Highly Overparameterized Optical Flow Using PatchMatch Belief Propagation

Michael Hornáček[1,⋆], Frederic Besse[2,⋆], Jan Kautz[2], Andrew Fitzgibbon[3], and Carsten Rother[4]

[1] TU Vienna, Austria
michael.hornacek@tuwien.ac.at
[2] University College London, UK
{f.besse,j.kautz}@cs.ucl.ac.uk
[3] Microsoft Research Cambridge, UK
awf@microsoft.com
[4] TU Dresden, Germany
carsten.rother@tu-dresden.de

Abstract. Motion in the image plane is ultimately a function of 3D motion in space. We propose to compute optical flow using what is ostensibly an extreme overparameterization: depth, surface normal, and frame-to-frame 3D rigid body motion at every pixel, giving a total of 9 DoF. The advantages of such an overparameterization are twofold: first, geometrically meaningful reasoning can be called upon in the optimization, reflecting possible 3D motion in the underlying scene; second, the 'fronto-parallel' assumption implicit in the use of traditional matching pixel windows is ameliorated because the parameterization determines a plane-induced homography at every pixel. We show that optimization over this high-dimensional, continuous state space can be carried out using an adaptation of the recently introduced PatchMatch Belief Propagation (PMBP) energy minimization algorithm, and that the resulting flow fields compare favorably to the state of the art on a number of small- and large-displacement datasets.

Keywords: Optical flow, large displacement, 9 DoF, PatchMatch, PMBP.

1 Introduction

One statement of the goal of optical flow computation is the recovery of a dense correspondence field between a pair of images, assigning to each *pixel* in one image a 2D translation vector that points to the pixel's correspondence in the other. Sun et al. [22] argue that classical models, such as the Horn and Schunck [11] can achieve good performance when coupled to modern optimizers. They point out the key elements that contribute to quality of the solution, including image pre-processing, a coarse-to-fine scheme, bicubic interpolation, robust

[⋆] Michael Hornáček and Frederic Besse were funded by Microsoft Research through its European Ph.D. scholarship programme.

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 220–234, 2014.
© Springer International Publishing Switzerland 2014

penalty functions, and median filtering, which they integrate into a new energy formulation. Xu et al. [28] observe that while a large number of optical flow techniques use a multiscale approach, pyramidal schemes can lead to problems in accurately detecting the large motion of fine structures. They propose to combine sparse feature detection with a classic pyramidal scheme to overcome this difficulty. Additionally, they selectively combine color and gradient in the similarity measure on a per pixel basis to improve robustness, and use a Total Variation/L1 (TVL1) optimizer [31]. Similarly, Brox et al. [6] integrate SIFT feature matching [14] into a variational framework to guide the solution towards large displacements.

Another way to define a correspondence is in terms of the similarity of *pixel windows* centered on each image pixel. Immediately, the size of the window becomes an important algorithm parameter: a small window offers little robustness to intensity variations such as those caused by lighting change, differences in camera response, or image noise; a large window can overcome these difficulties but most published work suffers from what we loosely term the ‘fronto-parallel’ (FP) assumption, according to which each point in the window is assumed to undergo the same 2D translation. The robustness of small-window models can be improved by means of priors over motion at neighboring pixels, but first-order priors themselves typically imply the fronto-parallel limitation, second-order priors are expensive to optimize for general energies [27] although efficient schemes exist for some cases [23]. Beyond second order, higher-order priors impose quite severe limitations on the state spaces they can model. In the case of optical flow, the state space is essentially continuous, and certainly any discretization must be very dense.

An alternative strategy to relax the FP assumption is to *overparameterize* the motion field. Previous work in optical flow has considered 3 DoF similarity transformations [3], 6 DoF affine transformations [18], or 6 DoF linearized 3D motion models [18]. In the case of stereo correspondence, the 1 DoF disparity field has been overparameterized in terms of a 3 DoF surface normal and depth field [4,5,13]. With such models, even first order priors can be expressive (e.g., piecewise constant surface normal is equivalent to piecewise constant depth derivatives rather than piecewise constant depth). However, effective optimization of such models has required linearization of brightness constancy [18] or has suffered from local optimality [13]. Recently, however, algorithms based on PatchMatch [2,3] have been applied to 3 DoF (depth+normal) stereo matching [4,5,10] and 6 DoF (3D rigid body motion) RGB-D scene flow [12], and it is to this class of algorithms that ours belongs.

In this paper, we employ an overparameterization not previously applied to the computation of optical flow, assigning a 9 DoF plane-induced homography to each pixel. In addition to relaxing the FP assumption, such a model allows for geometrically meaningful reasoning to be integrated in the optimization, reflecting possible 3D motion in the underlying scene. Vogel et al. [25] recover scene flow over consecutive calibrated stereo pairs by jointly computing a segmentation of a keyframe and assigning to each segment a 9 DoF plane-induced homography, optimized using QPBO [21] over a set of proposal homographies. For optical flow from a pair of images without strictly enforcing epipolar geometry, we show that PatchMatch Belief Propagation (PMBP) of Besse et al. [4] can be adapted to optimize the high-dimensional, non-convex optimization problem of assigning a 9 DoF plane-induced homography to each pixel and that the resulting flow fields compare favorably to the state of the art on a number of datasets. The model parameterizes, at each pixel, a 3D plane undergoing rigid body motion, and can be specialized for piecewise rigid motion, or indeed for a single global rigid motion [24,26].

2 Algorithm

Let (I_1, I_2) be an ordered pair of images depicting a static or moving scene at different points in time and/or from different points of view, and let (G_1, G_2) be the analogous gradient images, each consisting of a total of p pixels. For one of the two views $i \in \{1,2\}$, let $\mathbf{x}_s = (x_s, y_s)^\top$ denote such a pixel, indexed by $s \in \{1, \ldots, p\}$. Let $N(s)$ denote the set of indices of the 4-connected neighbors of $\mathbf{x}_s$ and $W(s)$ the set of indices of pixels in the patch centered on $\mathbf{x}_s$. At every pixel $\mathbf{x}_s$, rather than seek a 2D flow vector, we shall aim to obtain a state vector $\boldsymbol{\theta}_s$ that determines a plane-induced homography $H(\boldsymbol{\theta}_s)$ to explain the motion of the pixels $\mathbf{x}_t, t \in W(s)$. We solve for the flow field by minimizing an energy defined over such state vectors, comprising *data terms* ψ_s and *smoothness terms* ψ_{st}:

$$E(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p) = \sum_{s=1}^{p} \psi_s(\boldsymbol{\theta}_s) + \sum_{s=1}^{p} \sum_{t \in N(s)} \psi_{st}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_t). \tag{1}$$

In the remainder of this section, we proceed first to introduce the parameterization and the data term, and follow by detailing the smoothness term.

2.1 Model and Data Term

Ignoring for the moment the details of the parameterization, let $I_i(\mathbf{x}_s)$ and $G_i(\mathbf{x}_s)$ denote the color and gradient, respectively, at pixel $\mathbf{x}_s$ in view i, $i \in \{1,2\}$. Given a pixel $\mathbf{x}$ in floating point coordinates, we obtain $I_i(\mathbf{x}), G_i(\mathbf{x})$ by interpolation. Let $\tilde{\mathbf{x}} = (x_1, x_2, x_3)^\top \in \mathbb{P}^2$ denote a pixel in projective 2-space, and $\epsilon(\tilde{\mathbf{x}}) = (x_1/x_3, x_2/x_3)^\top \in \mathbb{R}^2$ its analogue in Euclidean 2-space. Let H_s be shorthand for $H(\boldsymbol{\theta}_s)$ and $\mathtt{H}_s$ denote the 3×3 matrix form of H_s, and let $H_s * \mathbf{x} = \epsilon(\mathtt{H}_s(\mathbf{x}^\top, 1)^\top) \in \mathbb{R}^2$ be the pixel obtained by applying the homography H_s to the pixel $\mathbf{x}$. This lends itself to a data term that, at pixel $\mathbf{x}_s$ in view i—which we shall call the *source* view—sums over the pixels of the patch $W(s)$:

$$\psi_s(\boldsymbol{\theta}_s) = \frac{1}{|W(s)|} \sum_{t \in W(s)} w_{st} \cdot \Big((1-\alpha)\big\|I_i(\mathbf{x}_t) - I_j(H_s * \mathbf{x}_t)\big\| + \alpha\big\|G_i(\mathbf{x}_t) - G_j(H_s * \mathbf{x}_t)\big\| \Big), \tag{2}$$

where $j \in \{1,2\}, i \neq j$, indexes the *destination* view, $w_{st} = \exp(-\|I_i(\mathbf{x}_s) - I_i(\mathbf{x}_t)\|/\gamma)$ implements a form of adaptive support weighting [30], and $\alpha \in [0,1]$ controls the relative influence of the color and gradient components of the data term. The data term is scaled by $1/|W(s)|$ in the aim of rendering the strength of the smoothness term in (1) invariant to the patch size.

Casting the standard FP model in these terms, one could define $\boldsymbol{\theta}^{\mathrm{FP}} = (\delta_x, \delta_y)^\top$ to be the 2D flow vector at pixel $\mathbf{x}_s$, and express the homography $H(\boldsymbol{\theta}^{\mathrm{FP}})$ in matrix form as

$$\mathtt{H}(\boldsymbol{\theta}^{\mathrm{FP}}) = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Nir et al. [18] propose a number of further variants of $H(\boldsymbol{\theta})$ including a 6 DoF affine transformation and a 6 DoF linearized 3D motion model. In [20], the fundamental matrix $\mathtt{F}$ is assumed to be known, and homographies consistent with $\mathtt{F}$ are parameterized by three parameters per pixel, yielding essentially an unrectified dense stereo algorithm. The three parameters are related to the 3 DoF parameterization of a scene plane at pixel $\mathbf{x}_s$, as used in [4,5]. We take the parameterization a step further, parameterizing not only a 3D plane at each pixel, but also a 3D rigid body motion transforming the points in the plane.

Let $\mathbf{n}_s$ denote the unit surface normal of a plane in 3D and Z_s the depth of the point of intersection of that plane with the back-projection $\mathbf{p}_s = \mathtt{K}^{-1}(\mathbf{x}_s^\top, 1)^\top$ of the pixel $\mathbf{x}_s$, where $\mathtt{K}$ is the 3×3 camera calibration matrix. The point of intersection is then given by $Z_s\mathbf{p}_s \in \mathbb{R}^3$. Let $\mathtt{R}_s, \mathbf{t}_s$ denote a rigid body motion in 3D. We write our overparameterized motion model $H(\boldsymbol{\theta}_s)$ in matrix form as

$$\mathtt{H}(\boldsymbol{\theta}_s) = \mathtt{K}\left(\mathtt{R}_s + \frac{1}{Z_s\mathbf{n}_s^\top\mathbf{p}_s}\mathbf{t}_s\mathbf{n}_s^\top\right)\mathtt{K}^{-1}, \tag{4}$$

where $\boldsymbol{\theta}_s = (Z_s, \mathbf{n}_s, \mathtt{R}_s, \mathbf{t}_s)^\top$, for a total of 9 DoF. Setting $Z_s\mathbf{n}_s^\top\mathbf{p}_s = -d_s$, we obtain the familiar *homography induced by the plane* [9], with plane $\boldsymbol{\pi}_s = (\mathbf{n}_s^\top, d_s)^\top \in \mathbb{P}^3$. For static scenes undergoing only camera motion, $\mathtt{R}_s, \mathbf{t}_s$ determine the pose of the camera of the destination view, expressed in the camera coordinate frame of the source view. More generally, such a homography lends itself to interpretation as $\mathtt{R}_s, \mathbf{t}_s$ applied to the point obtained by intersecting $\boldsymbol{\pi}_s$ with a pixel back-projection in the source view, and projecting the resulting point into the destination view (cf. Fig. 1), with the pose of both cameras kept identical. On this interpretation, we may reason about scenes undergoing pure camera motion, pure object motion, or joint camera and object motion in the same conceptual framework.

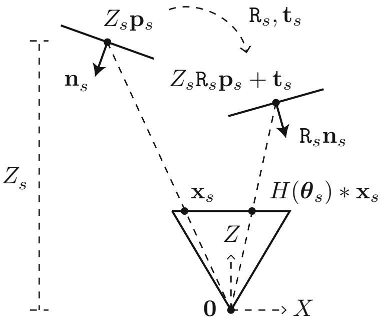

$Z_s\mathbf{p}_s$
$\mathtt{R}_s, \mathbf{t}_s$
$\mathbf{n}_s$
$Z_s\mathtt{R}_s\mathbf{p}_s + \mathbf{t}_s$
$\mathtt{R}_s\mathbf{n}_s$
Z_s
$\mathbf{x}_s$
$H(\boldsymbol{\theta}_s) * \mathbf{x}_s$
Z
$\mathbf{0}$
X

Fig. 1. Depiction of the geometric interpretation of a homography $H(\boldsymbol{\theta}_s)$, $\boldsymbol{\theta}_s = (Z_s, \mathbf{n}_s, \mathtt{R}_s, \mathbf{t}_s)^\top$, assigned to a pixel $\mathbf{x}_s$ as a 3D plane with unit normal $\mathbf{n}_s$ intersecting the back-projection of the pixel $\mathbf{x}_s$ at depth Z_s and undergoing the rigid body motion $\mathtt{R}_s, \mathbf{t}_s$. Applying $H(\boldsymbol{\theta}_s)$ to an arbitrary pixel $\mathbf{x}_t$ has the effect of intersecting the back-projection of $\mathbf{x}_t$ with this plane to obtain a point $\mathbf{P}_t \in \mathbb{R}^3$, transforming $\mathbf{P}_t$ by the motion $\mathtt{R}_s, \mathbf{t}_s$ to obtain $\mathbf{P}'_t = \mathtt{R}_s\mathbf{P}_t + \mathbf{t}_s$, and finally projecting $\mathbf{P}'_t$ back to image space.

Recognizing that a plane whose normal does not point toward the camera is meaningless, and one that is close to orthogonal to the look direction is of no practical use in obtaining matches, we additionally wish to flatly reject such *invalid* states without taking the time to compute the data term in (2). Accordingly, a homography $H(\boldsymbol{\theta}_s)$ is deemed invalid if the source and destination normals $\mathbf{n}_s, \mathtt{R}_s\mathbf{n}_s$ do not both face toward the camera and are not both within 85° of the source and destination look direction vectors, respectively. We additionally deem invalid states that encode negative source or destination depth or states for which $H(\boldsymbol{\theta}_s) * \mathbf{x}_s$ lies outside the destination image.

2.2 Smoothness Term

The role of the smoothness term ψ_{st} is to encourage the action of the homographies parametrized by states $\boldsymbol{\theta}_s, \boldsymbol{\theta}_t$ assigned to neighboring pixels to be similar. One approach to defining a such a smoothness term could be to define distances between the geometric quantities encoded in the state vectors, specifically depth, normal, and rigid body motion. Reasoning directly in terms of the similarity of the parameters of the model would introduce a number of algorithm tuning parameters, as the natural scales of variation of each parameter type are not commensurate. While these could be determined using a training set, a large training set may be required. We instead focus our attention directly on the smoothness of the resulting 2D flow—since it is a smooth 2D flow field that we aim to obtain as output of our algorithm—and introduce a considerably more intuitive smoothness term:

$$\psi_{st}(\boldsymbol{\theta}_s, \boldsymbol{\theta}_t) = \lambda \cdot \min\left(\kappa, \left\|H_s * \mathbf{x}_s - H_t * \mathbf{x}_s\right\| + \left\|H_t * \mathbf{x}_t - H_s * \mathbf{x}_t\right\|\right), \qquad (5)$$

where $\lambda \geq 0$ is a smoothness weight and $\kappa > 0$ is a truncation constant intended to add robustness to large state discontinuities, particularly with object boundaries in mind. This smoothness term has only two parameters (λ and κ) and is in units of pixels.

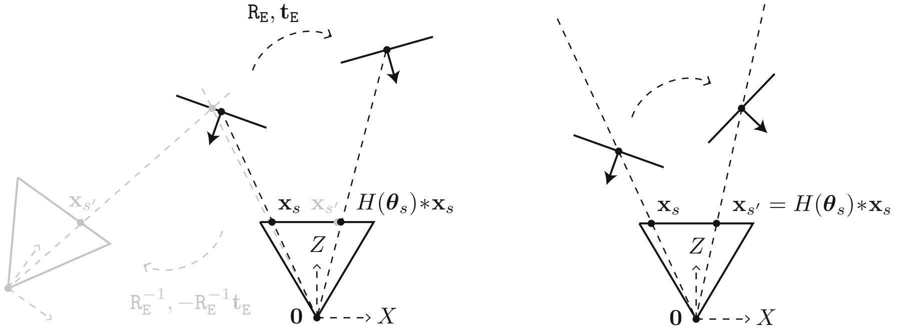
2.3 Energy Minimization

While it may be easy to formulate a realistic energy function, such a function is of little practical use if it cannot be minimized in reasonable time. Minimizing the energy in (1) is a non-convex optimization problem over a high-dimensional, continuous state space. The recently introduced PatchMatch Belief Propagation (PMBP) algorithm of Besse et al. [4] provides an avenue to optimizing over such a state space by leveraging PatchMatch [2,3] for exploiting the underlying spatial coherence of the parameter space by sampling from pixel neighbors (spatial propagation), and belief propagation [29] for the explicit promotion of smoothness.

We adapt PMBP in the aim of assigning to each pixel $\mathbf{x}_s$ an optimal state $\boldsymbol{\theta}_s$, mapping the projectively warped patch centered on $\mathbf{x}_s$ in the source view to its analogue in the destination view. Since our parameterization has a geometric interpretation in terms of rigidly moving planes in 3D, we are able to tailor PMBP to make moves that are sensible in 3D. We begin by (i) initializing the state space in a semi-random manner, making use of knowledge about the scene that we are able to recover from the input image pair (*initialization*). Next, for i iterations, we traverse each pixel $\mathbf{x}_s$ in scanline order, first (ii) attempting to propagate the states assigned to neighbors of $\mathbf{x}_s$ (*spatial propagation*) and then (iii) trying to refine the state vector (*random search*), in each case adopting a candidate state if doing so yields lower disbelief than the current assignment. We do this in both directions (view 1 to view 2, view 2 to view 1) *in parallel* and in opposite traversal orders, and as a last step when visiting $\mathbf{x}_s$ we additionally (iv) attempt to propagate the state at $\mathbf{x}_s$ from the source view to $H(\boldsymbol{\theta}_s) * \mathbf{x}_s$ in the destination, rounded to the nearest integer pixel (*view propagation*); accordingly, by the time a pixel is reached in one view, the most recent match available from the other has already been considered.

Initialization. In order to promote convergence to correct local minima, we constrain our choice of initializing state vectors using knowledge we are able to recover from the input image pair. We estimate the dominant rigid body motion of the scene by feeding pairs of keypoint matches obtained using ASIFT[-3] [17] to the 5 point algorithm [19] with RANSAC [8], giving an essential matrix $\mathtt{E} = [\mathbf{t}_{\mathtt{E}}]_{\times}\mathtt{R}_{\mathtt{E}}$ that we subsequently decompose into a rigid body motion $\mathtt{R}_{\mathtt{E}}, \mathbf{t}_{\mathtt{E}}$ [9]. One might consider iteratively recovering additional dominant rigid body motions by culling inlier matches and re-running the 5 point algorithm with RANSAC on the

[-3] The publicly available ASIFT code carries out a form of epipolar filtering using the Moisan-Stival Optimized Random Sampling Algorithm (ORSA) [16]. We remove this feature in order to obtain all matches recovered by the ASIFT matcher.


$\mathtt{R}_\mathtt{E}, \mathbf{t}_\mathtt{E}$
$\mathbf{x}_{s'}$
$\mathbf{x}_s$
$\mathbf{x}_{s'}$
$H(\boldsymbol{\theta}_s){*}\mathbf{x}_s$
Z
$\mathtt{R}_\mathtt{E}^{-1}, -\mathtt{R}_\mathtt{E}^{-1}\mathbf{t}_\mathtt{E}$
$\mathbf{0}$
X
$\mathbf{x}_s$
$\mathbf{x}_{s'} = H(\boldsymbol{\theta}_s){*}\mathbf{x}_s$
Z
$\mathbf{0}$
X

(a) Initialization from inlier matches. **(b)** Initialization from general matches.

Fig. 2. (a) Initialization from ASIFT match pairs $(\mathbf{x}_s, \mathbf{x}_{s'})$ that are inliers of a recovered dominant rigid body motion $\mathtt{R}_\mathtt{E}, \mathbf{t}_\mathtt{E}$, with depth Z_s determined by triangulation and $\mathbf{n}_s$ as the only free parameter. (b) Initialization from general ASIFT match pairs $(\mathbf{x}_s, \mathbf{x}_{s'})$, constrained in that $\mathbf{x}_{s'} = H(\boldsymbol{\theta}_s) * \mathbf{x}_s$; an alternative expression of this constraint is the requirement that $Z_s \mathtt{R}_s \mathbf{p}_s + \mathbf{t}_s$ project exactly to the pixel $\mathbf{x}_{s'}$.

matches that remain, or consider alternative rigid motion segmentation techniques [7]. We triangulate the ASIFT matches that are inliers of the recovered dominant motion, giving seed points for which only the plane normal $\mathbf{n}_s$ remains a free parameter (cf. Fig. 2a). Since we wish to allow deviation from recovered dominant motions yet would like to leverage all of the available ASIFT matches, we additionally use the full set of ASIFT match pairs $(\mathbf{x}_s, \mathbf{x}_{s'})$ for seeding by estimating, for each pair, a tailored rigid body motion constrained by the requirement that $\mathbf{x}_{s'} = H(\boldsymbol{\theta}_s) * \mathbf{x}_s$ (cf. Fig. 2b), with depth Z_s in addition to normal $\mathbf{n}_s$ as free parameters. At pixels where more than one such seed is available, we choose one at random. For unseeded pixels, we set $\mathtt{R}_s, \mathbf{t}_s$ to one of the recovered dominant motions, with depth Z_s and normal $\mathbf{n}_s$ again free.

Spatial Propagation. In the usual manner of PatchMatch [2,3,4], we traverse the pixels of the source image in scanline order and consider, at the current pixel $\mathbf{x}_s$, the subset of states $\{\boldsymbol{\theta}_t \mid t \in N(s)\}$ assigned to the 4-connected neighbors of $\mathbf{x}_s$ that have already been visited in the iteration, and adopt such a state $\boldsymbol{\theta}_t$ if doing so gives lower disbelief than the current assignment. Note that owing to our parameterization, adopting the state $\boldsymbol{\theta}_t = (Z_t, \mathbf{n}_t, \mathtt{R}_t, \mathbf{t}_t)^\top$ at pixel $\mathbf{x}_s$ calls for recomputing the depth by intersecting the plane $\boldsymbol{\pi}_t$ with the back-projection of $\mathbf{x}_s$; the remaining components of the state vector $\boldsymbol{\theta}_t$ are simply copied.

Random Search. We perturb, at random, either depth Z_s and normal $\mathbf{n}_s$ or the rigid body motion $R_s, \mathbf{t}_s$ of the state vector θ_s currently assigned to the pixel $\mathbf{x}_s$. When $\mathtt{R}_s, \mathbf{t}_s$ are locked, we are effectively carrying out stereo matching. When $Z_s, \mathbf{n}_s$ are locked, we perturb the translational component of the motion with the effect of sampling within a 3D radius around $Z_s \mathtt{R}_s \mathbf{p}_s + \mathbf{t}_s$; perturbation of

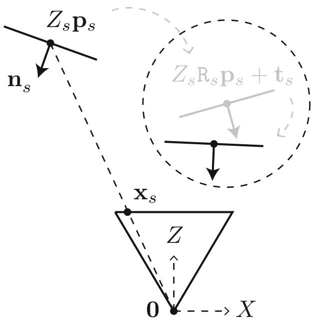


Fig. 3. Refinement of the rigid motion $\mathtt{R}_s, \mathbf{t}_s$ for plane parameters $Z_s, \mathbf{n}_s$ fixed. Perturbation of the translational component $\mathbf{t}_s$ is carried out with the effect of applying a translation to the current $\mathbf{P}'_s = Z_s\mathtt{R}_s\mathbf{p}_s + \mathbf{t}_s$ within a radius of $\mathbf{P}'_s$ in 3D (depicted by the dashed circle). Perturbation of the rotational component $\mathtt{R}_s$ serves effectively to rotate the transformed plane around the current $\mathbf{P}'_s$.

the rotational component serves effectively to change the normal of the transformed plane (cf. Fig. 3). We carry out several such perturbations of the four components of the assigned state vector, reducing the search range with every try. We adopt a proposed perturbation if doing so gives lesser disbelief than the current assignment.

If $\mathtt{R}_s, \mathbf{t}_s$ are reasonable and if at least parts of the reconstructed depth map are already plausible, a geometrically sensible move to promote convergence to correct local minima is to attempt to refine $\mathbf{n}_s, Z_s$ by fitting a plane to the already computed minimum disbelief recovered 3D points $\{Z_t\mathbf{p}_t \mid t \in W(s), \boldsymbol{\theta}_t = (Z_t, \mathbf{n}_t, \mathtt{R}_t, \mathbf{t}_t)^\top\}$, using RANSAC. The candidate normal is simply the normal vector—constrained to point toward the camera—of this plane, and the candidate depth is obtained by intersecting the plane with the back-projection of $\mathbf{x}_s$. We carry out such a plane fit as the first step in random search, and follow with the perturbations described above.

View Propagation. Most similarly to [12], which in turn builds upon [4,5], as a last step when visiting a pixel $\mathbf{x}_s$ and given its assigned state vector $\boldsymbol{\theta}_s = (Z_s, \mathbf{n}_s, \mathtt{R}_s, \mathbf{t}_s)^\top$, we propose the inverted state $\boldsymbol{\theta}'_s = (Z'_s, \mathbf{n}'_s, \mathtt{R}'_s, \mathbf{t}'_s)^\top$ in the destination view. We compute $\boldsymbol{\theta}'_s$ by $\mathbf{n}'_s = \mathtt{R}_s\mathbf{n}_s$, $\mathtt{R}'_s = \mathtt{R}_s^{-1}$, $\mathbf{t}'_s = -\mathtt{R}_s^{-1}\mathbf{t}$; the depth Z'_s is obtained by intersecting the transformed plane with the back-projection of $Z_s\mathtt{R}_s\mathbf{p}_s + \mathbf{t}_s$ projected to the nearest integer pixel, which is where in the destination view we then evaluate $\boldsymbol{\theta}'_s$. Geometrically, this amounts to considering the inverse rigid body motion applied to the transformed plane. Since we carry out our algorithm on both views in parallel and in opposite traversal orders, the most recent corresponding match available from the destination view has thus already been considered by the time $\mathbf{x}_s$ is reached.

2.4 Post-processing

In areas of the scene that are occluded in one of the two views, subject to the aperture problem, or poorly textured, our algorithm is likely to assign states

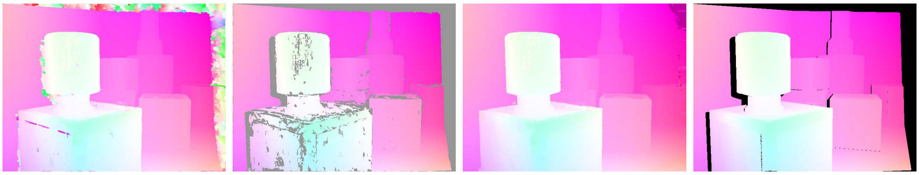


Fig. 4. Effect of our post processing on the Crates1Htxtr2 data set. Only the pixels that fail the consistency check (indicated in gray) undergo post-processing.

that do not correspond to the correct flow (cf. Fig. 4). If flow is computed in both directions, we can identify inconsistent state assignments by running a consistency check over 'forward' and 'backward' flow, labelling as inconsistent each pixel $\mathbf{x}_s$ that fails the following condition:

$$\left\| \mathbf{x}_s - H(\boldsymbol{\theta}_s^{\mathrm{B}}) * \big(H(\boldsymbol{\theta}_s^{\mathrm{F}}) * \mathbf{x}_s\big) \right\| \leq 1, \tag{6}$$

where $\boldsymbol{\theta}_s^{\mathrm{F}}$ determines the forward flow assigned in the source view to pixel $\mathbf{x}_s$, and $\boldsymbol{\theta}_s^{\mathrm{B}}$ the backward flow assigned in the destination view to the pixel $\boldsymbol{\theta}_s^{\mathrm{F}} * \mathbf{x}_s$ rounded to the nearest integer coordinates. This generates a pixel mask that identifies pixels that subsequently undergo post-processing. For each $\mathbf{x}_s$ that failed the check, we first consider the pixels in a window around $\mathbf{x}_s$ that passed, adopting the homography of the pixel that is closest in appearance. Next, for pixels $\mathbf{x}_s$ that still fail the check, we seek the nearest pixels above and below $\mathbf{x}_s$ that passed, and adopt the homography of the pixel closest in appearance. Finally, we proceed similarly for left and right.

3 Evaluation

We tested our method on the UCL optical flow data set [15] and on a subset of the Middlebury optical flow benchmark [1] for which ground truth flow was available. Accordingly, we considered data sets exhibiting flow at small and large displacements (we set the threshold between the two at 25 pixels) and undergoing rigid, piecewise rigid, and non-rigid motions. A comparison over end point error (EPE) is provided in Table 1 with respect to four competing methods. We ran our algorithm on all data sets in the table with a patch size of 21 × 21 for three iterations on a single particle. As in [4,5], we set the weight α that balances the influence of gradient over color in (2) to 0.9, and γ in the adaptive support weighting to 10. The truncation constant κ of the smoothness term in (5) was set to 1 in all our experiments. Only a single dominant rigid body motion was recovered per data set, in the manner described in Sec. 2.3. Minimum depth was fixed to 0; maximum depth per view was set to the maximum depth of triangulated matches that were inliers of the dominant motion. In the random search stage, maximum allowable deviation from the current rigid body motion was set to 0.01 for both the rotational (expressed in terms of quaternions) and translational components of the motion. Analogously to [4,5], we set maximum flow per dataset. Camera calibration matrix K was fixed such that the focal length was 700 pixels and the principal point was in the image center.

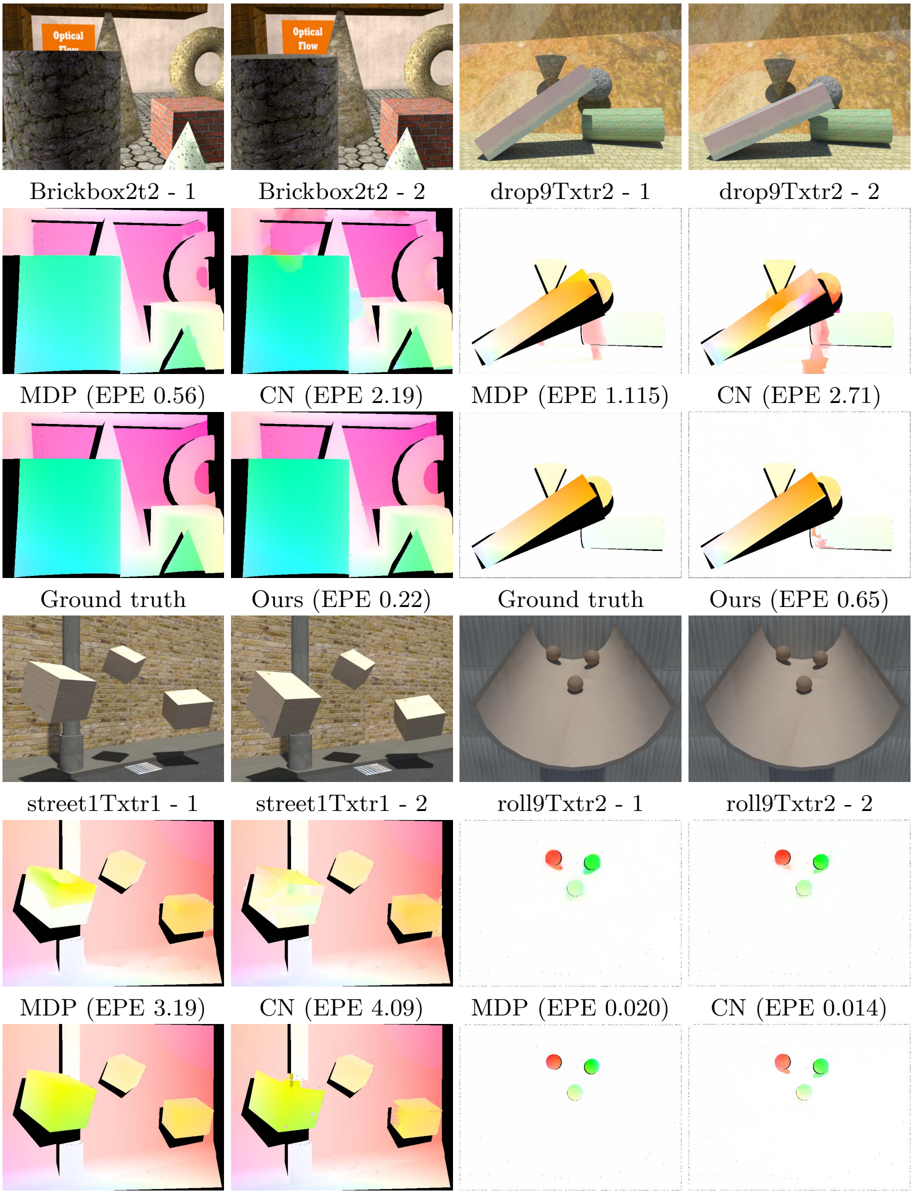
	TV	LD	CN	MDP	Ours$_{\lambda=0.005}$	Ours$_{\lambda=0}$	Ours$_{\lambda=0.01}$
UCL Lg. Displ.							
Crates1	3.46	3.10	3.15	**1.65**	2.37	2.62	2.9
Crates2	4.62	2.51	10.4	**1.35**	1.71	1.84	1.73
Mayan1	2.33	5.56	1.71	0.48	**0.16**	0.17	0.18
Robot	2.34	1.21	1.53	**0.7**	1.85	2.14	1.96
Crates1Htxtr2	1.11	0.54	1.64	**0.28**	0.29	0.39	0.3
Crates2Htxtr1	3.13	0.81	8.8	**0.37**	0.47	0.45	0.64
Brickbox1t1	1.09	2.6	0.22	0.2	**0.15**	0.16	0.15
Brickbox2t2	7.48	3.51	2.19	0.56	**0.22**	0.2	0.22
GrassSky0	2.1	1.04	1.3	0.47	**0.27**	0.3	0.27
GrassSky9	0.72	0.51	0.27	0.29	**0.25**	0.34	0.26
blow19Txtr2†	0.53	0.32	**0.19**	0.26	0.22	0.23	0.27
drop9Txtr2†	5.2	4.37	2.71	1.15	**0.65**	0.75	0.86
street1Txtr1†	3.65	2.66	4.09	3.19	**0.92**	1.72	1.45

	TV	LD	CN	MDP	Ours$_{\lambda=0.005}$	Ours$_{\lambda=0}$	Ours$_{\lambda=0.01}$
UCL Sm. Displ.							
Mayan2	0.44	0.35	0.21	0.23	**0.17**	0.19	0.18
YosemiteSun†	0.31	**0.18**	0.23	3.79	0.33	0.35	0.38
GroveSun	0.58	0.48	**0.23**	0.43	0.24	0.24	0.23
Sponza1	1.01	**0.91**	1.1	1.08	2.75	2.84	2.8
Sponza2	0.53	**0.48**	1.6	1.77	2.61	2.58	2.61
TxtRMovement	3.17	0.36	**0.13**	0.19	1.71	1.7	1.72
TxtLMovement	1.52	0.6	**0.12**	0.23	1.73	1.76	1.76
blow1Txtr1†	0.09	0.08	**0.03**	0.05	0.04	0.04	0.04
drop1Txtr1†	0.12	0.08	0.05	0.06	**0.04**	0.04	0.04
roll1Txtr1†	0.004	**0.002**	**0.002**	**0.002**	**0.002**	0.002	0.002
roll9Txtr2†	0.04	0.02	**0.01**	0.02	**0.01**	0.01	0.01
Middlebury							
Dimetrodon†	0.211	0.117	**0.115**	0.153	0.169	0.174	0.17
Grove2	0.220	0.149	**0.091**	0.15	0.184	0.187	0.3
Grove3	0.745	0.657	**0.438**	0.53	0.517	0.455	0.97
Hydrangea†	0.196	0.178	**0.154**	0.164	0.222	0.207	0.234
RubberWhale†	0.135	0.120	**0.077**	0.09	0.114	0.12	0.125
Urban2	0.506	0.334	**0.207**	0.32	0.3	0.312	0.29
Urban3	1.132	0.600	**0.377**	0.42	0.905	1.27	1.03
Venus	0.408	0.433	**0.229**	0.28	0.342	0.342	0.434

Table 1. End point error (EPE) comparison. TV = A Duality Based Approach for Realtime TV-L1 Optical Flow [31]. LD = Large Displacement Optical Flow [6]. CN = Secrets of Optical Flow [22]. MDP = Motion Detail Preserving Optical Flow [28]. Cell colors indicate ranking among the five methods, from best to worst: green, light green, yellow, orange, red. Gray cells are shown for comparison but are not included in the ranking. † indicates that the scene is non-static.

Our method performs particularly well on the large displacement cases of the UCL dataset, and produces reasonable results for smaller displacements. Quantitative results show that our technique outperforms all four other methods in ca. 1/3 of the data sets (ca. 1/2 of the cases for large motion), while the end point error is lower than that of TV and LD in most of the cases. The color scheme used in Table 1 indicates that our approach is the one that is most frequently ranked in the first two positions (ca. 2/3 of the cases), when compared to the other four techniques. A visual comparison for four data sets is given in Fig. 5. The effect of the smoothness term can be seen in Fig. 6, where we compare the resulting 2D flow for our algorithm with $\lambda = 0$ (no smoothness) and $\lambda = 0.005$ on the Middlebury Dimetrodon data set. Additionally, we give the EPE results for $\lambda = 0$ and $\lambda = 0.01$ for all data sets in Table 1.

(Piecewise) Unrectified Stereo. For scenes undergoing only a single dominant rigid body motion, one could run our algorithm with no deviation allowed from the recovered dominant rigid body motion $\mathtt{R}_\mathtt{E}, \mathbf{t}_\mathtt{E}$. We show precisely such a reconstruction for the Brickbox2t2 data set in Fig. 7, providing a coloring of the recovered normals, the depth map, and a colored point cloud rendered at a novel view. Locking the motion reduces our algorithm to an unrectified stereo matcher with slanted support windows, most closely akin to [4].



Brickbox2t2 - 1 Brickbox2t2 - 2 drop9Txtr2 - 1 drop9Txtr2 - 2

MDP (EPE 0.56) CN (EPE 2.19) MDP (EPE 1.115) CN (EPE 2.71)

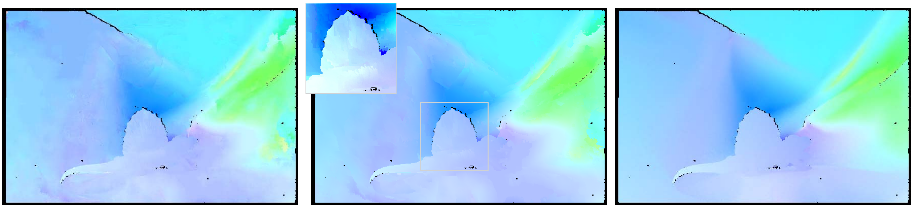
Ground truth Ours (EPE 0.22) Ground truth Ours (EPE 0.65)

street1Txtr1 - 1 street1Txtr1 - 2 roll9Txtr2 - 1 roll9Txtr2 - 2

MDP (EPE 3.19) CN (EPE 4.09) MDP (EPE 0.020) CN (EPE 0.014)

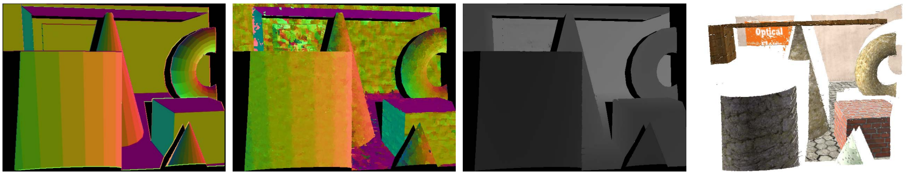
Ground truth Ours (EPE 0.92) Ground truth Ours (EPE 0.01)

Fig. 5. Optical flow colorings for a subset of the UCL optical flow data set. EPE = End Point Error. CN = Secrets of Optical Flow [22]. MDP = Motion Detail Preserving Optical Flow [28]. Results correspond to Table 1.



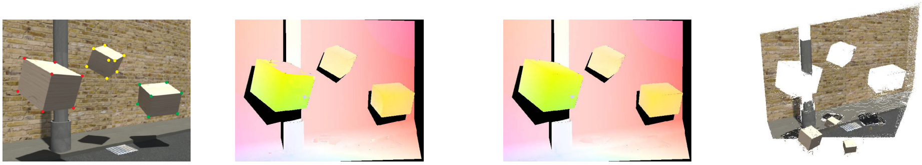
$\lambda = 0$ (EPE: 0.175) $\lambda = 0.005$ (EPE: 0.169) Ground truth

Fig. 6. The effect of the smoothness term on the Dimetrodon data set for $\lambda = 0$ (no smoothness) and $\lambda = 0.005$. Inlay shown with contrast stretch; results best viewed zoomed in. Flow coloring and EPE without post-processing.



GT normals Recovered normals Recovered depth Colored point cloud

Fig. 7. Restriction to the recovered dominant rigid body motion $\mathtt{R}_\mathtt{E}, \mathbf{t}_\mathtt{E}$ for the Brickbox2t2 data set. Estimation of the plane normals and depth on a static scene, and rendering as a colored point cloud.



Manual seeds Unrestricted (EPE 0.73) Restricted (EPE: 0.468) Restricted

Fig. 8. Result obtained on the street1Txtr1 data set by seeding with the dominant motion obtained by the 5 point algorithm with RANSAC on all ASIFT matches and on the three additional sets of manually provided matches (indicated in red, yellow, and green), giving four motions in total. Results shown for deviation allowed from those four motions, and for no deviation allowed. Otherwise, we used the same parameter settings to compute our results as in Table 1.

In order to give an impression of the limits of the approach, we recover the dominant rigid body motion on the street1Txtr1 data set in the manner described in Sec. 2.3 and obtain motions on the three independently moving cubes by manually supplying correspondences to the 5 point algorithm using RANSAC (cf. Fig. 8). We show the result for allowing deviations from those four motions, and for allowing no deviation. We additionally show the resulting point cloud where no deviation is allowed. Note that the three cubes are not reconstructed with commensurate size; this is a consequence of each piecewise reconstruction being individually up to a scale ambiguity.

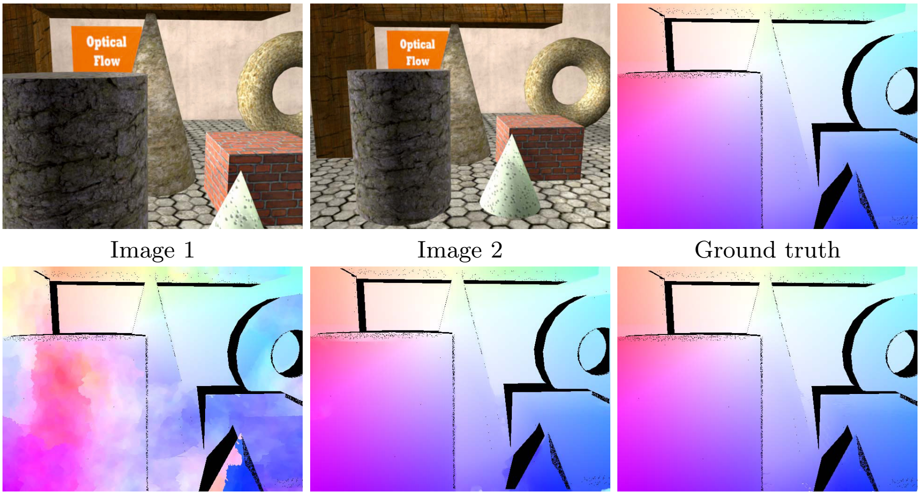


Image 1 Image 2 Ground truth

CN (EPE: 38.25) MDP (EPE: 1.02) Ours (EPE: 0.42)

Fig. 9. Result on a challenging case with large displacement camera zoom, causing a radial flow pattern. Note that this sequence is not part of the published UCL optical flow data set. We used the same parameter settings to compute our results as in Table 1.

Radial Flow. Certain types of camera motions can be difficult to handle for flow methods that use a 2D parametrisation. For instance, camera zoom induces a radial flow pattern around the viewing direction, which conflicts with a smoothness assumption that promotes neighboring flow vectors to be similar. However, our approach is flexible enough to recover the homographies induced by this motion, as illustrated in Fig. 9.

Limitations. We kept the patch size identical across all our experiments, regardless of image size or scale. As in patch-based stereo techniques, our approach is sensitive to the aperture problem, and more generally to poorly textured surfaces. It is this problem of inadequate match discriminability that accounts for the comparatively poor performance of our algorithm for the Robot, Sponza1, Sponza2, TxtRMovement, and TxtLMovement data sets. An obvious way to alleviate this problem where applicable is to set the patch size appropriately. A direction for future work could be to develop a smoothness term that promotes not only smoothness of the 2D flow, but explicitly exploits the geometric interpretation of the paramterization to promote similarity of the 9 DoF states themselves.

4 Conclusion

We have presented a new optical flow technique that uses a simple and geometrically motivated model and exploits that model to carry out the optimization in a manner that makes geometrically reasonable moves. While the model lives in a high-dimensional space that would prove challenging to optimize using conventional methods, we show PMBP to be well suited for the task. We obtain a 2D flow that compares favorably to other state-of-the-art techniques and manage to handle both small and large displacements. Our smoothness term helps promote smoothness of the obtained 2D flow fields. A side effect of our approach is that—provided rigid body motions are reasonable—depth can be directly extracted from the parameterization, which can be used to construct a point cloud and flowed to intermediate time steps.

References

1. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. Intl. J. of Comp. Vis. (2011)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: PatchMatch: a randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (2009)
3. Barnes, C., Shechtman, E., Goldman, D.B., Finkelstein, A.: The generalized patch-Match correspondence algorithm. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 29–43. Springer, Heidelberg (2010)
4. Besse, F., Rother, C., Fitzgibbon, A., Kautz, J.: PMBP: PatchMatch belief propagation for correspondence field estimation. In: Proc. BMVC (2012)
5. Bleyer, M., Rhemann, C., Rother, C.: PatchMatch stereo-Stereo matching with slanted support windows. In: Proc. BMVC (2011)
6. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: Proc. CVPR (2009)
7. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. Intl. J. of Comp. Vis. (2012)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM (1981)
9. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, vol. 2. Cambridge University Press (2000)
10. Heise, P., Klose, S., Jensen, B., Knoll, A.: PM-Huber: PatchMatch with Huber regularization for stereo matching. In: Proc. CVPR (2013)
11. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence (1981)
12. Hornáček, M., Fitzgibbon, A., Rother, C.: SphereFlow: 6 DoF scene flow from RGB-D pairs. In: Proc. CVPR (June 2014)
13. Li, G., Zucker, S.W.: Surface geometric constraints for stereo in belief propagation. In: Proc. CVPR (2006)
14. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV (1999)
15. Mac Aodha, O., Humayun, A., Pollefeys, M., Brostow, G.: Learning a confidence measure for optical flow. IEEE T-PAMI (2012)
16. Moisan, L., Stival, B.: A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. Intl. J. of Comp. Vis. (2004)
17. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences (2009)

18. Nir, T., Bruckstein, A., Kimmel, R.: Over-parameterized variational optical flow. Intl. J. of Comp. Vis. (2008)
19. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE T-PAMI (2004)
20. Rosman, G., Shem-Tov, S., Bitton, D., Nir, T., Adiv, G., Kimmel, R., Feuer, A., Bruckstein, A.: Over-parameterized optical flow using a stereoscopic constraint. Scale Space and Variational Methods in Computer Vision (2012)
21. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary MRFs via extended roof duality. In: Proc. CVPR (2007)
22. Sun, D., Roth, S., Black, M.: Secrets of optical flow estimation and their principles. In: Proc. CVPR (2010)
23. Trobin, W., Pock, T., Cremers, D., Bischof, H.: An unbiased second-order prior for high-accuracy motion estimation. Pattern Recognition (2008)
24. Valgaerts, L., Bruhn, A., Weickert, J.: A variational model for the joint recovery of the fundamental matrix and the optical flow. Pattern Recognition (2008)
25. Vogel, C., Schindler, K., Roth, S.: Piecewise rigid scene flow. In: Proc. ICCV (2013)
26. Wedel, A., Pock, T., Braun, J., Franke, U., Cremers, D.: Duality TV-L1 flow with fundamental matrix prior. Image and Vision Computing (2008)
27. Woodford, O., Torr, P., Reid, I., Fitzgibbon, A.: Global stereo reconstruction under second-order smoothness priors. IEEE T-PAMI (2009)
28. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE T-PAMI (2012)
29. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Generalized belief propagation. In: NIPS (2000)
30. Yoon, K., Kweon, I.: Adaptive support-weight approach for correspondence search. IEEE T-PAMI (2006)
31. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L1 optical flow. Pattern Recognition (2007)