Bootstrap Optical Flow Confidence and Uncertainty Measure

Jan Kybic[a,*], Claudia Nieuwenhuis[b,1]

[a]*Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering,*
Czech Technical University in Prague, Czech Republic
[b]*Heidelberg Collaboratory for Image Processing, University of Heidelberg*
Heidelberg, Germany

Abstract

We address the problem of estimating the uncertainty of optical flow algorithm results. Our method estimates the error magnitude at all points in the image. It can be used as a confidence measure. It is based on bootstrap resampling, which is a computational statistical inference technique based on repeating the optical flow calculation several times for different randomly chosen subsets of pixel contributions. As few as ten repetitions are enough to obtain useful estimates of geometrical and angular errors. For demonstration, we use the combined local-global optical flow method (CLG) which generalizes both Lucas-Kanade and Horn-Schunck type methods. However, the bootstrap method is very general and can be applied to almost any optical flow algorithm that can be formulated as a pixel-based minimization problem. We show experimentally on synthetic as well as real video sequences with known ground truth that the bootstrap method performs better than all other confidence measures tested.

Keywords: optical flow, bootstrap, confidence measure, motion estimation, uncertainty estimation

1. Introduction

Recovering optical flow (OF) from an image sequence is one of the fundamental algorithms in computer vision [8, 15, 16], a crucial step in motion analysis which is important in a variety of application domains including scene interpretation, video compression and medical imaging. The problem is difficult, ill-posed, and inherently ambiguous because of appearance and illumination changes, imaging system imperfections, noise, lack of texture, and the aperture effect. Consequently, optical flow can only be recovered approximately and the error is spatially varying. However, standard OF algorithms do not provide any estimate of this error.

The aim of this work is to provide an algorithm estimating the uncertainty of the calculated OF. It does not require any a priori knowledge or any other input besides the images being registered. The method is applicable to sequences, although for simplicity we consider here only the two-image case.

Let us emphasize that we are *not* providing a better OF algorithm in the sense of providing a better flow estimate. Instead, we present a technique for estimating the flow accuracy for a given OF algorithm. Our accuracy estimation method is very general and can be applied to almost any OF algorithm that can be formulated as a minimization problem. The method is novel, based on statistical bootstrap resampling.

We have evaluated our method in the context of confidence measures [2, 4] as this is where most of the prior art is. We show that our method provides the best estimation of the OF accuracy from among a number of other accuracy estimation methods and confidence measures that we have tested (Section 5).

1.1. Problem definition

Given two images $g(x,y,t)$ with $t \in \{0,1\}$, an OF algorithm calculates a flow field $[u\,v](x,y)$, such that

$$g(x+u(x,y),y+v(x,y),t+1) \approx g(x,y,t) \qquad (1)$$

with $t = 0$. At each pixel location i with given ground truth flow, we evaluate two error measures. First, the geometric error (also known as the warping index [39] or endpoint error [3])

$$\varepsilon_i = \big\|[u_i\,v_i] - [u_i^*\,v_i^*]\big\| = \sqrt{(u_i - u_i^*)^2 + (v_i - v_i^*)^2}, \quad (2)$$

where $[u_i\,v_i]$ is the estimated flow and $[u_i^*\,v_i^*]$ the true motion field. Second, we calculate the angular error [3, 4]

$$\phi_i = \text{angle}\big([u_i\,v_i],[u_i^*\,v_i^*]\big) \qquad (3)$$
$$\text{with} \qquad \text{angle}\big([u_1\,v_1],[u_2\,v_2]\big) =$$
$$\frac{180}{\pi} \arccos \frac{u_1 u_2 + v_1 v_2 + 1}{\sqrt{(u_1^2 + v_1^2 + 1)(u_2^2 + v_2^2 + 1)}} \text{ [deg]}.$$

Our aim is to estimate ε_i and ϕ_i for all pixels i, solely from the knowledge of the input g and the OF algorithm. A less ambitious objective is to find an uncertainty measure ψ_i such that the relative ordering of ψ_i is as similar as possible to the ordering of ε_i or ϕ_i. Equivalently, $-\psi_i$ is a confidence measure as normally defined in the literature; higher value corresponds to high confidence and low expected error.

---

*Corresponding author
[1]Presently at Faculty of Informatics, Technical University of Munich, Germany

1.2. Proposed method

Our uncertainty measure is based on bootstrap resampling [12]. The basic idea is as follows: If we had multiple realizations of the given OF estimation problem, such as multiple recordings of the same sequence, we could solve all instances and compare the solutions to estimate the variability of the results. As we are only given one instance of the input data, we will use bootstrap resampling to create a number of similar but slightly perturbed OF estimation problems and proceed as before (see Section 3 for details). Our bootstrap resampling works at a pixel level, the generated problems use different randomly chosen subsets of pixel contributions that are assumed to have the same statistics as the complete set. The interesting and novel aspect is that instead of sampling from a set of values and evaluating a function of these values as in standard bootstrap, we sample from a set of functions and evaluate a functional of these functions. This is a new paradigm which has not yet been theoretically analyzed and it might even be too complicated for a complete analysis ever to be performed, so we can only show its usefulness experimentally.

We have chosen to demonstrate our technique on a combined local-global OF method (CLG) [8] which generalizes both Lucas-Kanade and Horn-Schunck type methods, the archetypes of OF estimation. Following the very same procedure, our estimation can be applied to any OF algorithm based on minimizing a variational image similarity criterion which is spatially decomposable, e.g. to pixel contributions. Most OF estimation approaches can be cast into this framework. For example, the two OF algorithms with the best results on the Middlebury database[2] [3] both minimize a criterion of this type [41, 44]. On the other hand, our method would need to be modified for the third best algorithm [28] which represents the image as a tree of over-segmented regions.

This article extends our earlier work using bootstrap for image registration accuracy estimation in the relatively simple case of block matching [25], i.e. with only two degrees of freedom and no regularization. In the OF case described here, the problem is much harder because the number of degrees of freedom is many orders of magnitude higher (there is one motion vector calculated for each pixel), the unknowns are strongly correlated, and regularization is essential, while we cannot afford more than $10 \sim 100$ bootstrap iterations.

1.3. Related work

There have been many attempts to derive useful confidence measures for OF methods, the main application being to identify unreliable flow vectors for error statistics reporting [4] and weighting or pruning for subsequent processing steps [21, 22].

One class of the confidence measures is based on local analysis of the input images. The simplest one is the image gradient [4], justified by the assumptions that we expect a higher accuracy in textured than in flat regions. Haussecker et al. [14]

and Bigün [5] use the structure tensor [13] and propose several confidence measures derived from its eigenvalues. Uras et al. [40] propose to use the condition number of the spatial Hessian H of the image; a related measure which seems to be slightly more reliable is its determinant $\det(H)$ [4]. Anandan et al. [1] evaluate the dependence of the sum of square differences (SSD) criterion on the displacement and define a confidence criterion as a function of principal curvatures of the SSD surface and the SSD value at minimum. In a unifying way, the 'surface measures' of Kondermann et al. [20] use principal curvatures to analyze the intrinsic dimensions [43] of image invariance functions based on brightness, SSD, gradient, and Hessian.

The second type of confidence measures looks at intermediate results or parameters of a particular OF estimation method. For Lucas-Kanade type local methods [30], we can analyze the conditioning of the system that we need to solve for each pixel. We can use the trace [37], the determinant [4], minimum eigenvalue [34], sum of eigenvalues, or the minimum eigenvalue weighted by the residual [17]. Sensitivity analysis can give us an esimate of the flow error from the residuals [10], this is similar to the FRAE method described in Section 4.1. Bruhn et al. [6] use the local contribution to the total energy being minimized to identify locations where model assumptions are not valid and assignes low confidence to them. Singh's method [38] calculates the local displacement as a mean of a probability distribution derived from the SSD criterion and uses an eigenvalue of its covariance matrix as a confidence measure.

The final group of confidence measures is based on the statistics of the calculated flow and their comparison with a learned model, assigning low confidence to deviations from the model. The linear subspace of the 'correct' flow can be learnt by PCA [21]. Alternatively, the dependence of the flow vectors in a patch on the central vector can be modeled as a multidimensional Gaussian [22].

The advantage of the first and third groups of confidence measure is that they are completely independent of the OF estimation algorithm. Besides an informal evaluation by Barron et al. [4] and a brief one by Bainbridge et al. [2], comparison of confidence measure results can be found for example in Kodermann et al. [22]. Several of the mentioned confidence measures are defined in Section 4.

Bootstrap resampling was used in image processing for example to evaluate the performance of detection algorithms [9, 18] and it was also used to assess the accuracy of a rigid motion estimation landmark-based algorithm [31, 32]. However, as far as we know, bootstrap resampling has never been used in the context of OF estimation and its only other application to area-based image registration is our own work [25] discussed above.

The basic idea of our bootstrap resampling is to run the same algorithm many times on different variants of the input data to analyze the variability. This is different from approaches such as the FusionFlow [29] which apply different algorithms to the same input images and combine the results to obtain a better estimate of the flow.

---

[2] As of April 2010, using average endpoint error and not taking into account unpublished work.

2. Combined local-global method revisited

We briefly present here the combined local-global optical flow method (CLG) by Bruhn et al. [8] which we use to demonstrate our bootstrap estimation technique. This method generalizes both local (Lucas-Kanade) and global (Horn-Schunck) methods. Note that for simplicity we are using neither the nonlinear, nor the spatiotemporal extension of the CLG method. In order to apply the bootstrap estimation, the CLG method needs to be reformulated using minimization completely in the discrete domain, while the original description [8] is based on discretized Euler-Langrange equations for a criterion formulated in the continuous domain. However, we show that the two formulations are equivalent.

The continuous criterion to minimize ([8], Eq. 7) is

$$E_{\text{CLG}}(\mathbf{w}) = \int_{\Omega^c} \mathbf{w}^T J_\rho(\nabla_3 f) \mathbf{w} + \alpha \|\nabla \mathbf{w}\|^2 \, dxdy \quad (4)$$

$$\text{with} \quad \mathbf{w} = [u \, v \, 1]^T,$$
$$\|\nabla \mathbf{w}\|^2 = \|\nabla u\|^2 + \|\nabla v\|^2,$$
$$\nabla_3 f = (f_x, f_y, f_t)^T,$$
$$J_\rho(\nabla_3 f) = K_\rho * (\nabla_3 f \nabla_3 f^T),$$
$$f = K_\sigma * g,$$

where $g(x,y,t)$ is the raw input sequence which is smoothed with a spatial Gaussian filter K_σ with standard deviation σ, $f(x,y,t)$ is the smoothed input image sequence, f_x, f_y, and f_t denote partial derivatives, Ω^c is the continuous image domain, u, v are the x and y components of the motion field to be found, and K_ρ is another Gaussian filter with standard deviation ρ. Using P0 (piecewise-constant) interpolation and replacing $\nabla \mathbf{w}$ by finite differences, Eq. (4) can be discretized as

$$E_{\text{CLG}} \approx E = \sum_{i \in \Omega} E_i \quad (5)$$

$$\text{with} \quad E_i = e_D(i) + \alpha e_S(i)$$
$$\text{and} \quad e_D(i) = h^2 [u_i \, v_i \, 1] J_i [u_i \, v_i \, 1]^T,$$
$$e_S(i) = \sum_{j \in N(i)} (u_j - u_i)^2 + (v_j - v_i)^2,$$

where Ω is the discrete set of image pixels, h is the pixel spacing, i denotes a pixel with coordinates (x_i, y_i), and $N(i)$ is a set of the four neighbors of i. A discretized version of u is $u_i = u(x_i, y_i)$ and similarly for $v \to v_i$ and $J_\rho \to J_i$. The spatial partial derivatives needed to calculate J_ρ are evaluated using a seven-point kernel[3]; temporal derivatives are calculated using first order finite differences, $f_t(x,y) = f(x,y,1) - f(x,y,0)$, as we register only one pair of images. The discretization is consistent, $E_{\text{CLG}} = \lim_{h \to 0} E$.

We find the minimum of the discretized energy

$$(\hat{u}, \hat{v}) = \arg\min_{u,v} E \quad (6)$$

with respect to all u_i and v_i by setting the partial derivatives $\partial E / \partial u_i$, $\partial E / \partial v_i$ to zero. This leads to the following linear system of equations

$$0 = \frac{1}{h^2} \sum_{j \in N(i)} (u_j - u_i) - \frac{1}{\alpha} (J_i^{11} u_i + J_i^{12} v_i + J_i^{13}), \quad (7)$$

$$0 = \frac{1}{h^2} \sum_{j \in N(i)} (v_j - v_i) - \frac{1}{\alpha} (J_i^{12} u_i + J_i^{22} v_i + J_i^{23}) \quad (8)$$

for all $i \in \Omega$ and where the upper indices denote elements in J, e.g. $J_i^{12} = (K_\rho * f_x f_y)(x_i, y_i)$. Note that we have multiplied the equations by h^{-2} in order to get an identical set of equations as (32–33) in [8]. The equations can be efficiently solved by successive over-relaxation (SOR, see Section 3.3)

3. Bootstrap resampling

Bootstrap resampling [11, 12, 18, 45–47] is a computationally based statistical inference technique. The idea is to create B derived datasets by sampling with replacement from the original dataset, apply the algorithm under test to each derived dataset and analyze the B results using the desired statistics.

More formally, in bootstrap resampling we take N i.i.d. samples $\mathsf{X} = \{x_1, \ldots, x_N\}$ of a random variable X. Let $\theta = \varphi(\mathsf{X})$ be some function of interest, e.g. the sample mean of X which approximates the true mean of the random variable X. The task is to find some statistics of θ, denoted $\Phi(\theta)$, e.g. its variance. The bootstrap approach is to create B multisets[4] $\mathsf{X}^{(b)}$, $b = 1 \ldots B$, each containing N elements from X chosen randomly with replacement. We calculate $\theta^{(b)} = \varphi(\mathsf{X}^{(b)})$ for all b and estimate $\Phi(\theta)$ from the set $\{\theta^{(b)}\}$. We refer the reader to a specialized literature for technical conditions of bootstrap convergence [12, 23]. We remark here only that as long as $\varphi(\mathsf{X})$ is a "reasonable" estimator of some statistics of X which depends continuously on the probability density of X, the bootstrap can be expected to work. However, it turns out that so far only relatively simple cases have been analysed theoretically, such as bootstrap estimators of the mean, variance, and confidence intervals. Our case is much more involved as for us the samples x_i are functions (see next section) and our operation of interest φ is a functional. As far as we know, this case has not yet been studied.

3.1. Application to optical flow

In our earlier work [25, 27] we have shown that bootstrap resampling can be applied to estimate the image registration accuracy in the case of block matching. Here we show how to extend it to the OF case. The main idea is to use the individual pixel contributions $e_D(i)$ of the data part of the criterion E given by Eq. (5) as the data set $\mathsf{X} = \{e_D(i); i \in \Omega\}$. The function φ is defined by the minimization $\mathsf{X} \to (\hat{u}, \hat{v})$ described by Eq. (6) and Φ is the desired uncertainty measure, such as the geometrical error ε defined in Eq. (2). In practical terms, we create B

---

[3]The kernel is $[1\ 9\ 45\ \underline{0}\ 45\ 9\ 1]/60$, as used by Bruhn et al. [8].

[4]A multiset is a generalization of a set, which can contain each element several times.

Algorithm 1: Bootstrap resampling for single level optical flow uncertainty estimation

---

Input: Images $g(x,y,t)$, $t = \{0,1\}$
Output: Flow field $[\hat{u}\,\hat{v}]$,
  confidence measures $\psi_{\text{bootg}}(i)$, $\psi_{\text{boota}}(i)$.
1 Calculate flow $[\hat{u}\,\hat{v}]$ using the CLG algorithm, Eqs. (5–8)
2 **for** $b = 1$ to B **do**
3     $\beta_i \leftarrow 0$ for all $i \in \Omega$
4   **for** $j = 1$ to $|\Omega|$ **do**
5        $\beta_k \leftarrow \beta_k + 1$,   choose k randomly from Ω
7   Calculate flow $[u^{(b)}\,v^{(b)}]$ using CLG with β,
      Eqs. (9–11)
8 Calculate $\psi_{\text{bootg}}(i)$, $\psi_{\text{boota}}(i)$ using Eqs. (15,17).

---

bootstrap energy functions $E^{(b)}$ and for each of them find the flow estimate $(\hat{u},\hat{v})^{(b)}$ as a solution of Eqs. (6–8).

For each $\mathsf{X}^{(b)}$, the corresponding energy function is

$$E^{(b)} = \sum_{i \in \Omega} \beta_i e_D(i) + \alpha e_S(i), \qquad (9)$$

where $\beta_i \in \mathbb{Z}_0^+$ is a multiplicity function[5], representing the number of times a pixel i appears in the multiset $\mathsf{X}^{(b)}$. Multiplicity functions are generated randomly (see Algorithm 1). The bootstrap energy functions are obtained by replacing contributions from some pixels by others. Note that the bootstrap process is applied only to the data part of the criterion e_D, not the smoothing part e_S, because only the data (image) part is stochastic and causes the variability; the smoothing part e_S is identical for all bootstrap realizations.

The modified bootstrap energy (Eq. 9) is minimized by solving the following system of linear equations, which is a simple modification of Eqs. (7–8):

$$0 = \frac{1}{h^2} \sum_{j \in N(i)} (u_j - u_i) - \frac{\beta_i}{\alpha}\left(J_i^{11}u_i + J_i^{12}v_i + J_i^{13}\right), \qquad (10)$$

$$0 = \frac{1}{h^2} \sum_{j \in N(i)} (v_j - v_i) - \frac{\beta_i}{\alpha}\left(J_i^{12}u_i + J_i^{22}v_i + J_i^{23}\right). \qquad (11)$$

3.2. Error estimation calculation

The system given by Eqs. (10,11) is solved B times for randomly generated multisets $\mathsf{X}^{(b)}$ (represented by multiplicity functions β_i), yielding B solutions[6] $[u^{(b)}\,v^{(b)}]$. We calculate the directional variances at all positions:

$$\sigma_u^2(i) = \text{var}_b\{u_i^{(b)}\} = \frac{1}{B}\sum_{b=1}^{B}\left(u_i^{(b)} - u_i^{(*)}\right)^2, \qquad (12)$$

$$\sigma_v^2(i) = \text{var}_b\{v_i^{(b)}\} = \frac{1}{B}\sum_{b=1}^{B}\left(v_i^{(b)} - v_i^{(*)}\right)^2 \qquad (13)$$

with the bootstrap estimates of the mean flow

$$u_i^{(*)} = \frac{1}{B}\sum_{b=1}^{B} u_i^{(b)} \quad \text{and} \quad v_i^{(*)} = \frac{1}{B}\sum_{b=1}^{B} v_i^{(b)}. \qquad (14)$$

The variances σ_u^2, σ_v^2 can be calculated using a numerically stable single-pass algorithm [19, 42], so that the individual bootstrap results $[u^{(b)}\,v^{(b)}]$ do not have to be stored, making the memory consumption independent of B. The total standard deviation at each pixel

$$\psi_{\text{bootg}}(i) = \sqrt{\sigma_u^2(i) + \sigma_v^2(i)} \qquad (15)$$

is a scalar quantity which estimates the geometrical error ε (2) and can be used as an uncertainty measure. This method is denoted *bootg* and it corresponds to a mean squared error. It is also possible to calculate the mean geometric error

$$\psi_{\text{bootgM}}(i) = \frac{1}{B}\sum_{b=1}^{B}\sqrt{(u_i - u_i^{(*)})^2 + (v_i - v_i^{(*)})^2} \qquad (16)$$

but we found that the results are very similar to *bootg* and a single-pass algorithm cannot be used, increasing memory consumption. We will therefore use only *bootg* in the experiments.

Bootstrap can also estimate the angular error ϕ (Eq. 3):

$$\psi_{\text{boota}}(i) = \frac{1}{B}\sum_{b=1}^{B} \text{angle}\left([u_i^{(b)}\,v_i^{(b)}], [\hat{u}_i\,\hat{v}_i]\right). \qquad (17)$$

Note that we have replaced the bootstrap mean $[u^{(*)}\,v^{(*)}]$ by the results $[\hat{u}\,\hat{v}]$ of a normal run of the OF algorithm without any resampling, again in order to avoid the need to store all B calculated flow fields.

3.3. Implementation

The system given by Eqs. (10,11) is solved by the successive overrelaxation (SOR) method [8, 36] with a relaxation parameter $\omega = 1.95$. Iteration is stopped after a fixed number of iterations (typically $100 \sim 1000$), when the ℓ_2 norm of the difference between $[u\,v]$ in two subsequent iterations is smaller than a given threshold (10^{-3}), or when the ℓ_2 norm of the residual in Eqs. (10,11) is smaller than another threshold (10^{-2}).

The CLG optical flow method is applied in a multiresolution fashion, as described in Bruhn et al. [8]. A multiresolution pyramid consisting of images with progressively decreasing size is created recursively from the input images by smoothing and downsampling, as long as the images are bigger than a predefined minimum size (32×32 pixels). The CLG algorithm is first applied on the coarsest level and the resulting motion field is used to warp the images at the next finer level. This is repeated recursively until the finest level is reached. The final motion field is obtained as a sum of the partial motion fields at all levels of the pyramid. The regularization part of the criterion $e_S(i)$ is calculated always on the total motion field. The set of equations (10,11) is modified as follows:

---

[5]a generalization of an indicator function
[6]For notational simplicity, we denote the partial bootstrap results as $[u^{(b)}\,v^{(b)}]$ instead of $[\hat{u}^{(b)}\,\hat{v}^{(b)}]$.

4

Algorithm 2: Multiresolution optical flow estimation. The `reduce` operation stands for reducing an image size by half, reducing its resolution, `expand` is the inverse operation.

Input: Images $g(x,y,t)$, $t = \{0,1\}$, multiplicities β_i, parameters α, σ, ρ.

Output: Flow field $[u\ v]$

1 **return** `loop`$([u\ v],\mathbf{0},\{\beta_i\},\sigma,\rho)$

2 **function** `loop`$(g,[u^-\ v^-],\{\beta_i\},\sigma,\rho)$

3 **if** images are small enough **then**

4   Calculate flow $[u\ v]$ from g using the CLG algorithm, Eqs. (18–21)

5 **else**

6     $g^R \leftarrow$ `reduce` g

7     $\{\beta_i\}^R \leftarrow$ `reduce` $\{\beta_i\}$

8     $[u^-\ v^-]^R \leftarrow$ `reduce` $[u^-\ v^-]/2$

9     $[u'\ v']^R \leftarrow$ `loop`$\big(g^R,[u^-\ v^-]^R,\{\beta_i\}^R,\sigma/2,\rho/2\big)$

10     $[u'\ v'] \leftarrow 2\,\text{expand}[u'\ v']^R$

11     $g^W \leftarrow$ `warp` $g(x,y,0)$ with $[u'\ v']$

12     $[u^-\ v^-] \leftarrow [u^-\ v^-] + [u'\ v']$

13   Calculate flow $[u\ v]$ from g^W and $[u^-\ v^-]$ by solving Eqs. (18–21)

14   **return** $[u\ v]$

$$0 = \frac{1}{h^2}\Big(s_i + \sum_{j\in N(i)}(u_j - u_i)\Big) - \frac{\beta_i}{\alpha}\big(J_i^{11}u_i + J_i^{12}v_i + J_i^{13}\big), \quad (18)$$

$$0 = \frac{1}{h^2}\Big(t_i + \sum_{j\in N(i)}(v_j - v_i)\Big) - \frac{\beta_i}{\alpha}\big(J_i^{12}u_i + J_i^{22}v_i + J_i^{23}\big), \quad (19)$$

$$s_i = \sum_{j\in N(i)}(u_j^- - u_i^-), \quad (20)$$

$$t_i = \sum_{j\in N(i)}(v_j^- - v_i^-), \quad (21)$$

where $[u^-\ v^-]$ is the total accumulated flow from previous resolution levels and $[u\ v]$ is the motion field increment being calculated at the current level.

Bootstrap estimation is incorporated into the multiresolution framework. The coefficients β_i are generated once at the finest level and subsequently reduced along with the images. Algorithm 2 summarizes the multiresolution procedure for finding the motion field. It is called from Algorithm 1, line 7, to obtain multiresolution bootstrap estimates.

All experiments were run with a very low number of bootstrap repetitions, $B = 10$. This is enough to approximately calculate a variance-type statistics [12, 25]. While a higher B improves the results slightly, the improvement does not outweigh the increased computation time [25].

4. Alternative uncertainty and confidence measures

Several alternative uncertainty measures were implemented and used in the experiments for comparison with the bootstrap method. Some of the measures could be simplified by an equivalent monotonous transformation; however, we have preferred to keep the original form as found in the literature, except for changing the sign to convert a confidence measure into an uncertainty measure, so that a low value of an uncertainty measure ψ corresponds to high confidence and vice versa.

4.1. Fast registration accuracy estimation (FRAE)

Fast registration accuracy estimation (*FRAE*) [24, 25] is a simple and fast method based on well known quadratic sensitivity analysis ideas, which we have modified for the CLG method (see Section 2 for notation). First we estimate the variance of the criterion contributions E_i (Eq. 5) for each pixel. Since only one pair of images is given, we use the spatially smoothed version of E_i as an approximation of its mean:

$$\sigma_E^2(i) = \text{Var}\{E_i\} \approx K_\rho' * \big(E_i - E_i * K_\rho'\big)^2,$$

where K_ρ' is a discretized and normalized version of an isotropic Gaussian spatial filter with standard deviation ρ. Second, we calculate the diagonal elements of the Hessian of E with respect to u_i, v_i:

$$H_{ii}^u = \frac{\partial^2 E}{\partial u_i^2} = 2h^2 J_i^{11} + 2(|N_i| + 1)\alpha,$$

$$H_{ii}^v = \frac{\partial^2 E}{\partial v_i^2} = 2h^2 J_i^{22} + 2(|N_i| + 1)\alpha,$$

where $|N_i|$ is the number of neighbors of each pixel (normally $|N_i| = 4$). For computational tractability, we assume that off-diagonal elements can be neglected. The *FRAE* estimate of the variances of u and v are then

$$\sigma_u^2(i) = \lambda\,\sigma_E(i)/H_{ii}^u,$$

$$\sigma_v^2(i) = \lambda\,\sigma_E(i)/H_{ii}^v,$$

where λ is a constant depending weakly on the approximations in the chosen *FRAE* variant and its parameters, such as a confidence level [24, 25]. We have used $\lambda = 2$. It has no effect on the experimental evaluation since only relative values are used. Using $\sigma_u^2(i)$, $\sigma_v^2(i)$ given above, the uncertainty measure is

$$\psi_{\text{fraeg}} = \sqrt{\sigma_u^2(i) + \sigma_v^2(i)}, \quad (22)$$

estimating the mean squared error similarly to *bootg* (Eq. 15)

$$\psi_{\text{fraeg}} \approx \sqrt{\text{Var}[u_i] + \text{Var}[v_i]}.$$

FRAE can also be used to estimate the uncertainty with respect to the angular error measure ϕ (Eq. 3) using a well-known formula $\text{Var}[f(x)] \approx f'\big(\text{E}[x]\big)^2 \text{Var}[x]$. Expanding with respect to both u and v yields

$$\psi_{\text{fraea}}(i) = \frac{180}{\pi}\sqrt{\frac{u_i^2\sigma_u^2(i) + v_i^2\sigma_v^2(i)}{\sigma_u^2(i) + \sigma_v^2(i)}} \quad (23)$$

as an estimate of

$$\psi_{\text{fraea}}(i) \approx \sqrt{\text{Var}[\phi_i]}.$$

5

4.2. Compliance with a learned motion model (PcaPVal)

The measure *PcaPVal* was proposed by Kondermann et al. [22]. It is generally applicable, which means that it can be used to estimate the reliability of flow vectors computed by an arbitrary OF method. The basic idea is to estimate a local model of the OF field based on training data. The derived model consists of the first and second order moments of the flow field patch distribution conditioned on the central vector. To obtain a confidence value for each flow vector a hypothesis test is carried out based on a suitable test statistic $d_M([u_i \, v_i])$. We convert d_M into a corresponding *p*-value, obtaining the following uncertainty measure:

$$\psi_{\text{PcaPVal}} = -\inf\{\alpha; d_M(\vec{v}_i) > G^{-1}(1-\alpha)\}, \quad (24)$$

where $G^{-1} : [0,1] \to \mathbb{R}^+$ is the inverse of the empirical cumulative distribution function computed from the training data.

4.3. Smallest eigenvalue of the structure tensor (StrEv3)

Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ stand for the three eigenvalues of the structure tensor [13] of $g(x,y,t)$ at a particular pixel. The measure *StrEv3* [14] is based on the following concept: The smaller the λ_3, the more likely $g(x,y,t)$ is locally flat in some direction. This is the case if the speed is zero, in case of an aperture problem or within homogeneous regions. The uncertainty measure is

$$\psi_{\text{StrEv3}} = \frac{1}{(1+\lambda_3)^2}. \quad (25)$$

4.4. Structure tensor total coherence (StrCt)

StrCt stands for the total coherence measure of the structure tensor. It is based on the same idea as *StrEv3* [14]. The uncertainty function is defined as

$$\psi_{\text{StrCt}} = -\left(\frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_3}\right)^2. \quad (26)$$

The advantage of this measure compared to the *StrEv3* is that it takes into account the anisotropy of the structure tensor. The measure is equal to -1 if $\lambda_1 \gg \lambda_3$ and it is equal to 0 if $\lambda_1 \approx \lambda_2 \approx \lambda_3$ and no movement can be computed in case of noise or homogeneous regions.

4.5. Structure tensor spatial coherence (StrCs)

StrCs stands for the spatial coherence measure of the structure tensor [14]. If we have an aperture problem and assuming that the brightness constancy equation holds, then there are two locally flat directions: the temporal direction and the direction along the object that causes the aperture problem. Therefore, the two smallest eigenvalues $\lambda_2 \geq \lambda_3$ of the structure tensor are nearly zero. This property can be measured by the spatial coherency measure *StrCs*:

$$\psi_{\text{StrCs}} = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2. \quad (27)$$

StrCs is high in case of an apperture problem and small otherwise.

4.6. Structure tensor corner measure (StrCc)

StrCc stands for the corner measure of the structure tensor [14]. It is defined as the difference between the total coherence measure *(StrCt)* and the spatial coherence measure *(StrCs)*.

$$\psi_{\text{StrCc}} = -\left(\frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_3}\right)^2 + \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2. \quad (28)$$

In this way, *StrCc* returns low values in locations where *StrCt* and StrCs are both small.

4.7. Gradient measure (Grad)

The idea behind the gradient measurement [4] is that the displacement field can be computed the more reliably the more texture is contained in the image. We use central differences[7] to compute the image gradient $\nabla_2 g = [g_x \, g_y]$.

$$\psi_{\text{grad}} = \frac{1}{(1 + \|\nabla_2 g\|)^2}. \quad (29)$$

4.8. Cost function based confidence measure (BWS)

Bruhn et al. [6, 8] propose to use directly the pixel contributions E_i from Eq. (5) at convergence as uncertainty measures. This combines information from both image and the motion field. The reasoning is that when the energy after registration is high, either the difference between the registered images is high or the deformation does not correspond to the smoothness assumptions of the regularization term. In these cases the correspondence is likely to be wrong. This method will be denoted *BWS*:

$$\psi_{\text{BWS}}(i) = E_i. \quad (30)$$

4.9. Ideal uncertainty measures

For comparison, we evaluate our criteria also using the true geometrical and angular errors as uncertainty measures:

$$\psi_{\text{idealg}}(i) = \varepsilon_i, \quad (31)$$
$$\psi_{\text{ideala}}(i) = \phi_i. \quad (32)$$

This represents the best achievable results for ε and ϕ, respectively.

5. Experiments

We have used 58 standard and freely available OF test sequences [3, 33]. Due to space limitations only a part of the experimental results can be shown here, see our technical rapport [26] for a more complete version. We want to emphasize that the purpose of our experiments is to compare the confidence measures between themselves rather than to test the motion estimation algorithm *per se*. For this reason, we

---

[7]The kernel is $[-0.5 \, \underline{0} \, 0.5]$, for compatibility with Barron et al. [4].

have not performed any extensive parameter tuning, for all sequences we have used the parameters $\alpha = 100 \sim 200$, $\sigma = 1.77$, $\rho = 2 \sim 4.55$.

We are aware that state-of-the art algorithms can be tuned to produce a truly negligible error on standard synthetic sequences, e.g. $1.02°$ error on the *Yosemite* sequence for the nonlinear 3D multiresolution variant of the CLG method [8]. However, this is not very useful for confidence measure testing, since in this case the error is small everywhere and the confidence measure is not needed. For more challenging sequences, reliable ground truth is usually not available.

Figure 1 shows example results of the linear 2D CLG method for the first two frames of the *Office* sequence [33]. The mean geometrical error is $\bar{\varepsilon} = 0.1$ pixels, mean angular error $\bar{\phi} = 5.5°$; errors occur because of aliasing (computer screen), shadows (on the table), occlusions (top of the chair) and in textureless regions with motion discontinuities (window). Note that the bootstrap method identifies the suspect regions very well, whereas the *BWS* (energy) method fails. The mean errors are evaluated over the whole image:

$$\bar{\varepsilon} = \text{mean}_i\, \varepsilon_i = \frac{1}{|\Omega|} \sum_{i \in \Omega} \varepsilon_i, \tag{33}$$

$$\bar{\phi} = \text{mean}_i\, \phi_i = \frac{1}{|\Omega|} \sum_{i \in \Omega} \phi_i. \tag{34}$$

5.1. Sparsification tests

A common approach for confidence measure evaluation is based on sparsification [8, 22], where pixelwise errors are ordered according to the confidence measure being tested and the mean error is calculated only using a given percentile of the best values. The idea is that we evaluate how successful a confidence measure is in identifying pixels with large motion estimation error. More formally, for the geometrical error ε we define a function $\hat{\varepsilon}(\xi)$ with $0 \leq \xi \leq 1$ as the mean geometrical error of the best $\xi|\Omega|$ pixels according to a given confidence measure and similarly for the angular error ϕ. If a mask is provided, masked pixels are ignored. The values $\hat{\varepsilon}(\xi)$ and $\hat{\phi}(\xi)$ are averaged over all frames in a sequence. Lower values for the same ξ mean a better confidence measure.

Figures 2–5 show results of the sparsification tests based on CLG flow fields for several synthetic and real test sequences [3, 33]. The middle and right columns show the results for the geometrical error $\hat{\varepsilon}(\xi)$ and the angular error $\hat{\phi}(\xi)$, respectively. We see that the bootstrap methods, *boota* and *bootg*, are almost always the best methods, leading to the lowest error at any relative number of retained pixel ξ, surpassed only by the ideal confidence measures *ideala* and *idealg* (in green). The *PcaPVal* method (dark green) is very good for sequences with motion field corresponding to the learnt model (such as the *Street* sequence) but fails for less common motion patterns (such as the *Sphere* sequence). Note also that sparsification using some methods can actually make the average error increase.

It is not possible to report the complete sparsification results here for all sequences, because of space limitations. Instead, we have ranked all confidence measures ψ for both angular

Table 1: Mean ranks with standard deviations for each confidence measure over all sequences and all fractions ξ of retained pixels, for the geometrical and angular errors. The rank within each column is given in parentheses.

Method	mean rank	
	geometrical error	angular error
ideala	3.134 ± 1.23 *(2)*	1.545 ± 1.72 *(1)*
idealg	1.545 ± 1.72 *(1)*	2.553 ± 1.42 *(2)*
boota	5.245 ± 0.56 *(4)*	4.306 ± 0.86 *(3)*
bootg	3.356 ± 1.16 *(3)*	4.763 ± 0.74 *(4)*
fraea	8.163 ± 0.65 *(8)*	8.008 ± 0.36 *(7)*
fraeg	8.120 ± 0.44 *(7)*	7.915 ± 0.31 *(6)*
bws	8.758 ± 0.67 *(10)*	9.172 ± 0.74 *(9)*
grad	7.841 ± 0.58 *(6)*	9.295 ± 0.83 *(10)*
pcaPVal	6.497 ± 0.37 *(5)*	5.720 ± 0.74 *(5)*
strCc	8.692 ± 0.65 *(9)*	8.120 ± 0.43 *(8)*
strCs	9.465 ± 0.81 *(11)*	10.399 ± 1.08 *(13)*
strCt	9.654 ± 0.87 *(12)*	9.614 ± 0.84 *(12)*
strEv3	10.530 ± 1.13 *(13)*	9.591 ± 0.88 *(11)*

and geometrical errors, for each sequence, and for 10 values of the fraction ξ of retained pixels $\xi \in \{0.1, 0.2, \ldots 1.0\}$. We are reporting the mean values over all sequences and over all ξ in Table 1. We have observed (see our technical report [26] that contains a more detailed breakdown of the results) that the bootstrap method *bootg* had the smallest geometric error $\hat{\varepsilon}(\xi)$ and the bootstrap method *boota* had the smallest angular error $\hat{\phi}(\xi)$, not counting the "ideal" confidence measures. The best non-bootstrap method in both cases was *pcaPVal*.

5.2. Average correctness

Another way of evaluating the performance of confidence measures is to calculate the average correctness, which is defined as the relative number of cases in which the comparison $\psi_i < \psi_j$ between uncertainty measures for two pixels i, j gives the same results as a comparison between the true errors $\varepsilon_i < \varepsilon_j$ or $\phi_i < \phi_j$, respectively. We use 10^6 randomly chosen pixel pairs.

The results in Table 2 show that on the average, *bootg* and *boota* are the most correct method (not counting the "ideal" methods) for the geometrical and angular errors, respectively. In both cases, *pcaPVal* is the next best method. See our technical report [26] for more details.

5.3. Reliable pixel selection

The uncertainty information can be used to improve the final quality of the optical flow estimation by masking locations where the movement estimate is unreliable. Unreliable locations (outliers) are found by thresholding the confidence measure, where a suitable threshold can be determined statistically. Then an improved optical flow can be found by motion inpainting [20], or, as we have done here, by rerunning the optical flow computation given by Eqs. (9–11) with $\beta_i = 1$ for pixels to be kept and $\beta_i = 0$ for pixels to be ignored.
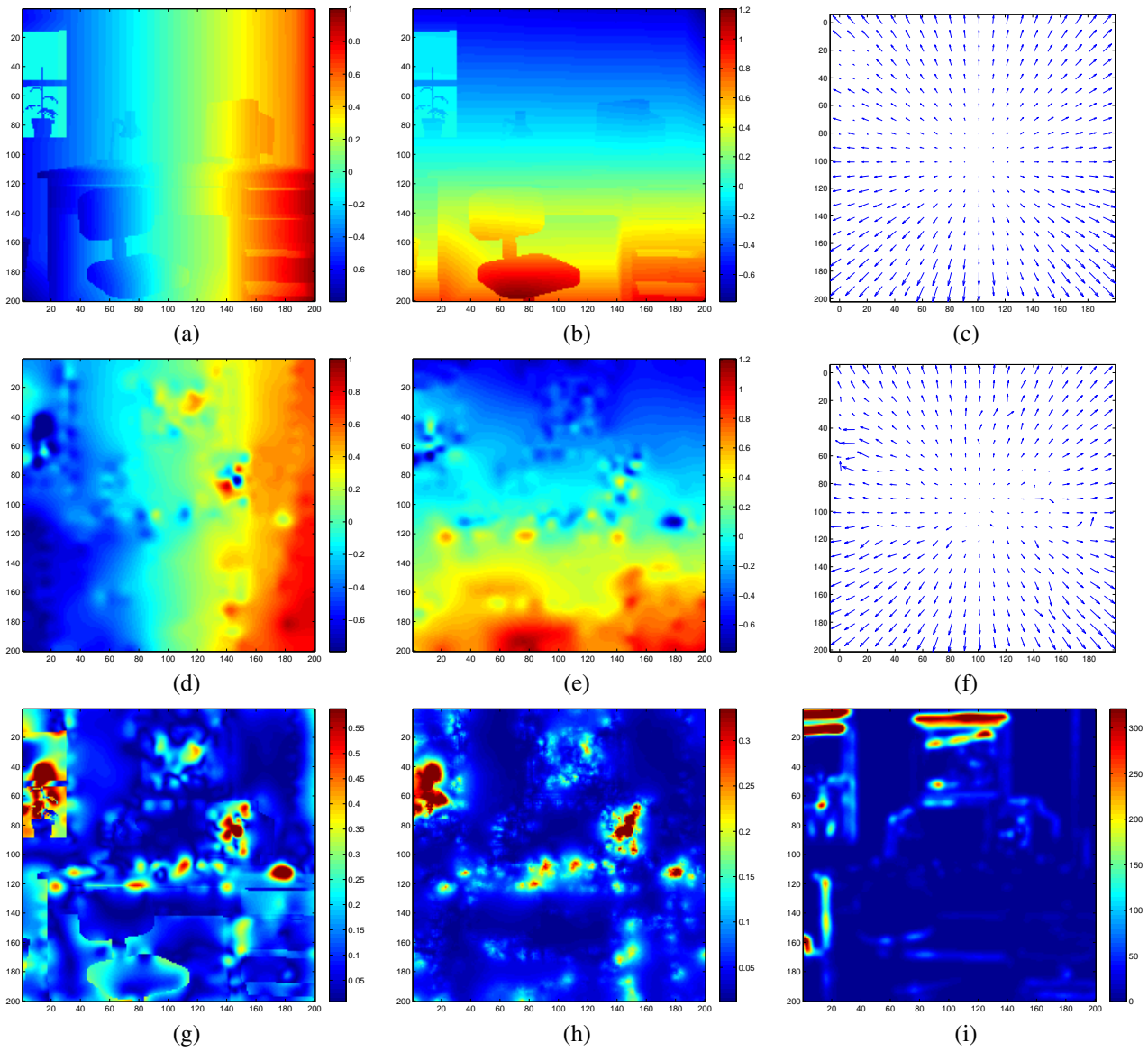
Figure 1: The *Office* sequence: (a) true *x* displacement, (b) true *y* displacement, (c) true displacement shown as a vector field, (d) calculated *x* displacement, (e) calculated *y* displacement, (f) calculated displacement shown as a vector field, (g) true geometrical error, (h) geometrical error estimated using bootstrap, (i) geometrical error estimated using the *BWS* method. In the color images, blue corresponds to low value and red to high values. The first frame of the sequence can be seen in the top left of Fig. 3.
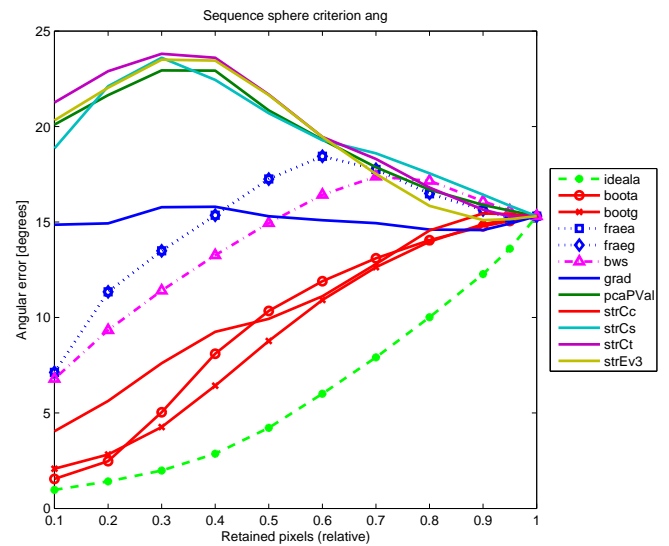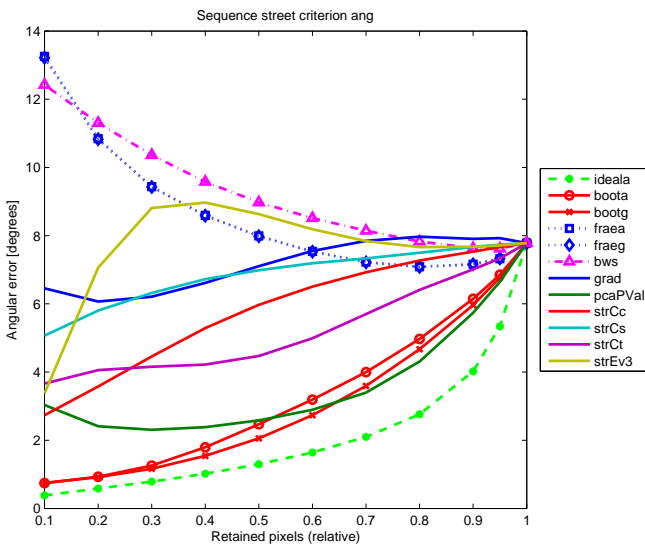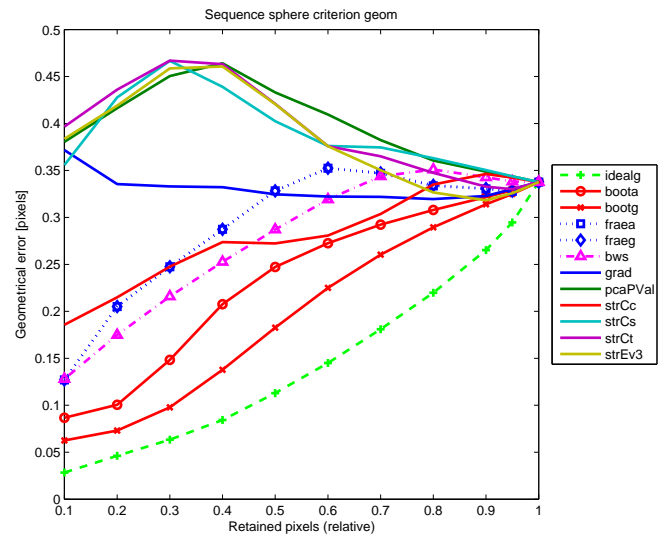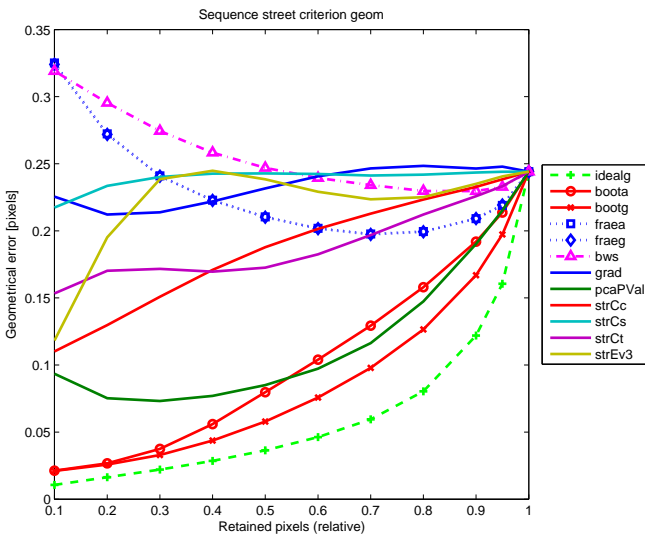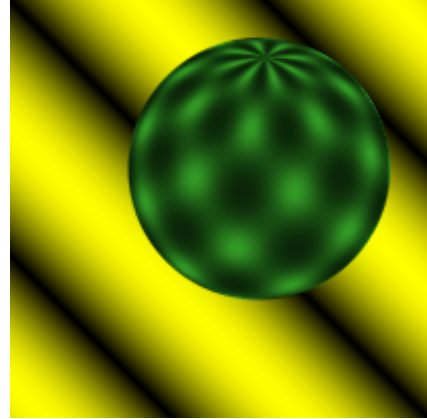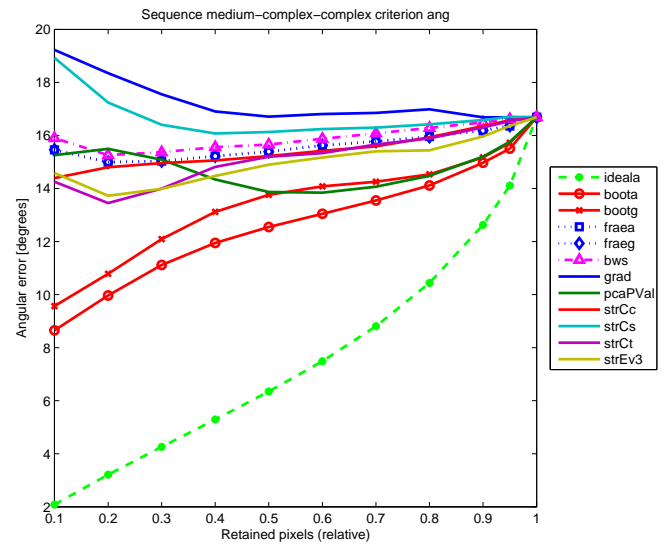
Figure 2: Sparsification tests for the geometrical error (middle row) and angular error (bottom row) for synthetic sequences *Street* and *Sphere* (from left to right) [33]. Each line in the graphs corresponds to one confidence measure and shows the mean erors $\varepsilon(\xi)$ and $\phi(\xi)$ with respect to a fraction ξ of retained pixels. The first row shows the first images of the sequences.
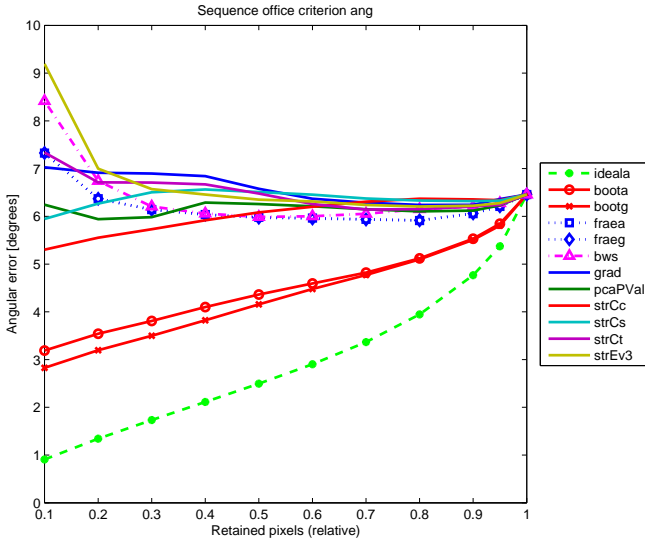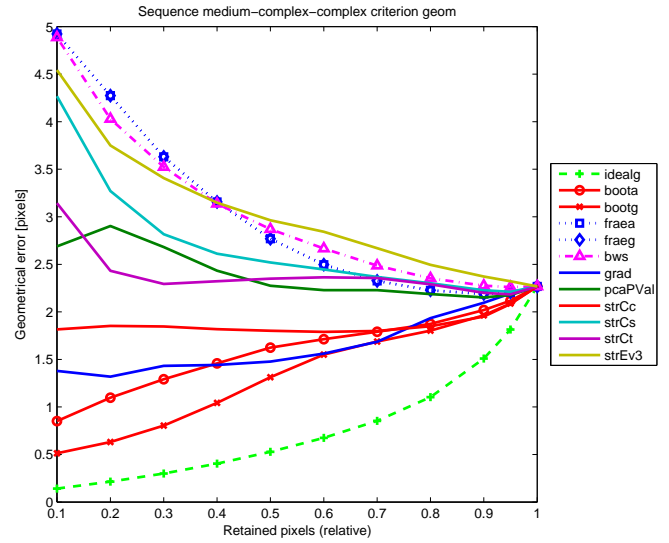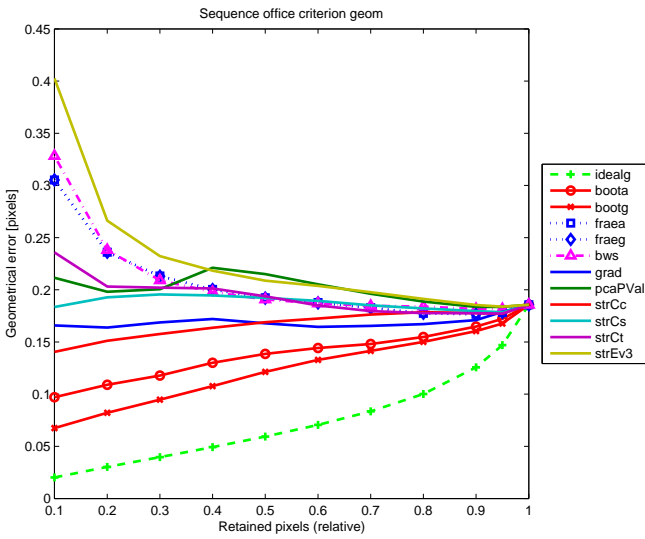
Figure 3: Sparsification tests for the geometrical error (middle row) and angular error (bottom row) for synthetic sequences *Office* and *Medium-complex-complex* (from left to right) [3, 33]. Each line in the graphs corresponds to one confidence measure and shows the mean errors $\varepsilon(\xi)$ and $\phi(\xi)$ with respect to a fraction ξ of retained pixels. The first row shows the first images of the sequences.
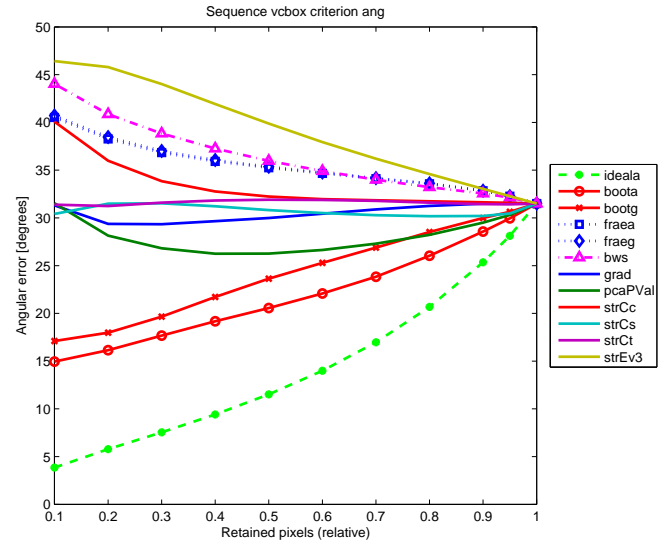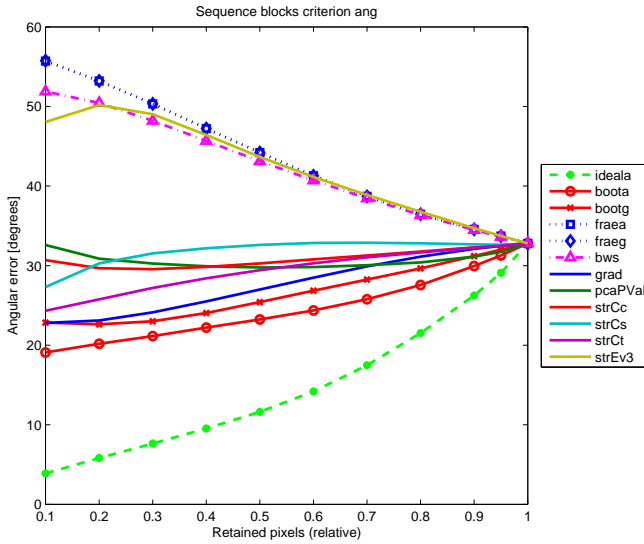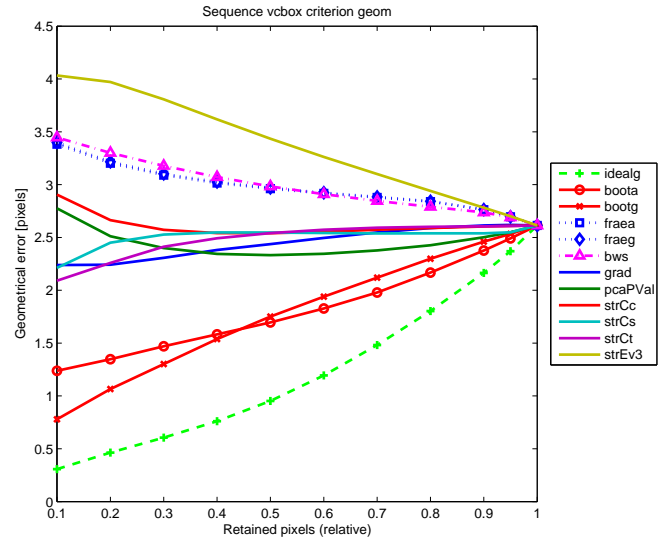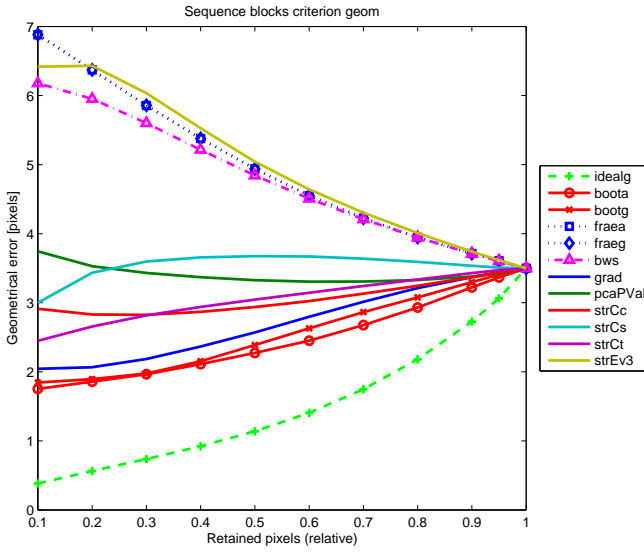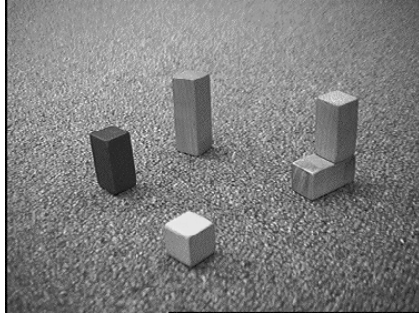
Figure 4: Sparsification tests for the geometrical error (middle row) and angular error (bottom row) for real sequences *Blocks* and *Vcbox* (from left to right) [33]. Each line in the graphs corresponds to one confidence measure and shows the mean erors $\varepsilon(\xi)$ and $\phi(\xi)$ with respect to a fraction ξ of retained pixels. The first row shows the first images of the sequences.

Figure 5: Sparsification tests for the geometrical error (middle row) and angular error (bottom row) for real sequences *Rubber Whale* and *Dimetrodon* (from left to right) [3]. Each line in the graphs corresponds to one confidence measure and shows the mean erors $\varepsilon(\xi)$ and $\phi(\xi)$ with respect to a fraction ξ of retained pixels. The first row shows the first images of the sequences.
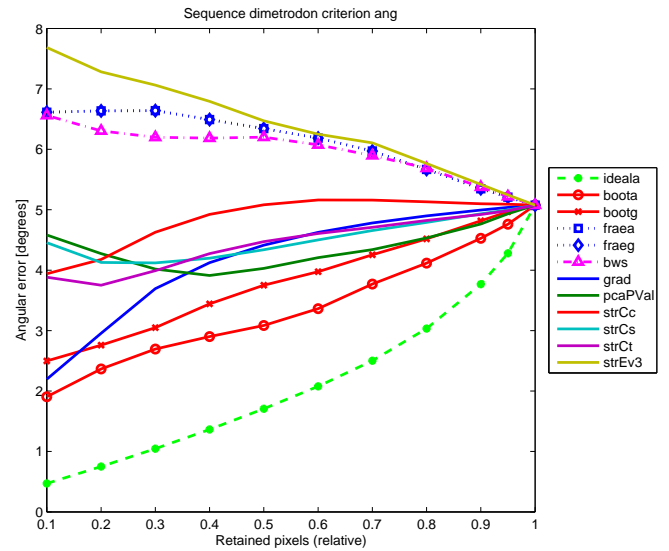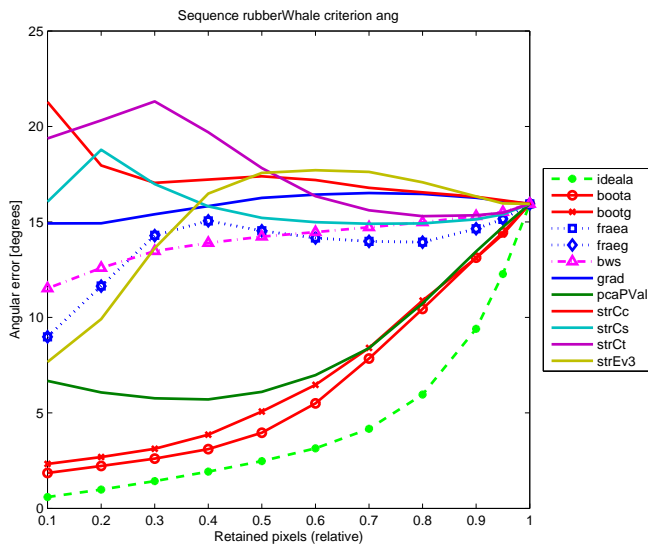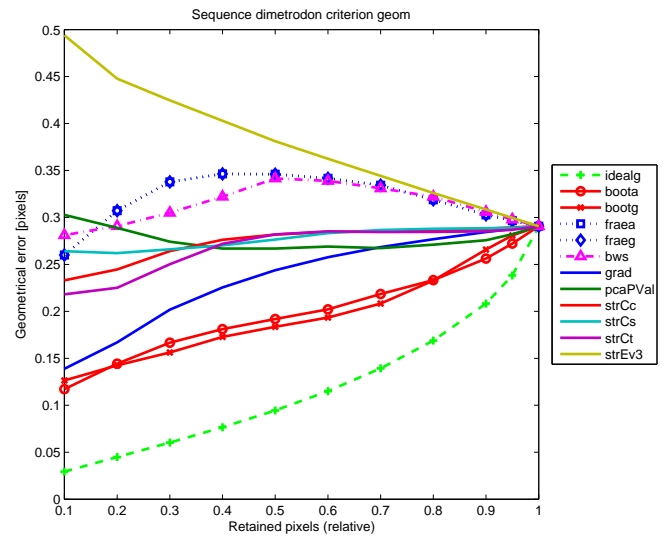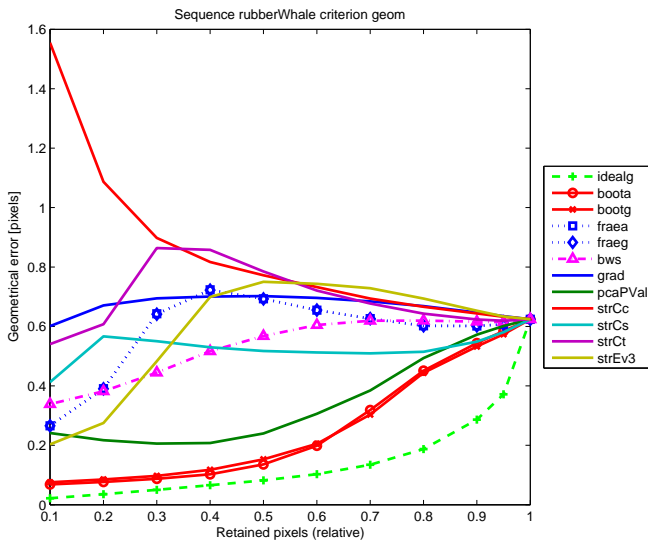
12

Table 2: Average correctness for the geometrical and angular errors averaged over all sequences for all confidence measures. We have also ranked the confidence measures for each sequence and we report the average ranks. The numbers in parentheses are the ranks within each column.

| Method | geometrical error | | angular error | |
	correctness	rank	correctness	rank
ideala	0.810 *(2)*	2.167 *(2)*	1.000 *(1)*	1.000 *(1)*
idealg	1.000 *(1)*	1.000 *(1)*	0.810 *(2)*	2.000 *(2)*
boota	0.648 *(4)*	4.694 *(4)*	0.645 *(3)*	3.806 *(3)*
bootg	0.703 *(3)*	3.111 *(3)*	0.619 *(4)*	4.444 *(4)*
fraea	0.523 *(7)*	8.306 *(7)*	0.509 *(8)*	8.528 *(8)*
fraeg	0.523 *(8)*	8.639 *(8)*	0.509 *(9)*	8.583 *(9)*
bws	0.505 *(10)*	9.500 *(11)*	0.492 *(11)*	9.889 *(11)*
grad	0.546 *(6)*	7.556 *(6)*	0.525 *(7)*	8.444 *(7)*
pcaPVal	0.579 *(5)*	6.222 *(5)*	0.572 *(5)*	5.417 *(5)*
strCc	0.508 *(9)*	9.444 *(9)*	0.531 *(6)*	7.667 *(6)*
strCs	0.500 *(11)*	9.500 *(11)*	0.483 *(12)*	10.778 *(13)*
strCt	0.491 *(12)*	9.500 *(11)*	0.494 *(10)*	9.778 *(10)*
strEv3	0.447 *(13)*	11.361 *(13)*	0.467 *(13)*	10.667 *(12)*

In Table 3 we show the relative improvement of the mean geometrical error (2) thanks to masking unreliable pixels in four sequences. To allow for a direct comparison between different confidence measures, the percentage of retained pixels was fixed to 95% and 90%. We can see that confidence measures lead to an improvement of the geometric error and the improvement is often substantial (e.g. the *Yossemite* or *Office* sequence). The largest improvement is in all cases but one obtained by the bootstrap *bootg* method.

5.4. Sequence registration

We address the task of robust finding of point trajectories from a video sequence (Fig. 6), in our case a 2D ultrasound sequence of a breast phantom for elastography [35]. It contains 48 frames representing about 5 s; the tissue is periodically compressed by a hand-held ultrasound probe. Recovering the trajectory of all points is a first step in estimating the strain and mechanical properties such as the Young modulus, which has high diagnostic value for identifying hard lesions, which are likely to be tumors.

The two standard approaches for image sequence registration are either (i) to register pairs of consecutive images and then accumulate the movement, or (ii) to choose one of the images as a reference and to register the remaining ones with respect to the reference, using displacement from neighboring frames for initialization. The first approach suffers from accumulation of registration errors; in the second case the registration is difficult because of important geometrical and appearance differences between the images being registered. These approaches use either none or all of the intermediate frames between the two frames of interest. Instead, in a novel approach we propose here, we select a suitable subset of intermediate frames, thus generalizing the previous approaches. The selection will be guided by the bootstrap uncertainty estimation.

Given a sequence of N frames, we apply the previously described optical flow algorithm with bootstrap uncertainty es-

Table 3: Mean geometrical error of OF estimation with 5% and 10% percent of pixels masked out. The pixels to retain were determined according to five different uncertainty measures (denoted 'method'). The error is reported relative to the original OF estimation geometric error (with no pixels removed). The number in parentheses are ranks within each column, for each sequence separately.

| sequence | method | Retained pixels (relative) | |
		0.95	0.90
street	bws	0.941 *(3)*	0.919 *(1)*
street	grad	0.998 *(5)*	0.995 *(5)*
street	pcaPVal	0.983 *(4)*	0.990 *(4)*
street	fraeg	0.931 *(2)*	0.923 *(3)*
street	bootg	0.913 *(1)*	0.920 *(2)*
office	bws	0.777 *(3)*	0.738 *(3)*
office	grad	0.999 *(5)*	0.989 *(5)*
office	pcaPVal	0.930 *(4)*	0.897 *(4)*
office	fraeg	0.745 *(2)*	0.708 *(2)*
office	bootg	0.632 *(1)*	0.595 *(1)*
blocks	bws	0.989 *(2)*	0.987 *(3)*
blocks	grad	0.998 *(5)*	0.995 *(5)*
blocks	pcaPVal	0.991 *(3)*	0.984 *(2)*
blocks	fraeg	0.993 *(4)*	0.989 *(4)*
blocks	bootg	0.988 *(1)*	0.978 *(1)*
yosemite	bws	0.751 *(2)*	0.660 *(2)*
yosemite	grad	0.992 *(5)*	0.986 *(5)*
yosemite	pcaPVal	0.951 *(4)*	0.898 *(4)*
yosemite	fraeg	0.757 *(3)*	0.680 *(3)*
yosemite	bootg	0.706 *(1)*	0.638 *(1)*

timation (Section 3.1) on each pair of frames (i, j) such that $|i - j| \in \{1, 2, 4, 8, 16 \dots\}$. This increases the computational complexity only by a factor $\log_2 N$. For each point that needs to be tracked, we recursively build a graph starting with a chosen reference frame, with nodes corresponding to intermediate frames and edge weights being the estimated variances $\psi_{\text{bootg}}^2 = \sigma_u^2 + \sigma_v^2$ (Eq. 15) at that point. The final displacement is calculated by accumulating the partial displacements along the path with the smallest total variance.

The improvement brought by this technique is spectacular (Fig. 6). The consecutive registration seriously underestimates the movement for three points out of four. The reference-based registration mostly underestimates less (except for point 1) but is less robust, giving incorrect results on several frames. In contrast, the bootstrap based graph technique successfully tracks all points with a good accuracy. Visual observation suggests that the tracking is often even better than the manual one. Numerically, the mean geometrical error with respect to the consecutive registration has been reduced $2 \sim 8$ times for each individual point, the mean error over all points decreased from 45.9 pixels to 10.4 pixels.

6. Conclusions and discussion

We have shown how to apply the bootstrap resampling method to estimate pixelwise the geometrical and angular error for optical flow algorithms. The method is applicable to any motion estimation technique that can be formulated in a variational setting, as a minimization of a criterion which can be decomposed as a set of pixel contributions. The only input to the method is the pair of images being registered and the registration algorithm itself.

No confidence measure we have tested is perfect. Even the best ones are relatively far from the best achievable solution. This shows that estimating accuracy from the input images alone without additional information about the underlying physical reality is a hard problem.

The major shortcoming of the bootstrap method is that it can only estimate the variance part of the error, not the bias. So if all bootstrap OF calculations fail in the same way, we will be mislead to believing that the results are accurate. However, this problem is not specific to bootstrap—if a model does not correspond to the reality, all methods will fail. On the other hand, bootstrap can predict well problems due to occlusions, motion discontinuities, as well as texture-less regions. In the last case, however, the image must either contain some sensor noise or we must add noise corresponding to the measurement uncertainty during the bootstrap process, otherwise in completely homogeneous regions the variance estimate would be zero. The bootstrap estimates are relatively well correlated with the true errors but the absolute values are not yet very reliable.

Another missing piece is the theoretical justification that the bootstrap estimate will converge, under what assumptions, and how fast. Alas, the idea of doing bootstrap on functions instead of values is very new and we know of no theory covering it. We have found experimentally on our data that the bootstrap

samples are only weakly correlated and locally identically distributed. This is reassuring as the standard bootstrap assumes i.i.d. samples. The weak correlation and independence can be explained by the fact that after convergence of the OF estimations the residuals are only due to the measurement noise.

We have shown experimentally that for CLG flow fields the bootstrap technique leads to a better confidence measure than all other confidence measures tested. It is true that bootstrap typically increases the computational complexity by a factor of $B = 10$. However, we do not believe this to be a serious problem nowadays. Already in 2003, Bruhn et al. reported [7] that multigrid techniques allow for real time calculation of the OF by the CLG method on which we are based. Second, multicore machines are becoming the norm and bootstrap is trivially parallelisable. As an illustration, the 2×4 core machine we are now using can perform 10 OF calculations in about the same time that the computer we were using 2.5 years ago when this technique was originally developed needed to calculate one OF.

We believe that bootstrap image registration and optical flow accuracy estimation is an extremely general and revolutionary technique which will find many uses in practice.

7. Acknowledgements

[1] Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision **2**(3), 283–310 (1989)

[2] Bainbridge-Smith, A., Lane, R.: Measuring confidence in optical flow estimation. Electronics Letters **32**(10), 882–884 (1996)

[3] Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV: International Conference on Computer Vision, pp. 1–8 (2007). DOI 10.1109/ICCV.2007.4408903

[4] Barron, J.L., Fleet D. J. nad Beauchemin, S.S.: Performance of optical flow techniques. International Journal of Computer Vision (12), 43–77 (1994)

[5] Bigün, J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. IEEE Journal of Pattern Analysis and Machine Intelligence **13**(8), 775–790 (1991)

[6] Bruhn, A., Weickert, J.: A confidence measure for variational optic flow methods. In: R. Klette, R. Kozera, L. Noakes, J. Weickert (eds.) Geometric Properties for Incomplete data, pp. 283–298 (2006). DOI http://dx.doi.org/10.1007/1-4020-3858-8_15. URL http://dx.doi.org/10.1007/1-4020-3858-8_15

[7] Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Real-time optic flow computation with variational methods. In: N. Petkov, M. Westenberg (eds.) Proc. Computer Analysis of Images and Patterns (CAIP'03), *LNCS*, vol. 2756, pp. 222–229. Springer (2003)

[8] Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. Int. J. Comput. Vision **61**(3), 211–231 (2005)

[9] Cho, K., Meer, P., Javier, C.: Performance assessment through bootstrap. IEEE Trans. Pattern Anal. Mach. Intell. pp. 1185–1198 (1997)

[10] Dev, A., Kröse, B.J.A., Groen, F.C.A.: Confidence measures for image motion estimation. In: Real World Computing Symposium, pp. 199–206 (1997)

[11] Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Statist. (7), 1–26 (1979)

[12] Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. No. 57 in Monographs on Statistics and Applied Probability. Chapman & Hall, CRC (1993)
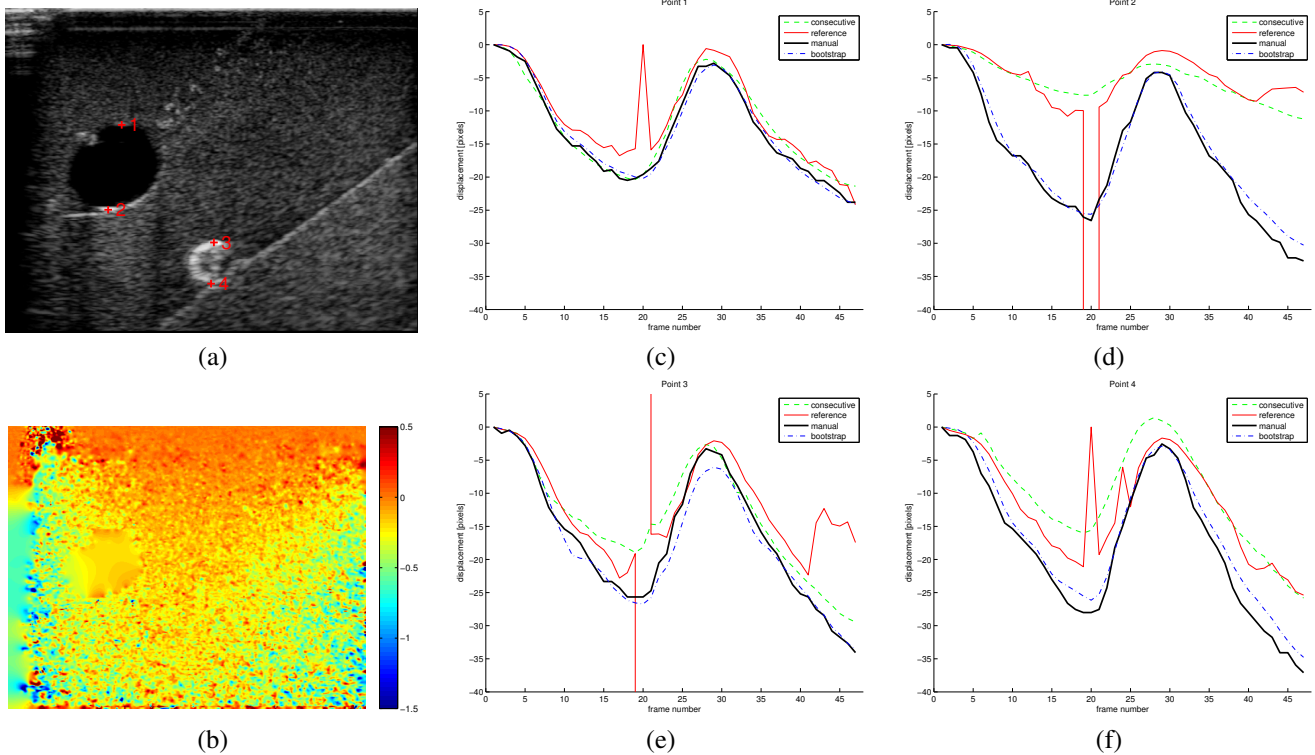
Figure 6: First image from an ultrasound elastography sequence of a breast phantom (a) with four landmarks and a corresponding recovered vertical flow (b). Graphs (c–f) show the vertical displacement of the four points recovered manually, by accumulating consecutive pairwise displacements, by direct registration of each frame with the first frame and by combining displacements selected using the bootstrap method.

[13] Förstner, W., Gulch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. Intercommission Conference on Fast Processing of Photogrammetric Data pp. 281–305 (1987)

[14] Haussecker, H., Spies, H.: Motion. In: B. Jähne, H. Haussecker, P. Geissler (eds.) Handbook of Computer Vision and Applications, vol. 2, chap. 13, pp. 336–338. Academic Press (1999)

[15] Heeger, D.J.: Optical flow using spatiotemporal filters. International Journal of Computer Vision **1**(4), 279–302 (1988)

[16] Horn, B., Schunck, B.: Determining optical flow. Artificial Inteligence **17**, 185–203 (1981)

[17] Jähne, B.: Spatio-Temporal Image Processing: Theory and Scientific Applications. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1993)

[18] Jain, A.K., Dubes, R.C., Chen, C.C.: Bootstrap techniques for error estimation. IEEE Trans. Pattern Anal. Mach. Intell. **9**(5), 628–633 (1987)

[19] Knuth, D.E.: The Art of Computer programming, *Computer Science and Information Processing*, vol. Fundamental Algorithms. Addison-Wesley (1968)

[20] Kondermann, C., Kondermann, D., Garbe, C.S.: Postprocessing of optical flows via surface measures and motion inpainting. In: G. Rigoll (ed.) DAGM-Symposium, *Lecture Notes in Computer Science*, vol. 5096, pp. 355–364. Springer (2008)

[21] Kondermann, C., Kondermann, D., Jähne, B., Garbe, C.: An adaptive confidence measure for optical flows based on linear subspace projections. In: Hamprecht, Schnörr, Jähne (eds.) Pattern Recognition, Lecture Notes in Computer Science, pp. 132–141. Springer-Verlag (2007)

[22] Kondermann, C., Mester, R., Garbe, C.S.: A statistical confidence measure for optical flows. In: D.A. Forsyth, P.H.S. Torr, A. Zisserman (eds.) ECCV (3), *Lecture Notes in Computer Science*, vol. 5304, pp. 290–301. Springer (2008)

[23] Künsch, H.R.: The jackknife and the bootstrap for general stationary observations. Ann. Statist. (17), 1217–1241 (1989)

[24] Kybic, J.: Fast no ground truth image registration accuracy evaluation: Comparison of bootstrap and Hessian approaches. In: Proceedings of 2008 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 792–795. IEEE Press, New York, US (2008). URL ftp://cmp.felk.cvut.cz/pub/cmp/articles/kybic/Kybic-ISBI2008.pdf

[25] Kybic, J.: Bootstrap resampling for image registration uncertainty estimation without ground truth. IEEE Trans. Image Processing **19**(1), 64–73 (2010). URL ftp://cmp.felk.cvut.cz/pub/cmp/articles/kybic/Kybic-ieeeTIP2009.pdf

[26] Kybic, J., Nieuwenhuis, C.: Bootstrap optical flow confidence and uncertainty measure. Research Report K333–10/10, CTU–CMP–2010–07, Department of Cybernetics, Faculty of Electrical Engineering Czech Technical University, Prague, Czech Republic (2010). URL ftp://cmp.felk.cvut.cz/pub/cmp/articles/kybic/Kybic-TR-2010-07.pdf

[27] Kybic, J., Smutek, D.: Image registration accuracy estimation without ground truth using bootstrap. In: R. Beichel, M. Sonka (eds.) CVAMIA: Computer Vision Approaches to Medical Image Analysis, no. 3117 in Lecture Notes in Computer Science, pp. 61–72. Springer (2006). DOI http://dx.doi.org/10.1007/11889762_6. URL ftp://cmp.felk.cvut.cz/pub/cmp/articles/kybic/Kybic-CVAMIA2006.pdf

[28] Lei, C., Yang, Y.: Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In: IEEE International Conference on Computer Vision (ICCV). Kyoto, Japan (2009)

[29] Lempitsky, V., Roth, S., Rother, C.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In: IEEE Computer Vision and Pattern Recognition (CVPR) (2008)

[30] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: P.J. Hayes (ed.) IJCAI, pp. 674–679. William Kaufmann (1981)

[31] Matei, B., Meer, P.: Optimal rigid motion estimation and performance evaluation with bootstrap. In: CVPR'99: Computer Vision and Pattern Recognition Conference, vol. 1, pp. 339–345 (1999)

[32] Matei, B., Meer, P., Tyler, D.: Performance assessment by resampling: Rigid motion estimators. In: K. Bowyer, P. Phillips (eds.) Empirical Evaluation Techniques in Computer Vision, pp. 72–95 (1998)

[33] McCane, B., Novins, K., Crannitch, D., Galvin, B.: On benchmarking

optical flow. Computer Vision and Image Understanding **84**(1), 126–143 (2001)

[34] Neumann, U., You, S.: Adaptive multi-stage 2D image motion field estimation. In: SPIE Conference on Applications of Digital Image Processing, vol. 3460, pp. 116–123 (1998)

[35] Ophir, J., Cespedes, I., Ponnekanti, H., Yazdi, Y., Li, X.: Elastography: A quantitative method for imaging the elasticity of biological tissues. Ultrasonic Imaging **13**(2), 111–134 (1991). DOI 10.1016/0161-7346(91)90079-W

[36] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C, second edn. Cambridge University Press (1992)

[37] Simoncelli, E.P., Adelson, E.H., Heeger, D.J.: Probability distributions of optical flow. In: Proc Conf on Computer Vision and Pattern Recognition, pp. 310–315. IEEE Computer Society, Mauii, Hawaii (1991). URL citeseer.ist.psu.edu/article/simoncelli91probability.html

[38] Singh, A.: An estimation-theoretic framework for image-flow computation. In: Proceedings of ICCV, pp. 168–177 (1990)

[39] Thévenaz, P., Ruttimann, U.E., Unser, M.: A pyramid approach to subpixel registration based on intensity. IEEE Transactions on Image Processing **7**(1), 1–15 (1998)

[40] Uras, S., Girosi, F., Verri, A., Torre, V.: A computetional approach to motion perception. Journal of Biological Cybernetics (60), 79–97 (1988)

[41] Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optic flow. In: IEEE International Conference on Computer Vision (ICCV). Kyoto, Japan (2009)

[42] Welford, B.P.: Note on a method for calculating corrected sums of squares and products. Technometrics **4**(3), 419–420 (1962)

[43] Zetzsche, C., Barth, E.: Fundamental limits of linear filters in the visual processing of two dimensional signals. Vision Research (30) (1990)

[44] Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.P.: Complementary optic flow. In: EMMCVPR '09: Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 207–220. Springer-Verlag, Berlin, Heidelberg (2009)

[45] Zoubir, A.M., Boashash, B.: The bootstrap and its applications in signal processing. IEEE Signal Processing Magazine pp. 56–76 (1998)

[46] Zoubir, A.M., Iskander, D.R.: Bootstrap Techniques for Signal Processing. Cambridge University Press (2004)

[47] Zoubir, A.M., Iskander, R.: Bootstrap methods and applications. IEEE Signal Processing Magazine **24**(4), 10–19 (2007)