TriangleFlow: Optical Flow with Triangulation-Based Higher-Order Likelihoods

Ben Glocker[1], T. Hauke Heibel[1], Nassir Navab[1],
Pushmeet Kohli[2], and Carsten Rother[2]

[1] Computer Aided Medical Procedures (CAMP),
Technische Universität München, Germany
{glocker,heibel,navab}@in.tum.de
[2] Microsoft Research, Cambridge, UK
{pkohli,carrot}@microsoft.com

Abstract. We use a simple yet powerful higher-order conditional random field (CRF) to model optical flow. It consists of a standard photo-consistency cost and a prior on affine motions both modeled in terms of higher-order potential functions. Reasoning jointly over a large set of unknown variables provides more reliable motion estimates and a robust matching criterion. One of the main contributions is that unlike previous region-based methods, we omit the assumption of constant flow. Instead, we consider local affine warps whose likelihood energy can be computed exactly without approximations. This results in a tractable, so-called, higher-order likelihood function. We realize this idea by employing triangulation meshes which immensely reduce the complexity of the problem. Optimization is performed by hierarchical fusion moves and an adaptive mesh refinement strategy. Experiments show that we achieve high-quality motion fields on several data sets including the Middlebury optical flow database.

1 Introduction

Currently most methods for optical flow estimation can be roughly divided into two groups: (i) variational methods based on the pioneering work of Horn and Schunck [1], and (ii) discrete methods utilizing combinatorial optimization such as graph-cuts [2]. Both approaches have their advantages and disadvantages. While variational methods often yield very high accuracy, these methods depend on rather local image properties and may also suffer from local minima during optimization of the cost function. Combinatorial optimization is often able to recover strong minima but only with respect to a rather sparse discretization of the search space. Recently, methods have been proposed [3,4] which successfully combine both worlds towards discrete-continuous optimization which is able to avoid local minima and obtain highly accurate (continuous) flow estimates at the same time. A rather comprehensive overview and comparison of latest optical flow methods can be found in [5] and on the website of the Middlebury optical flow database[1].

[1] http://vision.middlebury.edu/flow/

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 272–285, 2010.

Still, a major limitation of existing algorithms is in the definition of the likelihood (or data) term within the energy formulation. Often, a matching criterion is defined pixel-wise for instance using squared differences on the intensities. In general, such a formulation yields an ill-posed problem since two-dimensional flow vectors have to be recovered from a one-dimensional signal (aperture problem). Ambiguities may arise for matching individual pixels independently. Here, regularization plays an important role to render the problem well-posed such that the optimization yields meaningful solutions.

In contrast, region-based approaches [6,7] use local image patches to estimate point correspondences. Here, a matching criterion such as the correlation coefficient (CC) is evaluated on the whole patch centered at a point for which the motion is to be determined. The distribution of such points can be dense or sparse (by employing a parameterization of the motion field) [8]. Region-based approaches yield a more robust definition of the likelihood compared to pixel-wise methods [9], but often introduce a rough approximation. In fact, in most approaches it is assumed that all pixels within the patch move with constant flow. However, except for pure translation within the patch, the assumption of constant flow does not hold.

One may claim that an optimal definition of the likelihood should be (i) *robust and reliable*, by considering a larger set of unknown variables simultaneously and (ii) *precise and tractable* by modeling the various motions for the set of variables beyond the assumption of constant flow. This leads us to our main contribution in this paper, which we call *higher-order likelihoods*. In the following, we will introduce the concept of higher-order likelihoods and their corresponding energy in a conditional random field (CRF). We demonstrate how triangulation meshes perfectly support our concept. The effectiveness of our approach is evaluated on several datasets including the Middlebury optical flow database. We also revisit the concept of motion layers [10] which, when integrated in our framework, enables us to handle occlusions in a natural way in form of overlapping meshes. We conclude our paper by a discussion on future work.

1.1 Related Work

Conditional random fields are ubiquitous in computer vision. Their success can be certainly attributed in large parts to the existence of powerful optimization methods which have been developed in the last decade. The most commonly used models in low-level vision applications are first-order CRFs[2], which contain cliques of size up to two. Here, the *unary potentials* play the role of the likelihood term evaluating how well a certain label fits to a variable w.r.t. to the observation, independently of all other variables. The *pairwise potentials* are then used to enforce smoothness by penalizing deviations of labelings between two neighboring variables. These models are quite intuitive due to their natural relationship to the image grid itself. Additionally, first-order models are attractive due to efficient optimization methods, which often guarantee to find the global optimum.

[2] Note that an n-th order CRF contains cliques of size up to $n + 1$.

Despite the popularity of first-order models, their modeling capabilities are very limited. As already mentioned, a likelihood term based on unaries is either not very reliable or rough approximations have to be used as in previous region-based methods. In some works (e.g. in [11,12,13]), the pairwise terms are considered for the likelihood in order to model a conditional data-dependency on a pair of variables which yields a more appropriate model for the problem at hand.

Recent advances in CRF optimization allow the use of higher-order potentials in an efficient and principled manner [14,15,16]. A combination of fusion moves [17,18], reduction techniques [19], and the QPBO algorithm [20,21] allows to use a second-order model in stereo [22], while a similar model is used for motion in [23] employing belief propagation. Both works use a second-order prior defined on triple-cliques to enforce smoothness based on second derivatives of the disparity/motion field. Still, only unary terms are used for the likelihood.

Recently, many techniques have been developed for larger cliques of up to several hundred variables, e.g. [15,24] just to mention a few. In order to deal with such large cliques in a tractable way , they must exhibit some internal structure. For instance in [15] it is assumed that only a few (important) label-configurations have a low energy and all remaining configurations a constant (high) cost.

In the following, we will introduce our concept of higher-order likelihoods for the task of optical flow. We will derive a likelihood term based on triple-cliques which models the costs of local affine motions exactly without approximations. Additionally, we propose two novel regularization terms, the first one being also based on triple-cliques, and the second one based on quadruple-cliques.

2 Concept of Higher-Order Likelihoods

Consider a set V of variables $i, ..., N$. In optical flow, the variables correspond to pixels and we seek for optimal assignments d_i[3] corresponding to two-dimensional flow vectors. Additionally, we introduce the power set C containing all possible cliques (subsets) c of variables. We define the cost for a *labeling* $\mathbf{d}$ (i.e. every variable is assigned a value d_i) in terms of a general CRF energy as

$$E(\mathbf{d}|\theta) = \sum_{c \in C} \psi_c(\mathbf{d}_c|\theta) \quad . \tag{1}$$

The clique potential functions ψ_c evaluate the cost for assigning a sub-labeling $\mathbf{d}_c$ to a clique c conditioned on the observation θ (the image data). In first-order models, the energy would then be simply the sum of unary potentials $\psi_i(x_i|\theta)$ plus the sum of pairwise potentials $\psi_{ij}(d_i, d_j|\theta)$. For simplicity, in the following we will neglect θ in the potential functions.

[3] Depending on the context we will treat $i, j, ...$ as random variables and as 2D coordinates. Similarly, we treat labels $d_i, d_j, ...$ also as 2D motion vectors.

Theoretically, reasoning jointly over all variables would be the best approach for finding an optimal labeling. The energy would simply consist of one higher-order potential for a clique containing all variables. Obviously, even for a small number of variables this approach is doomed in practice regarding the computational complexity. A compromise has to be found between the clique size and the tractability of the problem.

Let us concentrate on the problem of optical flow. Determining the flow vector of individual pixels is clearly not well defined due to the aperture problem mentioned earlier. In contrast, solving for the flow for a group of pixels might be more reliable. Assume we are seeking for the optimal flow vectors within a discretized search space L (a set of labels). Then, for a clique of K pixels the solution space for the labeling problem has the cardinality $|L|^K$. Evaluating all of the potential labelings is infeasible. We discuss two alternative solutions to this dilemma. We realize one of these solutions in our practical system, which we discuss in detail in Sec. 2.1.

Let us first consider the alternative solution, which we only discuss theoretically. It is based on the recent work [15], where higher-order cliques are modeled by sparse higher-order representations. Only a few labelings have assigned the correct higher-order cost and all other remaining labelings are assigned a constant (high) cost, which approximates their true cost. The key question is now which labelings should be modeled? Note that there is actually only one labeling, i.e. the *maximum a posteriori* (MAP) labeling $\hat{\mathbf{d}}$, which has to be modeled. This is the labeling which corresponds to the global optimum of the CRF energy, which is obviously unknown. One approach is to design a data-driven prediction function which has the observation as input and possible labelings as output. Also, an iterative optimization procedure can be envisioned, where the higher-order terms, which only approximate the current MAP labeling by a constant cost, are redefined and thus improve the modeling of the MAP labeling in the next iteration. However, such an approach might be computationally very expensive. In this paper, we present a simple yet powerful model overcoming this limitation by exploiting inherent properties of optical flow.

2.1 Reduction of Complexity Using Triangulations

Optical flow estimation consists of recovering the apparent motion from two dimensional images capturing a scene of three dimensional objects moving over time. We make two observations: (i) often the scene contains mainly solid objects, which might translate, rotate, and/or scale from one image to another, (ii) the motion of non-solid objects (such as textiles) can be sufficiently represented by several local affine motions. These observations are consistent with other approaches previously proposed for optical flow [25,26,27].

If we restrict the set of labelings to the ones representing affine motions only, we already achieve an immense reduction of complexity. An affine motion in 2D is fully defined by three two-dimensional points (i.e. six degrees of freedom). So, estimating an affine motion from $K(>3)$ pixels is an over-determined problem which allows further simplifications. Additional reduction of complexity can be

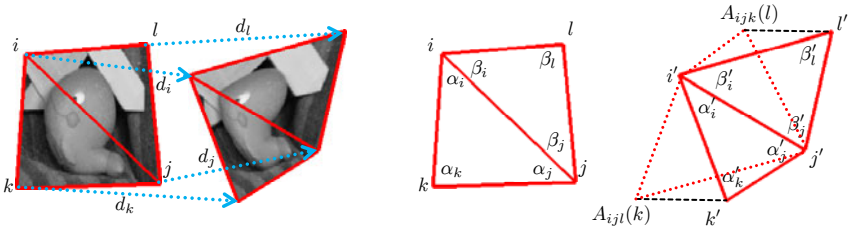


Fig. 1. Left: the triangles (ijk) and (ijl) represent higher-order likelihoods and define local affine warps when labels (d_i, d_j, d_k, d_l) are assigned to the triangle points. Right: illustration of the two different regularization terms. The ADP penalizes changes between initial angles (α, β) and angles (α', β'). The NAMP determines how well the warp of one triangle describes the warp of the other one by computing the (normalized) distance between the warped points k', l' and their locations $A_{ijl}(k), A_{ijk}(l)$ if warped by the neighboring triangle.

achieved by a parameterization of the cliques motion using a simple geometrical transformation model in terms of triangulation. A triangle in 2D space defines an affine warp. We propose to represent a clique of pixels by a single triangle. Then, the task becomes to find the optimal displacements of the triangle points, instead of seeking for individual displacements for each pixel. Let us now derive the energy for this model.

2.2 Likelihood Term

First, we need to define a matching criterion. In this work, we consider the correlation coefficient (CC). For two sets of measurements X and Y, the CC is defined as

$$\mathrm{CC}(X,Y) = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} = \frac{\mathrm{cov}(X,Y)}{\sigma_x \sigma_y} \quad , \tag{2}$$

where $\bar{x}$ and $\bar{y}$ are the two means and σ_x and σ_y the standard deviations. The CC takes values from $[-1, 1]$, where 1 indicates a perfect linear relationship, 0 indicates no linear relationship, and -1 an inverse linear relationship. In order to use the CC score within an energy minimization, we modify the original term into $\mathrm{CC}' = (1 - \mathrm{CC})$ taking values from $[0, 2]$.

Second, we formalize the local affine motion model based on a triangulation mesh. Assume that a set of triangles covering the image domain is given. We can define a *local affine warp* T_{ijk} of a point $p = (x, y)^\top$ lying in a triangle (ijk) as the sum of the products of the barycentric coordinates $(\omega_i, \omega_j, \omega_k)$ of p and the three displacement vectors (d_i, d_j, d_k) as

$$T_{ijk}(p) = p + \omega_i d_i + \omega_j d_j + \omega_k d_k \quad . \tag{3}$$

This is a simple linear triangle interpolation. The warping is illustrated in Fig. 1(left). Note that instead of expressing the local warp as a linear combination

of the three displacements, we can equivalently define an affine transformation matrix A_{ijk} as

$$A_{ijk} = \begin{bmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad , \tag{4}$$

which maps (homogeneous) image points to their new locations. The matrix can be determined by solving a simple linear system of equations.

From A_{ijk} we can extract two linear functions $P^x_{ijk}(p) = a_x\, x + b_x\, y + c_x$ and $P^y_{ijk}(p) = a_y\, x + b_y\, y + c_y$, together defining the movement of point p. These definitions are later used in one of our regularization terms.

For convenience, we define some further notation used in the following equations. Given an image I, then I' denotes the warped image $I \circ T$. Additionally, I_{ijk} denotes the triangular sub-image containing only the pixels lying within the triangle (ijk).

Based on the above matching criterion and the triangle motion model, and given two images I and J (i.e. the two adjacent frames in an optical flow sequence), we can now define the higher-order likelihood in terms of triple-clique potential functions

$$\psi_{ijk}(d_i, d_j, d_k) = \mathrm{CC}'\left(I'_{ijk}, J_{ijk}\right) = 1 - \frac{\mathrm{cov}(I'_{ijk}, J_{ijk})}{\sigma_{I'_{ijk}} \sigma_{J_{ijk}}} \quad . \tag{5}$$

In fact, any labeling (d_i, d_j, d_k) yields a potential affine warp and the resulting matching cost is evaluated exactly (without approximations) for the set of pixels within the triangular sub-image. One problem remains, which is that the space of affine transformations also includes reflections. This type of transformations should not be considered in case of optical flow. We can enforce this by a simple modification on the likelihood term

$$\psi_{ijk}(d_i, d_j, d_k) = \begin{cases} \mathrm{CC}'\left(I'_{ijk}, J_{ijk}\right) & \text{if } O(i,j,k) = O(i',j',k') \\ 2 & \text{otherwise} \end{cases} \quad , \tag{6}$$

where $O(i, j, k)$ determines the orientation (i.e. clockwise or counter-clockwise) of a triangle. Note that this is a very simple and efficient geometrical operation to check whether a triangle warp constitutes a reflection. The assignment of the maximum cost of 2 for reflections avoids such unwanted warps.

An energy based on the sum of such triple-clique potentials could be sufficient for estimating the flow. It imposes some implicit regularization on the transformation since the cliques overlap at the common edge of neighboring triangles. However, texture-less regions and small triangles might benefit from an explicit regularization.

2.3 Regularization Term

Triangles covering homogeneous regions might lead to unreliable estimates. Regularization is needed such that discriminative triangles with reliable motion drive

the less reliable triangles towards a good solution. There are several ways for employing a regularization on the mesh of triangles. Here, we propose two different terms. Which of these two terms should be used depends on the application and the motion we expect to be present in the image sequence. We evaluate the performance of both terms later in our experiments.

The first regularization term is based on triple-clique potential functions and we call it the *angle deviation penalty* (ADP). The ADP is defined as

$$\psi_{ijk}(d_i, d_j, d_k) = \|(\alpha_i, \alpha_j, \alpha_k) - (\alpha'_i, \alpha'_j, \alpha'_k)\| \quad . \tag{7}$$

The term penalizes the change between the initial angles $(\alpha_i, \alpha_j, \alpha_k)$ and the angles of the warped triangle $(\alpha'_i, \alpha'_j, \alpha'_k)$ (see also Fig. 1(right)). The ADP is invariant to similarity transformations (i.e. all transformations containing only translation, rotation, and isotropic scaling).

The second term is more general and defined on quadruple-cliques. It regularizes the motion between neighboring triangles (ijk) and (ijl). We call this term *non-affine motion penalty* (NAMP) and define it as

$$\psi_{ijkl}(d_i, d_j, d_k, d_l) = \left\| \begin{matrix} \theta_k \\ \theta_l \end{matrix} \right\| \quad , \tag{8}$$

with

$$\theta_k = \left\| \begin{matrix} \delta(P^x_{ijl}, k, k'_x) \\ \delta(P^y_{ijl}, k, k'_y) \end{matrix} \right\| \quad \theta_l = \left\| \begin{matrix} \delta(P^x_{ijk}, l, l'_x) \\ \delta(P^y_{ijk}, l, l'_y) \end{matrix} \right\| \quad \delta(P, p, v) = \frac{|P(p) - v|}{\sqrt{a^2 + b^2 + 1}} \quad . \tag{9}$$

Intuitively, the term determines how well the warp of one triangle, represented by the linear functions P^x and P^y, describes the motion of the other one. If the two local warps A_{ijk} and A_{ijl} constitute an affine motion on the rectangle $(ijkl)$, then the penalty term evaluates to zero. A geometrical interpretation is illustrated in Fig. 1. We adopted the NAMP from the closely related *distances from planes* measure proposed in [28]. The NAMP can be seen as the multivariate extension.

The final energy of our higher-order CRF is then the weighted sum of the likelihood energy and the regularization energy

$$E(\mathbf{d}) = E_{\text{likelihood}}(\mathbf{d}) + \lambda\, E_{\text{regularization}}(\mathbf{d}) \quad , \tag{10}$$

where λ controls the influence of the regularization term.

3 Triangulation

So far, we have defined an energy model which enables us to use any triangulation for estimating optical flow. Since there are various ways for obtaining such triangulations, which might be more or less suitable for optical flow, we would like to discuss some of them in the following, which are all based on the popular Delaunay triangulation [29].

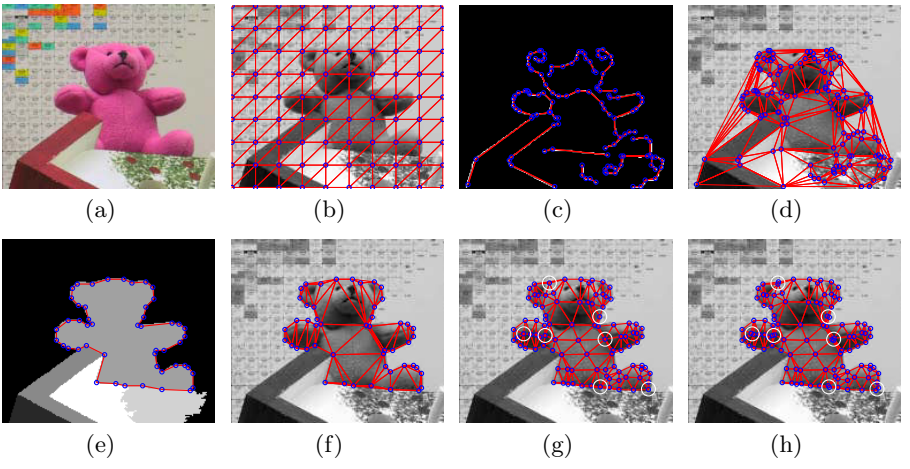


Fig. 2. Illustration of different approaches for obtaining triangulations (cf. Sec. 3) for an input image (a). Triangulation based on a regular mesh in (b), based on Canny edges in (c,d), and based on segmentation in (e,f). Mesh refinement with and without merging step in (g) and (h) (cf. Sec. 3.2).

The simplest way of defining a mesh of triangles is through a uniform distribution of nodes along the image domain (cf. Fig. 2(b)). Such regular meshes have been previously used for optical flow [8], and they can be represented by a small number of parameters (e.g. number of nodes or node spacing). While they have the advantage of simplicity, regular meshes have the drawback of a missing relation to the underlying image data. Triangles might cover different objects and thus probably different layers of motion. Here, data-dependent triangulation (DDT) seems to provide more suitable triangulations. Low-level data-dependence (e.g. using Canny edges as shown in Fig. 2(c)) would allow to place triangle edges along image edges (cf. Fig. 2(d)). However, image edges do not necessarily follow motion boundaries. In [30], a method is proposed which extracts occlusion boundaries from a single image. These boundaries might follow the real motion boundaries more closely. Another approach could be based on object segmentation. In Fig. 2(e), we utilize a mean-shift color segmentation[4] to extract the shape of the teddybear. We perform a Delaunay triangulation for boundary nodes and discard triangles outside the segmentation (cf. Fig. 2(f)). In all these examples, the nodes can be obtained with the Douglas-Peucker algorithm for line simplification [31] from any given boundary or edge image.

3.1 Layered Representation

An elegant and promising approach for motion estimation is based on a multi-layer representation, starting with the work of Wang and Adelson [10] and

[4] `http://www.caip.rutgers.edu/riul/research/code/EDISON/`

numerous ongoing developments, e.g. [32,33,12] just to name a few. However, this approach has fallen a little bit into oblivion when reviewing the list of methods in the popular Middlebury optical flow ranking. In this work, we revisit a simple but effective method for determining motion layers. We follow a similar approach as described in [33]. Initially, we use a mean-shift color segmentation on the first frame to obtain an over-segmentation. Then we estimate affine warps in a least-squares sense from displacements of the pixels in each segment. The displacements are taken from an initial motion field, which we compute in advance using our energy model and a regular mesh. Next, segments with similar affine motions are grouped by spectral clustering. For that purpose we use the end-point distance of warped image boundary points as a distance measure on affine warps and a fixed value of 15 clusters. This approach allows us to define independent meshes, one for each cluster, where each cluster represents a motion layer. This also allows us to handle occlusions and preserve discontinuities between motion layers in a natural way. Whenever two meshes overlap, we consider the mesh with a higher CC score in the overlap area to be in front of the other.

3.2 Mesh Refinement and Area Importance

As discussed earlier, larger triangles are in general more robust in providing reliable flow estimates due to the larger set of pixels considered simultaneously. Now, imagining two neighboring triangles where one of them is significantly larger than the other one, we would trust more in the motion corresponding to the energy minimum of the larger one. However, the actual energy value is independent of the size of the triangles. To this end, we propose to add an area weighting factor. The modified likelihood term becomes

$$\psi_{ijk}(d_i, d_j, d_k) = \begin{cases} \Delta_{ijk}\,\mathrm{CC}'\left(I'_{ijk}, J_{ijk}\right) & \text{if } O(i,j,k) = O(i',j',k') \\ 2\,\Delta_{ijk} & \text{otherwise} \end{cases} , \quad (11)$$

where Δ_{ijk} is the area of the triangle (ijk). Similarly, we add a weighting factor to the ADP regularization term[5].

Still, smaller triangles are more suitable for recovering local flow, in particular for areas undergoing non-rigid motion. To this end, we propose a hierarchical mesh refinement. Starting with an initial triangulation containing larger triangles which will drive the estimation in the beginning, we subsequently refine the mesh by inserting a node at the center of each edge and recompute the triangulation. Each triangle will be separated into four smaller triangles all having the same size. On this refined mesh we continue the optical flow estimation.

We demonstrate the effectiveness of this refinement strategy in a small experiment on the RubberWhale sequence, for which the ground truth flow field is available. In four different runs, we distribute triangles of same sizes – with different initial sizes in each run – over the whole image domain. We run our energy minimization over four to five levels of refinement (depending on the

[5] The NAMP already has an inherent bias towards larger triangles.

initial size), where in each level the motion of the triangles is initialized with the motion from the previous level. The motion of inserted nodes is linearly interpolated. We compute the average angular error for the estimated flow of each level. In Fig. 3 we plot the progress of the error versus the triangle size. The error decreases along with the level of refinement until a certain point where the error increases in all four runs. There seems to be a critical point where the triangle sizes are becoming too small to provide reliable motion estimates.

We conclude that a refinement of triangles improves the result, while a certain size should be preserved. This is exactly the range, where all four runs have their minimum error. In order to preserve these sizes, while still refining triangles above this range, we add a threshold on the edge length in the refinement. Nodes are only inserted on edges having at least a length of 15 pixels which results in minimum triangles of sizes between 100 and 25px^2.

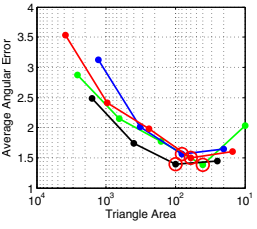
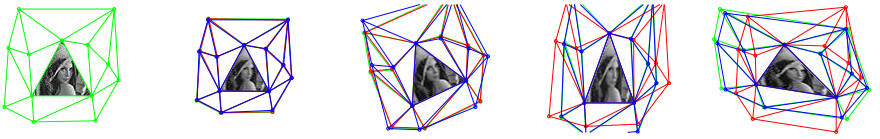


Fig. 3. Error versus Area. Colors show different runs.

In some cases the node insertion can lead to nodes lying very closely next to each other. To this end, after each mesh refinement we identify nodes whose initial position is located at almost the same position and replace the nodes by one averaged node and compute its motion as the average motion of the replaced ones. The refinement with and without this merging step is illustrated in Fig. 2(g) and 2(h).

4 Optimization

In order to optimize our CRF energy, we employ a discrete optimization over hierarchical sets of displacement vectors. We generate a search space for each optimization sweep by defining a maximum range and a sub-sampling of this range by a fixed number of displacements along the eight main directions in 2D (i.e. positive and negative horizontal, vertical, and diagonal direction). A similar quantization strategy has been previously used in [13]. The energy minimization is performed by subsequent sweeps using the QPBO-I algorithm [34], iteratively over the set of displacements. Higher-order potential functions are transformed into pairwise terms based on the reduction techniques for triple-cliques [19], and quadruple-cliques [16]. After an optimization sweep, the displacement set and thus the search range is re-scaled by a user defined factor. This procedure is repeated for a fixed number of sweeps, before we initiate a mesh refinement and rerun the optimization on the refined mesh. Throughout this work, we use fixed setting. We set the initial maximum range to 10 pixels and the number of sub-sampling steps to 5 yielding 41 displacements (including the zero-displacement). We perform 5 sweeps on one mesh level, and after each run we refine the displacements by a factor of 0.66 while we use a total of 4 mesh levels.



(a) (b) (c) (d) (e)

Fig. 4. Experiment on regularization behavior of ADP and NAMP for different types of transformations (cf. Sec. 5.1). We show the initial triangulation in (a), and in (b-e) the warp applied on (a) in green and the results for ADP in red and for NAMP in blue.

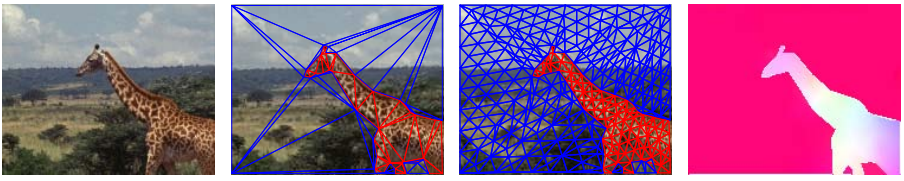
5 Experiments

5.1 ADP versus NAMP

The purpose of this experiment is to investigate the behavior of the two different regularization terms in a fully controlled setting. Remember, that ADP is invariant to similarity transformations, while NAMP is invariant to affine transformations. We define a triangulation on a test image (cf. Fig. 4(a)) where only one triangle is covering a textured part of the image. The likelihood of this triangle will be the driving force for the alignment to four different warped images. The warped images are generated by applying warps to the initial image and triangulation, i.e. an isotropic scaling, a rotation, an anisotropic scaling, and a shearing (cf. Fig. 4(b) to 4(e)). Except for the one triangle in the middle, the motion of the other triangles will result only from the regularization term. We find that both terms yield very good alignments for the outer triangles in case of similarity transformations. For pure rotation, ADP performs even slightly better, most probably due to the higher invariance of NAMP. In contrast, NAMP yields accurate alignments in case of the two affine transformations, while here ADP prevents a proper alignment of the outer triangles. We conclude that ADP should be used, when mostly similarity transformations are expected. It is also much more efficient w.r.t. to computational time than NAMP. Beyond this experiment, we experienced that NAMP based on quadruple-cliques is currently impracticable for triangulations with several thousands of triangles due to its computational demands. In the following experiment, we will again use both terms and measure the performance w.r.t. to computational time.

5.2 Giraffe

In this experiment, we perform a motion estimation on two frames of the Giraffe sequence (180×144 pixels), where the Giraffe deforms considerably. Segmentations of the giraffe are available, so we can define two motion layers, one for the giraffe and one for the background. We run the estimation with both regularization terms, and each run with three levels of mesh refinement (≈ 800 triangles on the finest level). We find a large difference in the running time. While using ADP, the optimization takes less than one minute, using NAMP takes almost



(a) (b) (c) (d)

Fig. 5. Experiment on Giraffe sequence. Target frame in (a), initial and final mesh in (b) and (c), and the resulting flow field in (d) (cf. Sec. 5.2).

Fig. 6. Flow fields for the Army and Teddy sequence for the single-layer approach using a regular mesh on the left, and results for the multi-layer approach on the right (cf. Sec. 5.3). Please note the sharp transitions at motion boundaries in case of the multi-layer approach.

ten minutes until convergence. We show the images, the initial and final meshes, and the color-encoded flow field using ADP in Fig. 5. The NAMP yields a similar result. Despite its more restrictive nature, we are able to obtain a high-accurate flow field using ADP even for the giraffe layer with highly non-rigid motion.

5.3 Middlebury

Finally, we perform an evaluation on the datasets of the Middlebury database. We compare two approaches for defining the triangulation. The first one is based on a single regular mesh, and the second one is based on the layered representation described in Sec. 3.1. Here, the resulting flow fields of the first approach are used for the affine motion clustering yielding the different motion layers. Throughout the experiments we use the ADP regularization with $\lambda = 0.3$. The remaining optimization parameters correspond to those described in Sec. 4. The initial node distance for the regular mesh is set to 60 pixels and subsequently refined to 30, 15, and 7.5. The initial motions of the multi-layer meshes are interpolated from the single-layer result.

The single-layer approach yields already quite reasonable results ranked in the midfield of the database. The multi-layer approach results in high-quality, discontinuity preserving motion fields which are competing with the best methods currently listed in the ranking, including advanced variational methods. In Fig. 6 we show some visual results. The detailed quantitative evaluation can be found online on the Middlebury website and in the supplementary material.

The computationally expensive part of our method is the likelihood evaluation, in particular on the finer mesh levels containing a large number of triangles ($>10,000$). Since the computations are based on rather simple geometrical triangle operations and linear interpolation, a tremendous speed-up might be achieved by GPU implementation providing efficient, hardware-supported functionalities.

6 Conclusion

We propose a novel CRF model with higher-order likelihoods for the application of optical flow beyond the assumption of constant flow. Likelihood terms are defined on local pixel regions whose motions are constrained to local affine warps through triangle-based parameterization. The energies are defined as triple-cliques for the likelihood as well as the similarity invariant regularization term, while non-affine motions can be penalized through quadruple-clique energies. To our best knowledge, this is the first time that higher-order CRF likelihoods are modeled in such a way. Here, the main advantage of our approach is that the energies are evaluated exactly without approximations yielding a robust and reliable matching process. An interesting direction would be to integrate the whole process of triangulation and motion layer definition into the optimization. A prior on the maximum number of layers, as well as a flow-dependent mesh-refinement could further improve the the results. A step beyond our current approach could allow for the definition of higher-order likelihoods with arbitrary shapes and without restrictions through the parametrization. We believe our model can be seen as a building block for new directions in CRF modeling in computer vision, which directly benefit from future advances in CRF optimization.

References

1. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17 (1981)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23 (2001)
3. Lempitsky, V., Roth, S., Rother, C.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In: CVPR (2008)
4. Trobin, W., Pock, T., Cremers, D., Bischof, H.: Continuous energy minimization via repeated binary fusion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 677–690. Springer, Heidelberg (2008)
5. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: Microsoft Research Technical Report MSR-TR-2009-179 (2009)
6. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)
7. Veksler, O.: Fast variable window for stereo correspondence using integral images. In: CVPR (2003)
8. Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., Navab, N.: Optical flow estimation with uncertainties through dynamic mrfs. In: CVPR (2008)
9. Hirschmuller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. PAMI 31 (2009)

10. Wang, J.Y.A., Adelson, E.H.: Representing moving images with layers. IEEE Image Processing 3 (1994)
11. Rother, C., Kolmogorov, V., Blake, A.: grabcut: Interactive foreground extraction using iterated graph cuts. ACM SIGGRAPH 23 (2004)
12. Kumar, M.P., Torr, P., Zisserman, A.: Learning layered motion segmentations of video. IJCV 76 (2008)
13. Heibel, T.H., Glocker, B., Groher, M., Paragios, N., Komodakis, N., Navab, N.: Discrete tracking of parametrized curves. In: CVPR (2009)
14. Komodakis, N., Paragios, N.: Beyond pairwise energies: Efficient optimization for higher-order mrfs. In: CVPR (2009)
15. Rother, C., Kohli, P., Feng, W., Jia, J.: Minimizing sparse higher order energy functions of discrete variables. In: CVPR (2009)
16. Ishikawa, H.: Higher-order clique reduction in binary graph cut. In: CVPR (2009)
17. Lempitsky, V., Rother, C., Blake, A.: Logcut - efficient graph cut optimization for markov random fields. In: ICCV (2007)
18. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. PAMI 32 (2010)
19. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? PAMI 26 (2004)
20. Hammer, P.L., Hansen, P., Simeone, B.: Roof duality, complementation and persistency in quadratic 0-1 optimization. Mathematical Programming 28 (1984)
21. Kolmogorov, V., Rother, C.: Minimizing nonsubmodular functions with graph cuts-a review. PAMI 29 (2007)
22. Woodford, O.J., Torr, P.H.S., Reid, I.D., Fitzgibbon, A.W.: Global stereo reconstruction under second order smoothness priors. In: CVPR (2008)
23. Kwon, D., Lee, K.J., Yun, I.D., Lee, S.U.: Nonrigid image registration using dynamic higher-order mrf model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 373–386. Springer, Heidelberg (2008)
24. Kohli, P., Ladicky, L., Torr, P.H.: Robust higher order potentials for enforcing label consistency. IJCV 82 (2009)
25. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In: CVPR (1996)
26. Béréziat, D.: Object based optical flow estimation with an affine prior model. In: ICPR (2000)
27. Nir, T., Bruckstein, A.M., Kimmel, R.: Over-parameterized variational optical flow. IJCV 76 (2008)
28. Dyn, N., Levin, D., Rippa, S.: Data dependent triangulations for piecewise linear interpolation. IMA Journal of Numerical Analysis 10 (1990)
29. Chew, L.P.: Constrained delaunay triangulations. In: Annual Symposium on Computational Geometry (SCG). ACM, New York (1987)
30. Hoiem, D., Stein, A.N., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV (2007)
31. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10 (1973)
32. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. IJCV 62 (2005)
33. Min, C., Medioni, G.: Motion segmentation by spatiotemporal smoothness using 5d tensor voting. In: CVPR Workshop (2006)
34. Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M.: Optimizing binary mrfs via extended roof duality. In: CVPR (2007)