in the past for tasks such as colorization, segmentation, or normal prediction. Those methods, however, only used per-pixel differences as a loss function, which is a low-level information that is not viable for high-level information, such as the image’s content. This problem can be mitigated by using high-level features from a pre-trained neural network for the loss function. In the paper, authors proposed an approach that uses feed-forward neural networks using high-level features for the loss function to quickly approximate solutions to the optimization problem in *A Neural Algorithm of Artistic Style*, and therefore getting very similar results much faster. The study of Johnson et al. did not focus solely on the style transfer task, but it experiments with the usage of feed-forward networks on single-image super-resolution as well. This part will be omitted in the rest of this thesis, as it is not particularly relevant to the topic.

It was not possible, however, to approximate the result of the *Neural Algorithm of Artistic Style* for any combination of inputs. Instead, Johnson et al. trained their feed-forward networks for a fixed input style image, which could then be applied to any content image. This meant that for each input style, a user would have to first train a network approximating this transformation (style-transfer), which was a long and expensive computation, but once the network is done the application to content images was extremely fast and cheap, making it possible for low-performance devices, such as smartphones, to process the final transformation. This workflow also found its use on the market: applications, such as *Prisma* [2], where the pre-trained style models are made by the creators of the application, who further distribute them to the end users via the store.

## 2.3 Summary

As described in this chapter, there are currently 2 main approaches to the example-based style transfer: algorithms using **guided texture synthesis**, and algorithms using **neural networks**.

Guided texture synthesis methods are generally faster and allow the user to greatly affect the final output by modifying the guiding channels. *StyLit* [13] performed a successful experiment, where an artist was painting the input style and the algorithm was performing the style transfer in real time. While the quality throughout these methods is better when using Light Path Expressions as the guiding channels, the overall quality in comparison to their neural counterparts are highly subjective and are the main goal of this thesis. The main issue of these methods is the fact that obtaining LPEs as guiding channels is difficult, if not impossible, for any general input. This issue has been addressed by *FaceStyle* in 2017 [6], where they proposed an algorithm that was able to generate the guiding channels itself but was limited to human faces only.

Neural network guided algorithms are capable of creating correspondences between input images themselves and require only the two inputs, which is a feature the previous group lacks. Some input limitations are, however, still in place. The method *A Neural Algorithm of Artistic Style* [15] did not have any limitations, but the method of Selim et al. [14] was designed for human faces only and Johnson et al. [16] required expensive computation of models for new input styles, which could then be quickly applied to any content. These algorithms are also quite expensive and not suitable for real-time usage yet, as most require long processing times even on high-end GPUs as shown in Section 4.3. The methods based on neural networks provide very little in terms of possibility to control the synthesis process. Changes made to either source content or source style can influence the final output in an unforeseeable way.

## 2.4 Deep Image Analogy

Adapting the notion of „Image Analogy“ of Hertzmann et al. [5], Liao et al. came with a research paper „Visual Attribute Transfer through Deep Image Analogy“ [4] in 2017, where they described how the original concept of *Image Analogies* could be improved by the introduction of convolutional neural networks. As Johnson et al. noted in the previous year, using per-pixel difference as a loss function does not capture any perceptual loss, as it is a high-level information. Instead of using the per-pixel difference, *Deep Image Analogies* used the features extracted from a deep convolutional neural network.

Using this approach, the authors constructed a framework that was able to perform a semantically meaningful style transfer and required the user to provide only the 2 input images (it was not important which image was the source content and which the source style, as the framework performed both combinations at the same time). Their implementation uses a 19-layer VGG-19 network to obtain correspondences between the given images, as opposed

to the approach by Gatys et al. 2 years prior, where the VGG network was only used as a parametrical image representation. Due to the semantically meaningful style transfer attribute, the algorithm requires the 2 input images to contain something semantically similar (human face, a landscape, etc.).

To speed up to style transfer process, *Deep Image Analogies* used a modified version of the PatchMatch algorithm, as proposed by Barnes et al. in 2009 [9]. By applying the algorithm to the neural network feature domain, rather than the pixel domain, the algorithm can quickly find semantically corresponding nearest neighbours.



---

# Test design

## 3.1 Test Objectives

As was stated in the introduction, goals of the tests were to find advantages and disadvantages of each method, as well as to compare the quality of the output images regarding both their ability to preserve the semantic structure of input content and their ability to reproduce given style. Since the whole task of qualitative comparison would be too broad and therefore too complex and difficult, the task has been simplified to the reconstruction of human faces only.

This simplification has been made for several reasons: this way, the quality of reconstruction can be judged by anyone since people generally have excellent ability of face perception that is independent of their field of study or other major factors. If we were to compare the quality of, for example, landscapes, people's opinion on the quality would differ based on their field of study or work experience since an artist would probably notice a lot more subtle details than a normal person would. This is one of the reasons why researchers use human faces as example images if their algorithm supports their reconstruction.

## 3.2 Dataset creation

To ensure the maximum objectivity of the test, a sufficient dataset had to be created for the tests. Since we knew from the start of the design phase that most of the questions in the tests/surveys would ask the respondents to choose which of the given images (method outputs) is better than the other, it would be desirable to have an output for the same input content and style for each pair of the tested methods, such as those shown in Figure 3.1. However, even though many authors of the tested methods use the same style images to present the capabilities of their methods in comparison of other algorithms, it was not possible to compile a sufficient dataset just from the data the authors

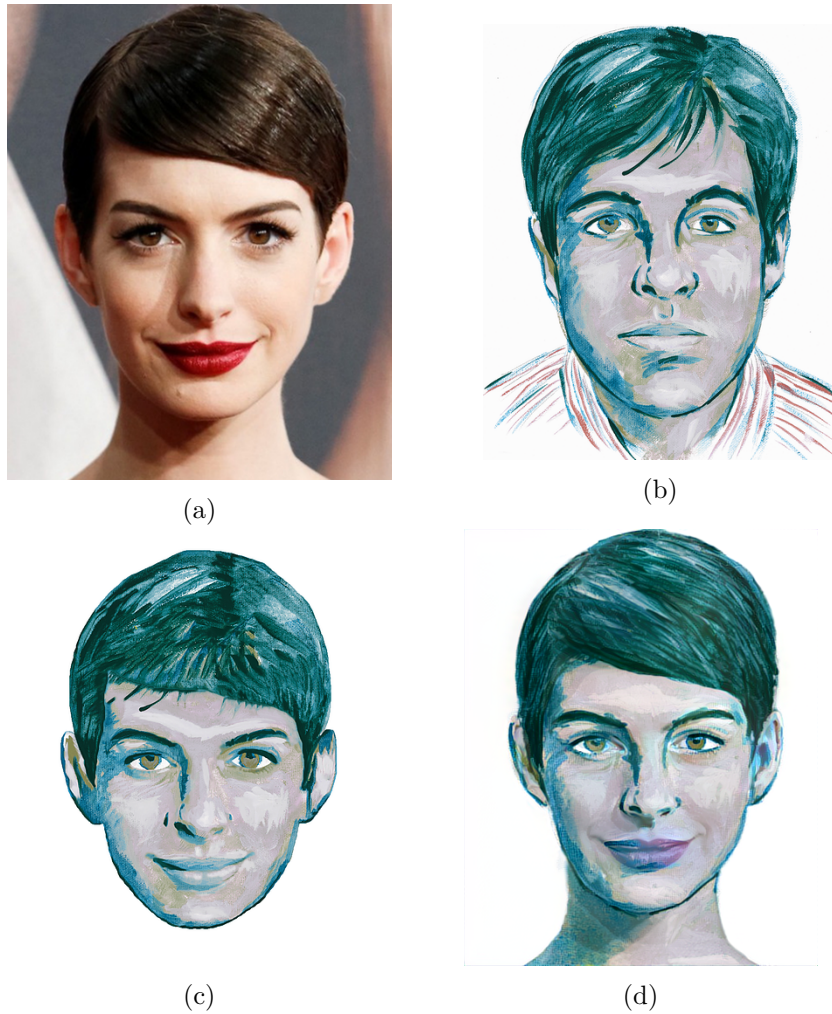


Figure 3.1: The testing dataset was designed to contain outputs from multiple methods for the same input content (a) and input style (b). In this case for this particular input data we have output from the Fišer et al. 2017 [6] on (c) and the output of Selim et al. 2016 [14] on (d)

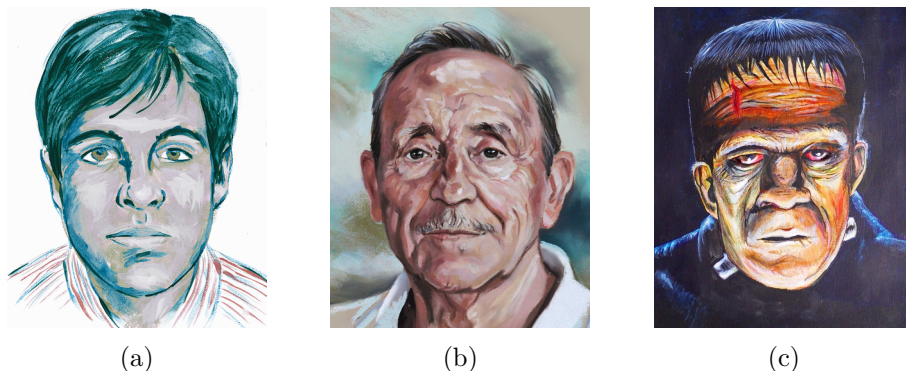


Figure 3.2: Paintings used as input styles for the style transfer. These are (from left to right): ©*Jen Garcia* via flickr, ©*Graciela Bombalova-Bogra*, ©*Scary Zara Mary* via facebook

offer in their papers. For this reason, it was needed to generate our own outputs for each of the tested methods. For the source styles I selected three images that were often used as example styles for these types of algorithms. These source styles are displayed in the Figure 3.2.

As a starting point, I collected the publicly available outputs of *StyLit* [6] and Selim et al. 2016 [14] methods from the research papers. For the Gatys et al. 2015 method, the online application *DeepArt.io* [3] has been used since it uses this exact method. As for the remaining two methods, no sufficient outputs were available and thus needed to be generated.

All of our testing data is available on the attached CD.

### 3.2.1 Deep Image Analogy dataset creation

For the Liao et al. 2017 method an implementation by the original authors *MSRA CVer* (Microsoft Research Asia, Computer Vision) was used, which is publicly available on GitHub under the name *Deep Image Analogy* [20].

This implementation is based off the Caffe framework, which is a deep learning framework allowing programmers to develop complex neural algorithms while using many preprogrammed methods and procedures the framework offers, making the whole development process much easier. This framework is developed by the *Berkerley Vision and Learning Center* (BLVC), but the *MSRA* group uses its own fork of the framework (it has been tested, however, that the *Deep Image Analogy* algorithm runs on both the *MSRA* and *BLVC* versions). This framework uses CUDA for GPU acceleration and can use the Nvidia cuDNN library for further computation speed increase.

The *Deep Image Analogy* implementation takes several arguments, which allow for a certain level of output customization. These are discussed on its

GitHub page, and are as follows:

- **path\_model**, path to the VGG model used
- **path\_A**, path to the input image A
- **path\_B**, path to the input image B
- **path\_output**, the path to the folder where the output images will be saved
- **GPU Number**, ID of the GPU to be used
- **Ratio**, ratio to resize the input images before being processed by the algorithm. This is a crucial argument and is further discussed in the following paragraphs.
- **Blend Weight**, the level of weights in the blending process. Also important and further discussed.
- **Flag of the WLS Filter**, flag for a better quality of photo to photo style transfer

Most of these arguments are not particularly interesting and do not need further explanation, such as input image path and GPU ID. The only detail worth noting here is that there is no content source image and no style source image, this algorithm will produce results for both combinations: A being the content, B being the style and vice versa.

The arguments **ratio**, **blend weight** and the **flag of the WLS filter** are the ones that directly determine how the final product will look like. I will omit the WLS filter flag from this description as we won't be using the photo to photo style transfer.

The **ratio** argument controls the downscaling, it says how much will the input images be shrunk before being processed. A ratio value of 1 means that there is no downscaling, ration value of 0.5 means that the images will be downscaled to 50% in each dimension (and will, therefore, have only 25% of original area). Although higher ratios than 1 are accepted, their output is extremely similar to images with 1.0 ratio and the only major effect is a significant increase in required video memory.

This argument has two major effects: it greatly affects the style transfer quality if the value is in the interval from 0 to 1, and it also greatly affects the required amount of video memory for the process. The ratio's influence in image quality can be observed in the Figure 3.3. The memory requirement is, however, a big problem. If the GPU does not have enough VRAM for the computation, the program will crash during runtime. This means that lowering the ratio allows the program to run even on weaker GPUs, but limits



Figure 3.3: Results of the *Deep Image Analogy* algorithm. Image 3.3a shows the original content image and the image 3.3b shows the original style image. Images (c) through (i) are final outputs of the algorithm for Blend Weights equal to 2 and various ratio values. The ratio values are  $0.4$  for (c),  $0.5$  for (d),  $0.6$  for (e),  $0.7$  for (f),  $0.8$  for (g),  $0.9$  for (h) and  $1.0$  for (i). Even though the authors suggest using ration equal to 0.5 for this case [20], ratio 1.0 provides substantially better result. This may be due to some inconsistencies of input image resolutions between our test and the authors' initial run. The effect of the ratio is clearly visible on these generated images: higher ratio provides better accuracy of facial features transfer from the content image, while lower ratios maintain larger patches of the style image. All the generated images are included on the attached CD in their original resolution.

the best quality a GPU can produce. Sadly, the maximal possible ratio on many middle-end GPUs is around 0.5, and the program sometimes crashes with a ratio equal to 1.0 even on some high-end GPUs (with input images 448x448px). This problem can be alleviated by compiling the Caffe framework without the cuDNN library, which increases the time needed to process the image, but lowers the VRAM requirement slightly allowing lower-end GPUs to generate better outputs. Used ratios and GPUs are further discussed in section 4.3.

For our testing dataset, a ratio equal to 1 and blending weight equal to 2 was used.

### 3.2.2 Fast Neural Style dataset creation

For the Johnson et al. 2016, the implementation by its original authors was used. This implementation is publicly available on GitHub [21], in a repository named *Fast neural style* shared by the author, Justin Johnson.

This specific application is implemented in the Torch framework, which is a Python/C++ computing framework similar to Caffe used in the previous section. This framework, like Caffe, uses CUDA and cuDNN for GPU acceleration. As described in the first chapter, this algorithm uses pre-trained models representing styles, which can be then applied to images rapidly. The authors provide several pre-trained models in the repository, unfortunately, none of these models represent styles used in other methods. This leads to the need to train our own models for the purpose of the tests.

The authors also provide a Python script for the model generations in the repository, along with a detailed description of how to use it. The user has to provide two things for the model generation to work: a model of a convolutional network for image recognition (in Torch format) and a sufficient training dataset packed as an HDF5 file. To keep the results as close as possible to the original, we used the same convolutional network model as the authors (the VGG-16 model) and also the same training dataset, which was the COCO dataset (Common Objects in Context). The training dataset contains a series of images accompanied by validation images, allowing to train neural networks in object recognition etc. The specific dataset used contained 20 000 images and 2 000 validation images in total.

Once the training dataset has been packed as an HDF5 file, the training script could be run to create a t7 file from a source style image. The process, however, requires two additional values: a **content weight** and a **style weight**. These values greatly affect how the final outputs stylized by the trained models will look like, as shown in the Figure 3.4. The values for our models were chosen in a way the results were as similar as possible to the outputs of *Prisma*, the smartphone application using this algorithm. This, however, proved very difficult. Not only that these weights are not publicly available, but the outputs of *Prisma* are more than likely further processed

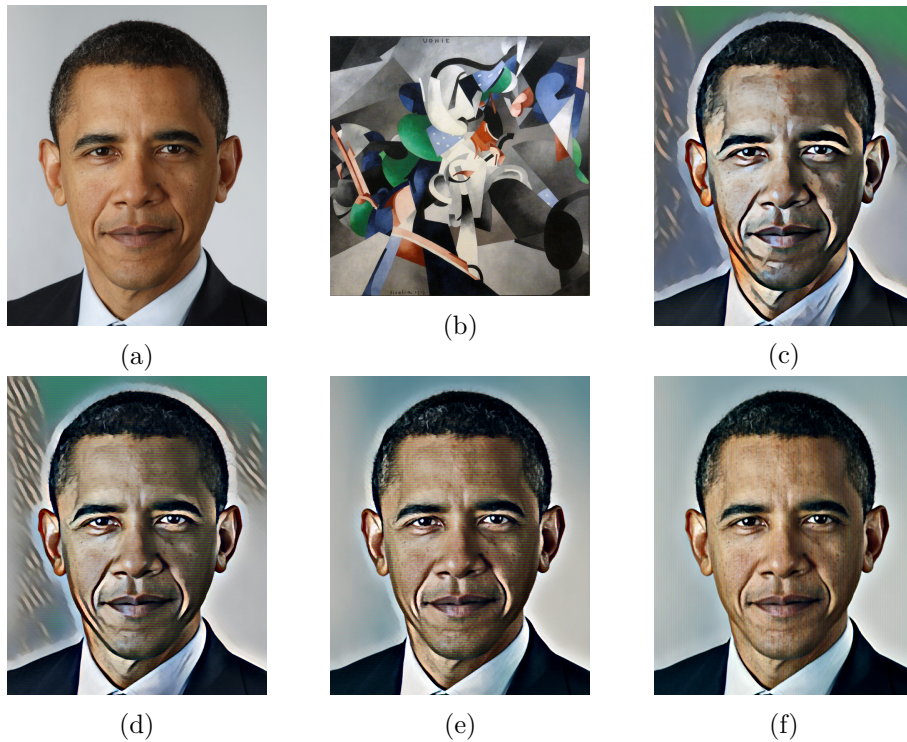


Figure 3.4: The effects of content weight on the final output (with fixed style weight equal to 1). Image (a) is the source content and image (b) the style on which the style model was trained on. Image (c) shows stylized output for content weight equal to 1, (d) equal to 2, (e) equal to 5 and (f) equal to 10. It can be seen that, as expected, the elements of source style are suppressed with higher content weight. It is also clearly visible that there are only minor differences in color between (e) and (f), and above the content weight of 2 the stylized output retains almost none of the structures in the source style.

by an unknown postprocessing method. This lead to the need to create style models with multiple content and style weight options and then choose the best combination, which was the one most similar to a *Prisma* output for the same style.

Style models were generated for style weight equal to 1 and for content weights 1, 2, 5 and 10 (as demonstrated in Figure 3.4 ). We were not able to easily generate models for more than these 4 combinations, as the generation of one style model took more than 8 hours on a Tesla K40 GPU ( more in Section 4.3 ). In the end, each tested style model was most similar to its *Prisma* counterparts with style weight equal to 1 and content weights also equal to 1. Because of that, these values were used for the source styles needed for the testing dataset. All the generated style models are available on the attached CD.

### 3.3 Perceptual experiment

The first step in the testing process was a small scale perceptual experiment. There were several goals of this experiment:

- Obtaining detailed information about the resulting quality. Since the final survey would only collect data about which output is better than other, this experiment was a great chance to try and find more information than that, specifically what features did the respondent consider more important than others (quality of the color transfer, quality of eye structure preservation, etc.)
- Obtaining tentative results of the final data.
- Finding areas of interest in the style transfer process.
- Testing the suitability of the question type. As there was only one chance of getting the survey questions right, I used question similar to those that would be in the final survey and intended to modify them based on the feedback.

To accommodate all above-said goals the experiment was devised as an individual perceptual test, where each respondent answered 2 questions at each output dataset and then explained his decision and described what features of each output were important to him. The two questions asked were:

- Which one of these stylizations better preserves the person's identity?
- Which one of these stylizations better reproduces the given artistic style?

Each respondent was also allowed to say that he's not able to decide which output is better. The format of questions can be seen in the file attached on the CD.

For this experiment the null hypothesis  $H_0$  has been set as:

$$H_0 = \text{There is no significant statistical difference between the quality of outputs of the compared methods}$$

To simplify the experiment and to get firmer results for at least a subset of the examined methods, the tested methods for this experiment were limited only to *FaceStyle* vs. every other method, as opposed to having each possible combination of the examined methods. To have 1 question for each one of the possible combination, a total of 10 questions would be required. However, 1 question for each combination *FaceStyle* vs. any other methods requires only 4 question in total, which means that the numbers of questions can be easily doubled to 8 to get better results and alleviate possible bias while still



maintaining low amount of questions so that respondents would not lose focus during the test. Questions were also added to compare the quality of the *StyLit* method vs. a new method by the same authors *StyleBlit*.

To further alleviate possible bias, the respondents were carefully chosen to represent multiple social groups. A total of 13 respondents was selected, among those:

- 6 were male and 7 female
- 4 were actively engaged in art, 4 occasionally enjoyed art and 5 were amateurs
- 2 had a deeper knowledge of the tested algorithms, 11 had none

The test took place on the university grounds, in the Virtual Reality laboratory called VRLab, where no other people except for the respondent and me were present during the test.

### 3.4 Online survey

A large-scale online survey was devised from the insight gained from the perceptual experiment. The goal of this survey was to get as much data as possible to determine overall quality differences between the style transfer capabilities of examined methods. To achieve this, a simple survey was designed off the questions used in the perceptual experiment. This time, however, each respondent got exactly one question for each combination of the examined methods, did not have to explain his decisions and also was not allowed to say he's not able to decide.

Several requirements were set for the used survey software. The software had to limit responses (or responses per month) as little as possible and had to keep the images used in questions as big as possible to alleviate possible bias caused by downscaling the images. Unfortunately, after trying multiple publicly available software, such as *Survio* and *Google Forms*, none accommodated the requirements sufficiently. This led to the need to create our own survey system from scratch.

#### 3.4.1 Survey system description and implementation

Several requirements were defined for our ideal system:

- Accessible via the internet, located under a trustworthy subdomain
- No limit to the number of responses
- Minimal image downscaling possible
- Simple UI without advertisement



- One question per page, no scrolling required
- Minimize image repositioning/resizing on subsequent pages
- Able to recognize multiple responses from the same person (and filter them during final data processing)
- Safe against cyber attacks

Since the system would be of our own making, no response limit or advertisement was a problem. The department of computer graphics and interaction provided us with space under their subdomain, which allowed us to place our survey system on URL *dotaznik.dcgi.felk.cvut.cz* and achieve the defined requirement with the trustworthy domain. The department also provided a server to run our survey system with *Microsoft Server 2016* OS.

As the survey system did not have to run for a long time (we approximated 4 months necessary), the *XAMPP* development package was used. This package is not optimal for a server that needs to run stably for long periods of time, but the 4 month period required only 1 restart to have the server running the entire time. The fact that the *XAMPP* package contained everything needed for the system (*Apache* server, *MySQL* database etc.) significantly reduced the overhead time cost of the development and allowed us to deploy the system earlier and thus collect data for a longer period of time.

To accommodate all the UI requirements, basic UI has been made. The survey system features a page where the user can choose which survey to complete (Figure 3.5). After selecting the survey, an introduction page is shown with information about the format of the survey and how to complete it (Figure 3.6). The GUI has been optimized to maximize the area of the images on various resolutions and monitor aspect ratios, which is visible in Figure 3.7. The radio buttons were placed in a way that it is obvious which answer is the user choosing. The system also requires the user to answer

### 3.4. Online survey

**Dobrý den,**

v tomto dotazníku je cílem dotázaného (Vás) rozhodnout, která ze stylizací pro stejné vstupní podklady je kvalitnější v dané kategorii. V každé otázce Vám budou předloženy 4 obrázky:

**Vstupní obsah | Vstupní styl : Obrázek A | Obrázek B**

Tedy 2 vstupní podklady a za nimi 2 stylizace. Těchto otázek je 10. V každé jsou dvě podotázky, a to:

**Která z těchto dvou stylizací (obrázků A nebo B) lépe zachovává identitu stylizované osoby?**

**Která z těchto dvou stylizací věrněji reprodukuje zadaný styl?**

Na Vás je rozhodnout u každé otázky a podotázky, která ze stylizací lépe odpovídá zadané kategorii.





**Za každý vstup a vyjádřený názor velice děkuji.**

< Předchozí

Další >

Figure 3.6: Survey system layout - Introduction page

Otázka 10.

Obsah	Styl	Obrázek A	Obrázek B
			
		Obrázek A	Obrázek B

Která z těchto dvou stylizací lépe zachovává identitu stylizované osoby?

Která z těchto dvou stylizací věrněji reprodukuje zadaný styl?

< Předchozí

Další >


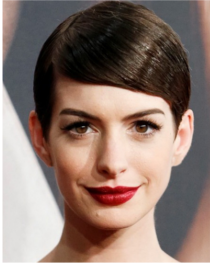





Figure 3.7: Survey system layout - Question page

Otázka 6.

Obsah	Styl	Obrázek A	Obrázek B
			
		Obrázek A	Obrázek B

Která z těchto dvou stylizací lépe zachovává identitu stylizované osoby?

Která z těchto dvou stylizací věrněji reprodukuje zadaný styl?

< Předchozí

Další >

Figure 3.8: Survey system layout - Question answered

### 3. TEST DESIGN

---

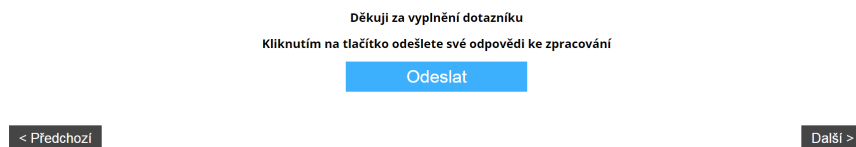


Figure 3.9: Survey system layout - Data submission

both questions before continuing to the next question, but it allows the user to return to any previously answered question and change his/her answers (Figure 3.8). The system then asks the respondent to confirm his/hers answers and to submit them (Figure 3.9).

As the survey system would be fairly simple in its inner structure (choosing A or B in each question), checking whether the submitted answer is one of these 2 options prevented the majority of possible errors and cyber attacks such as SQL Injection, which has been tested and confirmed with an automated penetration test. For purposes of possible future filtration of collected data, the system also saves the IP address of the computer used to submit the survey. This way we were able to get rid of accidental multiple submits (respondent refreshing the page at the wrong time) and possible fake responses.

After the survey system was finished two surveys were made. The first compared all the studied methods and used the data described in the Section 3.2. The second one had the exact same format, but used data comparing outputs of *StyLit* and the outputs of a new algorithm by the same authors, that was still in development at the time.

The entire survey system's source code and databases can be found on attached CD.

---

## Test results

### 4.1 Perceptual experiment

#### 4.1.1 Collected data

The perceptual experiment has been performed as described in Section 3.3. The data were evaluated separately for each pair of tested methods. Within each pair, the data were also evaluated separately for the questions about content and style, and after that, they were evaluated once more as one concatenated set. At each question, if the respondent answered that one output is better than the other, that method got 1 point and the other got 0 points. If the respondent answered that he is not sure, both methods got 0 points. All the collected data are available on the attached CD. Student's t-test was used to analyze the data and the null hypothesis for the test was set as:

$$H_0 = \text{There is no significant statistical difference between the quality of outputs of the compared methods}$$

The results from the test are shown in the Table 4.1. Each row displays one method, and in each cell is a number which says how many percent of respondents preferred the method over the one in the column. While the *FaceStyle* method did not win (score more points) over any other method in the question of identity preservation, it won against every other method in the question of style reproduction.

The results of the t-test are shown in Table 4.2. Each cells shows the probability of correct rejection of the null hypothesis  $H_0$ . In most cases the probability of rejection is over 99%, which means that there is a statistically significant difference in the category.

In the Table 4.3 are the results of the comparison of the outputs of the *StyLit* and *StyleBlit* methods. It is clear that the *StyleBlit* method performs far better in terms of identity preservation, where 100% of the respondents

#### 4. TEST RESULTS

---

Identity preservation					
	<i>FaceStyle</i>	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	-	42.5%	4%	11.5%	4%
Liao	<b>54%</b>	-	X	X	X
Selim	<b>92%</b>	X	-	X	X
Gatys	<b>77%</b>	X	X	-	X
Johnson	<b>92%</b>	X	X	X	-

Style reproduction					
	<i>FaceStyle</i>	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	-	<b>73.5%</b>	<b>65.5%</b>	<b>88.5%</b>	<b>100%</b>
Liao	23.5%	-	X	X	X
Selim	34.5%	X	-	X	X
Gatys	11.5%	X	X	-	X
Johnson	0%	X	X	X	-

Table 4.1: Results of the perceptual experiment. The number in row labeled  $A$  and column  $B$  represents the percentage of respondents that preferred the output of  $A$  over the output of  $B$ . Bold numbers mean that  $A$  got higher score in the comparison with  $B$ . X means that the measurement for that category was not performed.

<i>FaceStyle</i> vs.	Liao	Selim	Gatys	Johnson
Identity preservation	55%	>99%	>99%	>99%
Style reproduction	>99%	89%	>99%	>99%
Overall	84%	96%	43%	33%

Table 4.2: The probabilities of correct rejection of  $H_0$  in each respective comparison.

	Identity preservation		Style reproduction	
	<i>StyLit</i>	<i>StyleBlit</i>	<i>StyLit</i>	<i>StyleBlit</i>
<i>StyLit</i>	-	0%	-	50%
<i>StyleBlit</i>	100%	-	42.5%	-
$H_0$ rejection prob.	>99%		30%	
$H_0$ rejection prob. overall	>99%			

Table 4.3: The results of the *StyLit* vs. *StyleBlit* comparison.

chose its output as the better one. These two methods performed similarly in terms of style reproduction, where the older *StyLit* method got a slightly better score.

The results of the discussion about the quality of the outputs can be found in the following subsections.

#### 4.1.2 FaceStyle vs Gatys et al.

**In the question of identity preservation**, majority of the respondents expressed themselves that the algorithm of Gatys et al. barely changes the input content and only slightly changes the colors and some structures. The majority favored the Gatys et al. output because it preserved all the important facial features almost unchanged: primarily the nose, head shape, and ears. The majority did not favor the eyes preservation at neither the Gatys et al. nor *FaceStyle* and considered their quality to be equal between the 2 outputs.

At the second dataset comparing the output of these two methods, several respondents favored the result of *FaceStyle* over the Gatys et al., stating that the output of Gatys et al. „overexaggerated some facial features“ and made the output look like a caricature.

**In the question of style reproduction**, in the discussion, the majority of the respondents preferred the output of *FaceStyle*, stating that the transfer of colors, structures, and brush strokes was far superior.

Several respondents, mostly the respondents with a background in arts, favored the output of *FaceStyle* for the faithful reproduction of the used medium, such as oil painting.

Several respondents stated that the algorithm *FaceStyle* was adding information into the output image that was not supposed to be there, such as scars on the forehead, which were in the source style but not the source content. This fact was negatively rated by the respondents and classified as an error, even by those with no background in art.

#### 4.1.3 FaceStyle vs Liao et al.

**In the question of identity preservation**, there were multiple arguments for and against both methods in this case. Respondents often stated that the quality of identity preservation is very similar, which is confirmed by the percentage of respondents favoring each method.

Respondents were reproaching the output of *FaceStyle* for the false information transfer, such as the scar mentioned in the previous case, or for the head shape distortion in the dataset with Anne Hathaway, which, as several respondents stated, made her look more like a male rather than a female.

The output of Liao et al. was reproached for facial features distortion, primarily eyes. This output was praised, however, for the hair reconstruction,

since the hairstyle in the output of Liao et al. was very similar to the one in the input content, but the hairstyle in *FaceStyle* was very similar to the one in the input style. This is probably due to the fact that *FaceStyle* transfers such details as a structure and does not distort them to improve the style quality.

**In the question of style reproduction**, respondents were reproaching the output of Liao et al. for worse quality of structure transfer from the source style, the loss of elements of hand-drawn painting, worse transfer of shadows and contours. Some were also pointing out some color transitions in the output (especially in the forehead region in the dataset with Barack Obama), which were better transferred by the *FaceStyle* algorithm.

Output of Liao et al. was praised for the overall quality and its overall consistency with the given input style. For example, even though the output of Liao et al. showed some hints of the scar in the input style, it was not nearly as strong as in the case of *FaceStyle*.

### 4.1.4 FaceStyle vs Selim et al.

**In the question of identity preservation**, the majority of the respondents praised the output of Selim et al. for better preservation of facial features such as the lips, nose, eyelashes, and the head shape. Several respondents also praised the output for being more realistic. One respondent has reproached the output of Selim et al. for the change of the facial expression.

**In the question of style reproduction**, the output of Selim et al. was reproached for the loss of style structures, brush strokes and elements of the used medium.

There were mixed opinions on the color transfer. While the respondents rated both outputs equally in the dataset with Barack Obama, many were praising the output of Selim et al. for better colors in the face and hair (including structures).

The lip color in the dataset with Anne Hathaway caused also very mixed reactions. While some were reproaching the output of Selim et al. for inconsistency with the input style („the person in the input style does not have a lipstick“), other were praising this detail for the consistency with the author’s intention („if he were to paint a person with a lipstick, he would have done it like this“).

### 4.1.5 FaceStyle vs Johnson et al.

**In the question of identity preservation**, similarly to the discussion at the outputs of Gatys et al., many respondents were commenting on the output of Johnson et al. that it’s barely changing the input content and only changing colors, which is why identity is better preserved by Johnson et al.

**In the question of style reproduction**, for the reasons stated in the content preservation, all of the respondents praised the style transfer of



*FaceStyle*. The output of Johnson et al. was reproached for transferring only the bare minimum of the style, colors mostly. Some respondents were commenting on the output that it looks like an entirely different style. Another comment that the respondents were saying often is that the output in the dataset with Anne Hathaway was unnaturally bright.

#### 4.1.6 StyLit vs StyleBlit

**In the question of identity preservation**, every respondent praised the outputs of *StyleBlit* for better reproduction of various details in the facial region, chest region and fingers. The only thing these outputs were reproached for was that sometimes the specular highlight was incorrectly placed in a region that should have been shadowed. Despite several of these negative comments, all of the respondents considered the output of *StyleBlit* better.

**In the question of style reproduction**, most of the respondents stated that both of the reproductions were extremely similar and had to be very picky about the details to choose which of the outputs was better. Many of the respondents commented that it is a shame that the output images do not have any background, which made it slightly more difficult to them to compare the style transfer.

Respondents were praising *StyLit* for the color transfer, correct color transitions and color bleeding from the background. The stroke reproduction was also better at the *StyLit* output.

Most praise of the *StyleBlit* output came from the fact that it preserved the original content better and therefore could better style the details which *StyLit* did not style at all. The respondents also praised the output of *StyleBlit* for better colors near the specular highlight regions.

#### 4.1.7 Summary

The results of identity preservation and style reproduction show clear signs of correlation within each tested pair, which is hardly surprising. The more an image is changed to look like a given style, the less of the original pixel values remain. The comparison with Johnson et al. was an extreme case, since absolute majority of the respondents stated that the output of Johnson et al. contained very little of the given style, which resulted in it having more than 90% of the respondents preferring its content preservation, but 0% of its style reproduction.

The only method in this test using the guided texture synthesis was the *FaceStyle* method [13] and its style reproduction was preferred by more than 50% of the respondents in each case, which clearly shows that the algorithm produces consistent faithful reproductions of the given styles. This method, however, was not preferred in identity preservation, which can be explained by the observed correlation. The only case where the votes for identity

## 4. TEST RESULTS

---

Category	<i>FaceStyle</i>	Liao	Selim	Gatys	Johnson
Content - Points total	62	304	228	339	597
Content - Percentage of max	4%	19.6%	14.8%	22%	38.8%
Content - Average	0.10	0.49	0.37	0.55	0.97
Content - Variance	0.09	0.25	0.23	0.24	0.02
Style - Points total	493	279	423	191	144
Style - Percentage of max	32%	18%	27.6%	12.4%	9.2%
Style - Average	0.80	0.45	0.69	0.31	0.23
Style - Variance	0.15	0.24	0.21	0.21	0.18

Table 4.4: Overall summary of the collected data. The points are taken from all comparisons and evaluated together.

preservation was mostly tied was the comparison with the algorithm of Liao et al.

## 4.2 Online survey

The two surveys comparing the overall quality of the outputs of the compared methods were publicly available since the 2nd of March 2018 and the data was collected on 10th of May 2018, making the survey run for approximately **2 months**. The first survey, comparing all the studies methods, got a total of **153** responses, and the second, comparing outputs of *StyLit* and *StyleBlit* got a total of **87** responses.

### 4.2.1 All methods comparison

To analyze the collected data, the ANOVA method has been used to find differences among the group means and test the null hypothesis, which states that there is no significant statistical difference between the quality of the tested outputs. Each pair has also been tested with a two-tailed paired t-test to find differences between each pair.

The ANOVA test in both the identity preservation and style reproduction categories returned P-value less than  $10^{-130}$ , which means there is a probability higher than 99% for correct rejection of the null hypothesis and that there is a statistically significant difference between the variances of the method’s outputs quality rating. The results of t-test can be seen in the Table 4.6. In all comparisons except one the probability of correct rejection of the null hypothesis was higher than 99%, implying a huge difference between the qualities of the tested outputs. The only case where the probability was less than 99% was the comparison of identity preservation between the outputs of Liao et al. and Gatys et al., where the probability was equal to 96%, which still implies a statistically significant

Identity preservation					
	<i>FaceStyle</i>	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	-	25%	3%	10%	2%
Liao	<b>75%</b>	-	<b>81%</b>	41%	1%
Selim	<b>97%</b>	19%	-	31%	3%
Gatys	<b>90%</b>	<b>59%</b>	<b>69%</b>	-	4%
Johnson	<b>98%</b>	<b>69%</b>	<b>97%</b>	<b>96%</b>	-
Style reproduction					
	<i>FaceStyle</i>	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	-	<b>73%</b>	<b>67%</b>	<b>86%</b>	<b>97%</b>
Liao	27%	-	26%	33%	<b>95%</b>
Selim	33%	<b>74%</b>	-	<b>85%</b>	<b>84%</b>
Gatys	14%	<b>67%</b>	15%	-	29%
Johnson	3%	5%	16%	<b>71%</b>	-

Table 4.5: All methods survey – comparison results. The number in row labeled  $A$  and column labeled  $B$  means that that percentage of respondents preferred the output of the method  $A$  over the output of the method  $B$ . Bold number mean that  $A$  got higher score in the comparison with  $B$  than  $B$  did.

Identity preservation				
	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	>99%	>99%	>99%	>99%
Liao	-	>99%	96%	>99%
Selim	>99%	-	>99%	>99%
Gatys	96%	>99%	-	>99%
Style reproduction				
	Liao	Selim	Gatys	Johnson
<i>FaceStyle</i>	>99%	>99%	>99%	>99%
Liao	-	>99%	>99%	>99%
Selim	>99%	-	>99%	>99%
Gatys	>99%	>99%	-	>99%

Table 4.6: All methods survey - probability of correct  $H_0$  rejection.

#### 4. TEST RESULTS

---

Category	<i>StyLit</i>	<i>StyleBlit</i>
Content - Points total	213	<b>396</b>
Content - Percentage of max points	35%	<b>65%</b>
Content - $H_0$ rejection prob.	>99%	
Style - Points total	198	<b>411</b>
Style - Percentage of max points	33%	<b>67%</b>
Style - $H_0$ rejection prob.	>99%	

Table 4.7: *StyLit StyleBlit* survey - summary of the collected data. Bold numbers show the better score.

difference. All the collected data and computed values are available on the attached CD.

In Table 4.4 are the statistic about the collected data. The maximal amount of points a method could score in each category was 612. In terms of content preservation, the method by Johnson et al. was consistently rated the best and the method *FaceStyle* the worst, which is consistent with the results of the previous perceptual experiment. In style reproduction, the best rated method was the *FaceStyle* method and the worst was the one by Johnson et al. Not only that this is also consistent with the previous perceptual experiment, but it also further confirms the observed correlation between these two categories. Table 4.5 shows how many percent of respondents preferred the output of the method in the row against the output of the method in the column. The correlation can be observed once again, as many methods that placed better in identity preservation were considered worse in style reconstruction in the same comparison. There are a couple of observed exceptions from this correlation, which is the comparison of the outputs of Gatys et al. and Liao et al., where the results of Gatys et al. were considered better in both categories, even if only slightly, and the comparison of outputs of Gatys et al. and Johnson et al. , where the outputs of Johnson et al. were considered much better in both categories.

#### 4.2.2 StyLit vs. StyleBlit

Table 4.7 shows the results of the *StyLit* and *StyleBlit* survey comparison. *StyleBlit* got better score in both content preservation and style reproduction categories. These results are consistent with the results obtained from the perceptual experiment, where the *StyleBlit* method got a much higher score in identity preservation and was roughly even in style reproduction. In both categories t-test was used and the null hypothesis  $H_0$  can be rejected with probability higher than 99%, which means that there is a statistically significant difference between the quality of these two methods and that the results of the *StyleBlit* are generally better.

	Execution time [s]									
Ratio	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
PC1	8.45	17.25	26.00	37.91	53.19	75.02	95.78	122.79	156.36	184.68
PC2	7.02	16.95	34.11	56.03	83.77	X	X	X	X	X

Table 4.8: Execution time of the Deep Image Analogy implementation for different values of the **Ratio** parameter. X in the place of the execution time means that the program crashed due to insufficient video memory. *WDDM Timeout Detection & Recovery* had to have been disabled on PC2 since tasks above Ratio of 0.1 took more than the default 2 seconds and OS was killing the process before its successful completion.

## 4.3 Performance measurement

### 4.3.1 Liao et al.

For the Deep Image Analogy implementation of the method by Liao et al., the following 2 PC configurations were used to measure the performance:

- PC1
  - Windows Server 2012 R2
  - Intel Xeon 32-core 2.4GHz
  - 64GB RAM
  - NVIDIA GeForce GTX TITAN Black (6GB)
- PC2
  - Windows 7
  - Intel i5 4-core 3.40GHz
  - 16GB RAM
  - NVIDIA GeForce GTX 770 (2GB)

To explore what parameters have a heavy impact on the implementation’s performance, multiple execution times were measured with different input values. The input content and style images remained the same for each execution and had a resolution of 448x448px. The values changed were **Ratio**, which I expected to have the biggest impact on the performance, and **Blend Weight**, which I expect to have a very little impact.

The first tested parameter was the **Ratio** parameter, which controls the scale of the input image, as described in Chapter 3. The runtimes were measured on both used PCs, and the time needed is shown in Table 4.8. These times are also shown in the graph in Figure 4.1. The runtimes are

#### 4. TEST RESULTS

---

Blending Weight	Execution time [s]			
	1	2	3	4
PC1	72.66	75.01	76.13	73.87

Table 4.9: Execution time of the Deep Image Analogy implementation for different values of the **Blending Weight** parameter. The Ratio parameter is fixed to 0.6

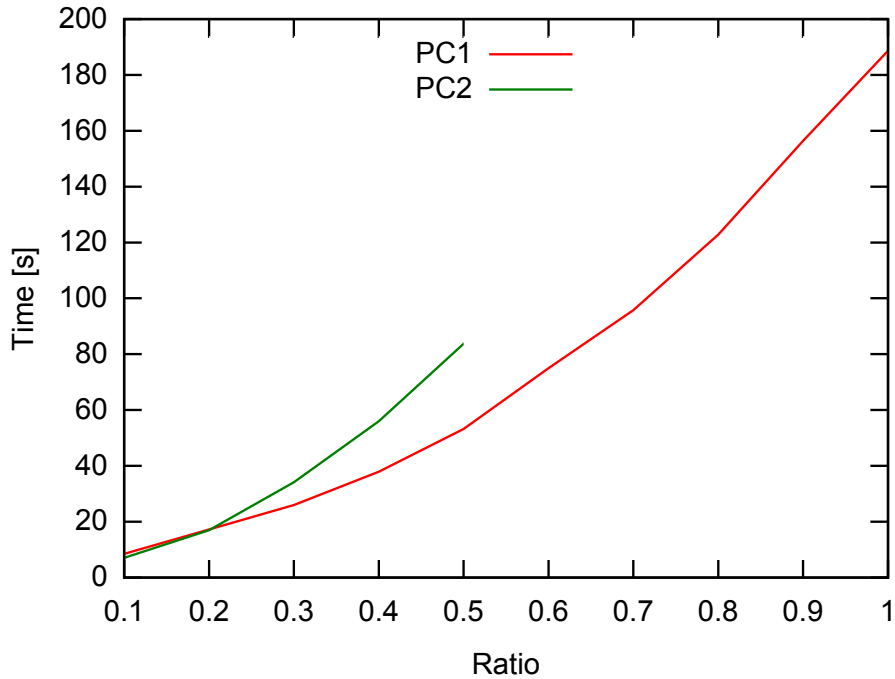


Figure 4.1: Deep Image Analogy execution times. Only the executions that ended successfully are shown in the graph.

fairly short for smaller Ratio values, but grow up to a couple of minutes for larger ratios. The growth also seems to have quadratic nature, which is not surprising given the fact that ratio is applied twice to the image, once for each side, and therefore the image has  $\text{ratio}^2$  pixels of its original size. The growth then seems to be linear to the number of pixels.

The **Blending Weight** parameter was tested on PC1 for a fixed Ratio of 0.6. The resulting execution times were only slightly different, and as expected, they do not seem to show signs of change based on the parameter. The execution times are shown in Table 4.9.

### 4.3.2 Fast Neural Style

As described in the previous chapter, the implementation by Johnson et al. „Fast neural style“ [21] has been used to generate outputs for this method. This method uses pre-trained style models in the Torch dataset format (.t7) to quickly apply the style to presented content images.

A dedicated virtual server has been used with the following configuration:

- Ubuntu OS
- 10-core CPU
- 32GB RAM
- Tesla K40 GPU

All the performed tasks were GPU accelerated using both CUDA and cuDNN if the acceleration was supported. First, a file saved in the Hierarchical Data Format (.h5) was created for the style training process. Afterward, three style models were trained, each with the same settings: A virtual server has been used with this configuration:

- Both content weight and style weight equal to 1
- 40 000 iterations (recommended by creators)
- 384 style image site (recommended by creators)

A total of three style models were trained, one for each of the chosen styles shown in Figure 3.2. The time needed to generate a style model for each was:

- 8 hours, 1 minute and 48 seconds
- 8 hours, 1 minute and 35 seconds
- 8 hours, 1 minute and 37 seconds

As can be seen from the measured times, they are all very consistent and have an average of **8 hours, 1 minute and 40 seconds**. There were no other processes running parallel to these computations (other than the bare necessities of the no-GUI OS) and the computations were slowed down only by the debug text output. After the model training was completed, applying the style to selected content images took less than 10 seconds each.

## 4.4 Summary

After all the performed experiments, the following can be said about each studies method:

- ***FaceStyle***. This method was consistently rated the best in style transfer in all of the experiments, but was not rated higher in identity preservation than any other method, getting close only to the method of Liao et al. We can, therefore, infer that this method produces visually pleasing results, truthful to the original artistic styles, and better in quality than those of the other methods in this study. This method also produces high-quality results when applied to video sequences and offers the user the option to influence the amount of temporal flickering, ranging from a very noisy result to a 3D texture transfer look.
- ***Deep Image Analogy*** (Liao et al.). This method provides consistently good results on a wide range of possible inputs, being rated high in identity preservation and below average in style reproduction. The versatility of this method is a big advantage, but the high computational overhead and very high HW requirements still pose a big issue. Nevertheless, this method still offers a lot for future research.
- **Selim et al.** This method produced consistently very good outputs in terms of style reproduction, being rated lower only in comparison with the *FaceStyle* method. However, in the identity preservation category, this method was generally rated very low, which once again coincides with the observed correlation and should not be taken as a major disadvantage. It can be concluded that this method provides very high-quality results, better than most of those provided by other methods in this study, with the exception of *FaceStyle*.
- **Gatys et al.** This method was rated very high in terms of identity preservation, being rated worse only in comparison with the outputs of Johnson et al., but did not score many points in style reproduction, in most comparisons it scored less than 30% of possible points. The only exception was the comparison with the method of Liao et al., where this method was rated better in both categories. According to the results of the first experiment, this method suffers from wash-out effects and loses a lot of important details of the used artistic medium, such as individual strokes.
- **Johnson et al.** This method was rated the best in identity preservation, in most comparison even scored over 90% of points, but was rated very low in style reproduction, being placed higher only than



the method of Gatys et al. According to the first experiment, this method does very little with the content image in terms of style transfer, and according to the respondents it only performs a sort of color transfer with no structures whatsoever. Creation of pre-trained style models is expensive and takes a long time even on high-end GPUs, but the speed of style transfer with a completed style model is a major advantage of this method, even if the quality suffers from it.

In the comparison of the *StyLit* and *StyleBlit* methods, the *StyleBlit* results were rated better in both categories, making the method mostly superior to its predecessor. Along with the fact that the *StyleBlit* works much faster even on low-end devices, we can safely infer that the method is a step in the right direction.



---

## Conclusion

In this thesis, I've described the current leading methods of example based style-transfer from the fields of both guided texture synthesis and neural network based synthesis. I've also designed and performed two experiments comparing the overall quality of their outputs: a perceptual experiment, describing the points of interest in the style transfer process and comparing the output quality of the *FaceStyle* method against outputs of other methods; and an online survey, collecting quantitative data comparing the overall style transfer quality of each of the method's outputs against the outputs of the rest.

All the performed tests show that there are huge differences between the output quality of the state-of-the-art methods. Unsurprisingly, all the test show strong signs of correlation between how people perceive the quality of identity preservation and style reproduction, where better rating in one category generally leads to a lower one in the other. *FaceStyle*, the only method in the tests from the group of guided texture synthesis, was consistently rated best in style reproduction, showing that this approach leads to visually more pleasing results.

The significant computational cost of the state-of-the-art methods precluded them from being used in real-time applications or required various quality downgrades to make the real-time usage possible in at least some way. The progress introduced by the new *StyleBlit* method, however, opens the methods to real-time usage even on low-end devices, and the experiments done in this thesis show that the changes do not influence the output quality in a negative way, but rather make the output slightly more visually pleasing.



---

## Bibliography

- [1] Semmo, A.; Isenberg, T.; et al. Neural Style Transfer: A Paradigm Shift for Image-based Artistic Rendering? In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR as part of Expressive, July 29–30, Los Angeles, CA, USA)*, edited by H. Winnemöller; L. Bartram, New York: ACM, 2017, pp. 5:1–5:13, doi:10.1145/3092919.3092920. Available from: <https://tobias.isenberg.cc/VideosAndDemos/Semmo2017NST>
- [2] Prisma Labs Inc. Prisma - AI Powered Art Styles. 2016. Available from: <https://prisma-ai.com/>
- [3] Bethge, M.; Ecker, A.; et al. Deepart.io - Become a digital artist. 2016. Available from: <https://deepart.io/>
- [4] Liao, J.; Yao, Y.; et al. Visual Attribute Transfer through Deep Image Analogy. *CoRR*, volume abs/1705.01088, 2017, 1705.01088. Available from: <http://arxiv.org/abs/1705.01088>
- [5] Hertzmann, A.; Jacobs, C. E.; et al. Image Analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, New York, NY, USA: ACM, 2001, ISBN 1-58113-374-X, pp. 327–340, doi:10.1145/383259.383295. Available from: <http://doi.acm.org/10.1145/383259.383295>
- [6] Fišer, J.; Jamriška, O.; et al. Example-Based Synthesis of Stylized Facial Animations. *ACM Transactions on Graphics*, volume 36, no. 4, 2017.
- [7] Sloan, P.-P. J.; Martin, W.; et al. The Lit Sphere: A Model for Capturing NPR Shading from Art. In *Proceedings of Graphics Interface 2001, GI '01*, Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2001, ISBN 0-9688808-0-0, pp. 143–150. Available from: <http://dl.acm.org/citation.cfm?id=780986.781004>

- [8] Bénard, P.; Cole, F.; et al. Stylizing Animation by Example. *ACM Trans. Graph.*, volume 32, no. 4, July 2013: pp. 119:1–119:12, ISSN 0730-0301, doi:10.1145/2461912.2461929. Available from: <http://doi.acm.org/10.1145/2461912.2461929>
- [9] Barnes, C.; Shechtman, E.; et al. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, volume 28, no. 3, Aug. 2009.
- [10] Newson, A.; Almansa, A.; et al. Video Inpainting of Complex Scenes. *CoRR*, volume abs/1503.05528, 2015, 1503.05528. Available from: <http://arxiv.org/abs/1503.05528>
- [11] Kaspar, A.; Neubert, B.; et al. Self Tuning Texture Optimization. *Computer Graphics Forum*, 2015, doi:10.1111/cgf.12565.
- [12] Jamriška, O.; Fišer, J.; et al. LazyFluids: Appearance Transfer for Fluid Animations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2015)*, volume 34, no. 4, 2015: p. 92.
- [13] Fišer, J.; Jamriška, O.; et al. StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. *ACM Transactions on Graphics*, volume 35, no. 4, 2016.
- [14] Selim, A.; Elgharib, M.; et al. Painting Style Transfer for Head Portraits Using Convolutional Neural Networks. *ACM Trans. Graph.*, volume 35, no. 4, July 2016: pp. 129:1–129:18, ISSN 0730-0301, doi:10.1145/2897824.2925968. Available from: <http://doi.acm.org/10.1145/2897824.2925968>
- [15] Gatys, L. A.; Ecker, A. S.; et al. A Neural Algorithm of Artistic Style. *CoRR*, volume abs/1508.06576, 2015, 1508.06576. Available from: <http://arxiv.org/abs/1508.06576>
- [16] Johnson, J.; Alahi, A.; et al. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.
- [17] Fišer, J.; Jamriška, O.; et al. StyleBlit: Fast Example-Based Stylization with Local Guidance. *ACM Transactions on Graphics*, volume 37, no. 4, 2018.
- [18] Stanford Vision Lab. Large Scale Visual Recognition Challenge 2014 Results. 2014. Available from: <http://image-net.org/challenges/LSVRC/2014/results>

- [19] Portilla, J.; Simoncelli, E. P. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, volume 40, no. 1, Oct 2000: pp. 49–70, ISSN 1573-1405, doi:10.1023/A:1026553619983. Available from: <https://doi.org/10.1023/A:1026553619983>
- [20] MSRACVer. Deep Image Analogy. 2017. Available from: <https://github.com/msracver/Deep-Image-Analogy>
- [21] Johnson, J. Fast Neural Style. 2017. Available from: <https://github.com/jcjohnson/fast-neural-style>





## Acronyms

**GUI** Graphical User Interface

**LPE** Light Path Expression

**MSRACVer** Microsoft Research Asia, Computer Vision

**UI** User Interface

**VGG** Visual Geometry Group



---

## Contents of enclosed CD

	readme.txt .....	the file with CD contents description
	attachments .....	the directory with the data
	generated_images .....	all the images generated and used for experiments
	src .....	the directory of source codes
	survey_system .....	survey system source code
	thesis .....	the directory of $\text{\LaTeX}$ source codes of the thesis
	text .....	the thesis text directory
	thesis.pdf .....	the thesis text in PDF format