Optical Flow Estimation using Laplacian Mesh Energy

Wenbin Li Darren Cosker Matthew Brown Rui Tang

MTRC, Department of Computer Science
University of Bath, BA2 7AY, UK
{w.li,d.p.cosker,m.brown,r.tang}@bath.ac.uk

Abstract

In this paper we present a novel non-rigid optical flow algorithm for dense image correspondence and non-rigid registration. The algorithm uses a unique Laplacian Mesh Energy term to encourage local smoothness whilst simultaneously preserving non-rigid deformation. Laplacian deformation approaches have become popular in graphics research as they enable mesh deformations to preserve local surface shape. In this work we propose a novel Laplacian Mesh Energy formula to ensure such sensible local deformations between image pairs. We express this wholly within the optical flow optimization, and show its application in a novel coarse-to-fine pyramidal approach. Our algorithm achieves the state-of-the-art performance in all trials on the Garg et al. dataset, and top tier performance on the Middlebury evaluation.

1. Introduction

Optical flow estimation is an important area of computer vision research. Current algorithms can broadly be classified into two categories – variational methods and discrete optimization methods. The former is a continuous approach [5, 6, 18] to estimate optical flow based on modifications of Horn and Schunck's framework proposed in [9]. Such approaches can provide high subpixel accuracy but may be limited by minimization of the non-convex energy function. The latter [4, 14] is based on combinatorial optimization algorithms such as min-cut and max-flow, which can recover non-convex energy functions and multiple local minima but may suffer from discretization artifacts, e.g. the optical flow field boundary is aligned with the coordinate axes. One desirable property of optical flow techniques is to preserve local image detail and also handle non-rigid image deformations. Under such deformations, the preservation of local detail is particularly important. Garg *et al.* [7] impose this by maintaining correlations between 2D trajectories of different points on a non-rigid surface using a variational framework. Pizarro *et al.* [12] propose a feature matching approach based on local surface smoothness, and also show particular application to non-rigidly deforming objects.

In computer graphics research, a common requirement is that surface meshes are globally editable, but capable of maintaining local details under mesh deformations. In order to provide a flexible representation to allow computation and preservation of such details, Laplacian mesh structures have previously been described [13, 11]. Such schemes impose constraints in differential Laplacian coordinates calculated upon groups of triangles associated with each vertex. Meshes have previously been used in optical flow estimation [8]. However, this is to reduce processing complexity as opposed to specifically imposing smoothness.

In this paper we present an variational optical flow model which introduces a novel discrete energy based on *Laplacian Mesh Deformation*. Such deformation approaches are widely applied in graphics research, particularly for preserving local details [13, 11]. In our work we propose that the same concept, i.e. that of an underlying mesh which penalizes local movements and preserves smooth global ones, can be of great use for optical flow and tracking. Constraints on the local deformations expressed in Laplacian coordinates encourage local regularity of the mesh whilst allowing global non-rigidity. Our algorithm applies a mesh to an image with a resolution up to one vertex per pixel. The *Laplacian Mesh Energy* is described as an additional term for the energy function, and can be applied in a straightforward manner using our proposed minimization strategy. In addition, a novel coarse-to-fine approach is described for overcoming the loss of small optical flow details during its propagation between adjacent pyramid levels.

We evaluate our approach on the widely recognized *Middlebury* dataset [2] as well as the publicly available non-rigid data set proposed by Garg *et al.* [7]. Our approach provides excellent performance ranked in the top tier of the *Middlebury* evaluation[1], and either outperforms or shows comparable accuracy against the leading publicly available non-rigid approaches when evaluated on the non-rigid data set of Garg *et al.*

[1] http://vision.middlebury.edu/flow/eval/results/results-e1.php

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.315

2. Hybrid Energy

In this section, we introduce our novel hybrid energy formula in which our algorithm considers a pair of consecutive frames in an image sequence. The current frame is denoted by $I_1(\mathbf{X})$ and its successor by $I_2(\mathbf{X})$, where $\mathbf{X} = (x, y)^T$ is a pixel location in the image domain Ω. We define the optical flow displacement between $I_1(\mathbf{X})$ and $I_2(\mathbf{X})$ as $\mathbf{w} = (\mathbf{u}, \mathbf{v})^T$. In the proposed optical flow estimation approach, the core energy function can be expressed by the following:

$$E(\mathbf{w}) = E_{Data}(\mathbf{w}) + \lambda E_{Lap}(\mathbf{w}) + \xi E_{Smooth}(\mathbf{w}) \quad (1)$$

where $E_{Data}(\mathbf{w})$ denotes a data term that expresses both *Intensity Constancy* and *Gradient Constancy* assumptions on pixel values between $I_1(\mathbf{X})$ and $I_2(\mathbf{X})$. Similar to [5, 9], a smoothness term is introduced into the formula, which controls global flow smoothness. The term E_{Lap} represents our core contribution, i.e. the *Laplacian Mesh Energy* $E_{Lap}(\mathbf{w})$. All the three terms are detailed in the following sections.

2.1. Continuous Intensity Energy

Following the standard optical flow assumption regarding *Intensity Constancy*, we assume that the gray value of a pixel is not varied by its displacement through the entire image sequence. In addition, we also make a *Gradient Constancy* assumption which is engaged to provide additional stability in case the first assumption (*Intensity Constancy*) is violated by changes in illumination. The data term of energy function encoding these assumptions is therefore formulated as:

$$E_{Data}(\mathbf{w}) = \int_{\Omega} \Psi(\|I_2(\mathbf{X}+\mathbf{w}) - I_1(\mathbf{X})\|^2 + \theta \cdot \|\nabla I_2(\mathbf{X}+\mathbf{w}) - \nabla I_1(\mathbf{X})\|^2)\mathbf{dX} \quad (2)$$

For robustness regarding occlusions and boundaries, we apply the Lorentzian as the penalty function $\Psi(s) = log(1 + s^2/2\epsilon^2)$ with $\epsilon = 0.001$ to solve this formula. The term $\nabla = (\partial_{xx}, \partial_{yy})^T$ is the spatial gradient and $\theta \in [0, 1]$ denotes a weight that can be manually assigned with different values. Furthermore, the smoothness term of our algorithm is a dense pixel based regularizer that penalizes global variation. The objective is to produce a globally smooth optical flow field:

$$E_{Smooth}(\mathbf{w}) = \int_{\Omega} \Psi(\|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{v}\|^2)\mathbf{dX} \quad (3)$$

where the robustness function $\Psi(s^2)$ is used again.

2.2. Discrete Laplacian Mesh Energy

In order to improve optical flow estimation against the local complexity of non-rigid motion, a novel *Laplacian Mesh Energy* concept is proposed in this section. The aim of this energy is to account for non-rigid motion in scene deformation. This concept is inspired by *Laplacian Mesh Deformation* research in graphics, which aims to preserve local mesh smoothness under non-linear transformation [13]. The usage of this concept in computer vision research for optical flow estimation is introduced for the first time here. Although non-rigid motion is highly non-linear, the movement of pixels in such deformations still often exhibits strong correlations in local regions. To represent this, we propose a quantitative *Mesh Deformation Weight* based on Laplacian coordinates. The scheme was originally presented by Meyer *et al.* [11] for mesh deformation. Let $\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ be a triangular mesh where $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}$ describes geometric positions of the vertices in absolute cartesian coordinates, $\mathbf{E}$ denotes the set of edges, and $\mathbf{F}$ the set of faces. Considering a small mesh region, each vertex v_i has a *neighborhood ring* denoted by $\mathcal{N}_i = \{j \mid (i, j) \in \mathbf{E}\}$ which is the set of adjacent vertices of vertex v_i. The *degree* d_i of v_i is the number of elements in $\mathcal{N}_i$. Here the mesh geometric motion is described by differentials instead of absolute Cartesian coordinates. We define the differentials set as $\mathbf{L} = \{\delta_1, \delta_2, \ldots \delta_n\}$ where the coordinate is presented as the difference between the vertex v_i and the geometric average of its neighbors, i.e. $\delta_i = \mathcal{L}(v_i)$. We have

$$\mathcal{L}(v_i) = v_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} v_j. \quad (4)$$

These uniform weights are found sufficient for the 2D mesh in our evaluation. Next, we have the mesh energy in Laplacian coordinates as follows:

$$E_{Lap}(\mathbf{w}) = \sum_{i=1}^{n} \|\mathcal{L}(v_i + \mathbf{w}_i) - \mathcal{L}(v_i)\|^2 \quad (5)$$

Where $\mathbf{w}_i$ denotes the motion of the vertices v_i. This term of the energy function penalizes the shape variance after vertex motion. The rationale of using this energy is that the Laplacian coordinates $\mathbf{L}$ encode relative information between vertices and can therefore be used to preserve shape under mesh deformation.

3. Optical Flow Framework

Table 1 outlines our overall optical flow framework. In order to utilize the *Laplacian Mesh Energy* it is required to create a mesh over the initial image I_1. Ideally, we desire that the triangles of this mesh do not overlap boundaries in the scene as this may lead to distortions given parallax motion between objects at different depths. We there-

Input: two images I_1 and I_2
1. Edge-Aware Mesh $\mathcal{M}_1$ Initialization (Sec. 3.1)
2. n-levels Gaussian pyramids are constructed for both the images and the mesh. Set the initial pyramid level $k = 0$ and initial flow field $\mathbf{w}^k = (0, 0)^T$
3. The flow field is propagated to level $k + 1$
  4. Detail-Aware Flow Field Enhancement (Sec. 3.2)
    4.1 Estimate the tracked mesh $\mathcal{M}_2^k$ for I_2^k.
    4.2 Flow field Enhancement using $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$.
  5. Hybrid Energy Optimization (Sec. 3.3)
    5.1 Generate continuous *Laplacian Mesh Energy* using meshes $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$.
    5.2 Nested fixed point iterations.
6. If $k \neq n - 1$ then $k = k + 1$ and go to step 3
Output: optical flow field

Table 1. The overall framework of our optical flow model.

fore first present an *Edge-Aware Mesh Initialization* scheme (Sec 3.1) as part of our framework.

We also present a novel coarse-to-fine pyramidal framework [5] to utilize our *Laplacian Mesh Energy* in a variational model. In our framework we overcome a previous limitation of such pyramidal approaches, i.e. the loss of small flow details when propagating flow field from coarse to finer pyramidal levels. In such cases, small image details at a finer level of the pyramid are lost due to flow computation being initially performed on a coarsely sampled version of the image. As such, the flow for these detailed regions is not remained and propagated to the finer level.

Finally, an optimization scheme (Sec. 3.3) is proposed to minimize the discrete *Laplacian Mesh Energy* on every level of the pyramidal framework. In the following subsections each step is described in detail.

3.1. Edge-Aware Mesh Initialization

The proposed algorithm is input by an image pair and a mesh with triangle edges that follow object boundaries in one of the images as closely as possible. We will discuss the implications of mesh design and its affect on our algorithms behavior in the evaluation. The underlying mesh is an essential part of *Laplacian Mesh Energy* computation. Using a uniform mesh with equal distances between vertices along its horizontal and vertical adjacent neighbors is one strategy that can be employed in our approach. However, in such a case the grid elements within the mesh will typically overlap the boundaries of objects scene, which results in unexpected errors in our energy minimization. This is because triangles within the mesh will be skewed given parallax motion between different objects at different image depths, resulting in a noisier flow field in these areas.

In order to address this issue, we propose an edge-aware meshing scheme which operates as follows: First, we create two edge maps on the input image using *SLIC Superpixels* [1] and *Sobel Kernel* edge detection respectively. We then apply a binary *AND Operation* on the two edge maps in order to deduce uncommon edges, and remove noise using a Gaussian filter. The rationale behind this approach is that the *Sobel kernel* returns a large number of candidate edges, but also multiple false-positive noise like edges relating to image detail as opposed to object boundaries. The *SLIC Superpixels* on the other hand is less likely to create boundaries relating to image detail. Performing an AND operation eliminates a great deal of the noisy edge boundaries and retains a large proportion of reliable ones. Finally, we construct a triangular mesh $\mathcal{M}_1$ using *Delaunay* triangulation on the remaining edge points.

Given the input mesh $\mathcal{M}_1$, an n-level image pyramid is built (Table 1). The input images I_1, I_2 along with the mesh $\mathcal{M}_1$ are resized with the same sampling rate on each level, denoted by I_1^k, I_2^k and $\mathcal{M}_1^k$, where $k = 1, 2, \ldots, n$. We then perform *Detail-Aware Flow Field Enhancement* and *Hybrid Energy Optimization* on each level.

3.2. Detail-Aware Flow Field Enhancement

As mentioned in the beginning of Sec. 3, the aim of this step is to preserve small flow details which may be lost when propagated from the adjacent coarser level. First, we estimate a mesh $\mathcal{M}_2^k$ by propagating the mesh $\mathcal{M}_1^k$ from I_1^k onto I_2^k. Next, we build a labelling model using vertex displacement vectors and solve it to retain small flow details. The whole process is detailed in the next two sections.

3.2.1 Frame-Frame Tracked Mesh $\mathcal{M}_2$ Estimation

Iterative Refinement Algorithm

$\mathbf{V}, \mathbf{V}_c, \mathbf{V}'_c$
$\mathbf{V}_c \rightarrow \mathbf{V}'_c$
$\mathbf{V}_c \subset \mathbf{V}$
while not $\mathbf{V}_c = \mathbf{V}$
  $\mathbf{V}' := \min_{\mathbf{V}'}\{\|\mathbf{L}\mathbf{V}' - \mathbf{L}\mathbf{V}\|^2 + \sum_{i=1}^{m} \|v'_{c\cdot i} - v_{c\cdot i}\|^2\}$
  for all $v \in \mathbf{V}, v' \in \mathbf{V}'$ **do**
    if $Err(v \rightarrow v') < \eta$ **then**
      $\mathbf{V}_c := \mathbf{V}_c \cup \{v\}, \mathbf{V}'_c := \mathbf{V}'_c \cup \{v'\}$
    end if
  end for
end while

Table 2. The iterative refinement algorithm for tracked mesh $\mathcal{M}_2^k$ estimation.

In order to propagate the mesh from $\mathcal{M}_1^k$ to $\mathcal{M}_2^k$ at pyramid level k, we employ an *Anchor Patch* based technique and *Laplacian Mesh Deformation*, which utilizes I_1^k, I_2^k and $\mathcal{M}_1^k$. We follow the *Anchor Patch* process outlined in [10] to achieve this mesh propagation: SIFT features are initially detected and matched between images I_1^k and I_2^k – given a corresponding set of features between each image. We then

go through every vertex v of $\mathcal{M}_1^k$ and search for the three nearest SIFT features f_* within a 9×9 search window centered on the vertex v in I_1^k. The corresponding features in I_2^k and *Barycentric Coordinate Mappings* – defined by the triangle formed by the 3 SIFT features – are used to calculate a corresponding vertex v' for $\mathcal{M}_2^k$ in I_2^k.

Next, we apply an error function $Err(v \rightarrow v')$ from [3, 10] on all the newly created vertex correspondences between I_1^k and I_2^k. This is carried out in order to select the most reliable vertex matches between the two images. The error function provides an *Error Score* which is decided by the Euclidean distance between the average pixel value of two small regions (3×3) centered on v and v' respectively. The vertex matches with low errors are selected as sets of control points – defined here as $\mathbf{V}_c, \mathbf{V}'_c$ – where $\{\mathbf{V}_c, \mathbf{V}'_c | \forall v_{c \cdot i} \in \mathbf{V}_c, \forall v'_{c \cdot j} \in \mathbf{V}'_c, Err(v_{c \cdot i} \rightarrow v'_{c \cdot j}) < \eta\}$ where η is our predefined error threshold. This method of creating anchor point correspondences between meshes has previously been utlized to obtain reliable vertex matches between images in several recent state-of-the-art tracking frameworks [3, 10].

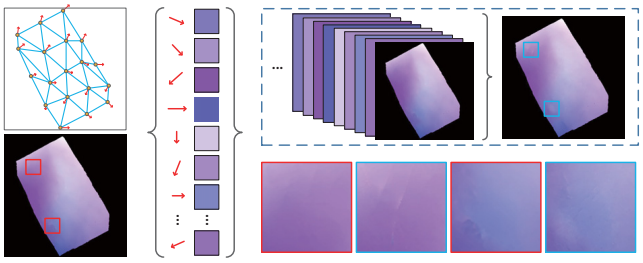
Finally, in order to estimate the positions of the remaining vertices in $\mathcal{M}_2^k$, *Laplacian Mesh Deformation* [13] is applied using $\mathcal{M}_1^k$ and the corresponding control points $\mathbf{V}_c$ and $\mathbf{V}'_c$. We minimize the following function to achieve this:

$$\min_{\mathbf{V}'} \{\|\mathbf{L}\mathbf{V}' - \mathbf{L}\mathbf{V}\|^2 + \sum_{i=1}^{m} \|v'_{c \cdot i} - v_{c \cdot i}\|^2\} \qquad (6)$$

where $\mathbf{L}$ is a Laplacian matrix computed using Eq. (4), $\mathbf{V}$ represents the vertex set of $\mathcal{M}_1^k$, and the number of control points is captured by m. After minimizing [13] Eq. (6), we obtain our initial mesh $\mathcal{M}_2^k$ for I_2^k, and denote this set of vertices as $\mathbf{V}'$. However, this set of vertices may contain outliers due to the limited number of control points. We therefore propose an iterative refinement algorithm to remove these (Table 2). In each iteration of our algorithm, we apply our evaluation function (Err) on the matches between $\mathbf{V}$ and $\mathbf{V}'$ to obtain low-scoring matches by which the control point sets $\mathbf{V}_c$ and $\mathbf{V}'_c$ are updated. These are propagated onto the next iteration until all the matches between $\mathbf{V}$ and $\mathbf{V}'$ reach the *Error Score* threshold (under η). In our implementation, if the $\mathbf{V}_c = \mathbf{V}$ does not converge within 15 iterations, a thin-plate spline is employed on the point set $\mathbf{V} - \mathbf{V}_c$ to estimate positions of the points with higher than acceptable errors.

3.2.2 Inter-Level Flow Field Enhancement

In this section we consider the small flow details encoded by mesh $\mathcal{M}_2^k$ from the previous section. In existing pyramidal approaches, small motion displacements on a finer level can be lost when the flow field is propagated from a coarser level. To address this issue we utilize a labeling model



(a) (b) (c)

Figure 1. Small flow Details Preservation. (a) **Top**: The mesh and vertex displacement vectors (red arrows). **Bottom**: The flow field $\mathbf{w}$ propagated from the adjacent coarser level. (b) The selected flow candidates and their color coding. (c) **Top**: The labeling model optimized using QPBO. **Bottom**: The visual comparison of closeups between $\mathbf{w}$ (red) and the optimized flow field $\widehat{\mathbf{w}}$ (blue).

which identifies discrepancies between the propagated flow field $\mathbf{w}$ and a displacement vector $\mathbf{w}'$. The process is outlined in Figure 1. Given our mesh $\mathcal{M}_2^k$ for image I_2^k, the displacement vector $\mathbf{w}' = \{w'_1, w'_2, \ldots, w'_n\}$ – where n is number of the vertices – is computed by using the vertices differences between $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$.

For each w'_i, we consider the flow vector w_i in $\mathbf{w}$ where w'_i and w_i share the same pixel position. We compute the Euclidean distance between w'_i and w_i, and for extra robustness, we also compute the Euclidean distances between w'_i and the 8 adjacent neighbors of w_i. The displacement w'_i is regarded as a potential flow candidate only if all 9 Euclidean distances are larger than 1 pixel. This process is repeated on all n displacement vectors in $\mathbf{w}$ to give a new flow candidate list $\{w'_{g \cdot 1}, w'_{g \cdot 2}, \ldots, w'_{g \cdot m}\}$ where $m \leqslant n$. The number of new flow candidates is usually between 20 and 25 in our experiments. These new flow candidates are typically distributed widely over the whole image including generally featureless regions (as opposed to those detected using e.g. SIFT [17]). Given the dense mesh matches (between $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$) used to calculate $\mathbf{w}'$, this provides additional robustness to small feature displacement changes.

Having obtained the new flow candidate list, we then assume that each pixel has $m + 1$ choices to be selected from either the new flow candidates or the original flow field $\mathbf{w}$ – which can be treated as a labeling problem. *Quadratic Pseudo-Boolean Optimization* (QPBO) is employed to solve this problem in our implementation, and has previously been used in other state-of-the-art optical flow methods for similar labeling solutions [17]. However, in our work the correspondences provided by $\mathbf{w}'$ are dense, and thus can potentially retain smaller flow details that would otherwise be lost in pyramidal flow propagation, or feature matching that might be less robust and sparser given non-rigidly deforming scenes.

The process described above (Figure 1) outputs a flow field $\widehat{\mathbf{w}}^k$ and $\widehat{\mathbf{w}}^k = (\widehat{u}^k, \widehat{v}^k)^T$, which is then used as the initial flow field for computing $\mathbf{w}^{k+1}$ on level $k+1$ as below.

3.3. Hybrid Energy Optimization

Due to the highly non-linear nature of the energy function $E(\mathbf{w})$, its optimization is an essential part of our algorithm. In this section, we introduce a numerical scheme to minimize this hybrid energy w.r.t. the discrete Laplacian mesh energy and the continuous intensity energy. We initially define mathematical abbreviations (similar to [5]) for our intensity energy minimization as follows:

$$\begin{aligned} I_x &= \partial_x I_2(\mathbf{X}+\mathbf{w}) & I_{yy} &= \partial_{yy} I_2(\mathbf{X}+\mathbf{w}) \\ I_y &= \partial_y I_2(\mathbf{X}+\mathbf{w}) & I_{xx} &= \partial_{xx} I_2(\mathbf{X}+\mathbf{w}) \\ I_z &= I_2(\mathbf{X}+\mathbf{w}) - I_1(\mathbf{X}) & I_{xz} &= \partial_x I_2(\mathbf{X}+\mathbf{w}) - \partial_x I_1(\mathbf{X}) \\ I_{xy} &= \partial_{xy} I_2(\mathbf{X}+\mathbf{w}) & I_{yz} &= \partial_y I_2(\mathbf{X}+\mathbf{w}) - \partial_y I_1(\mathbf{X}) \end{aligned}$$

In order to minimize the mesh energy in our variational model, we define its uniform weights in polar coordinates. We have $\mathcal{L} = (\mathcal{L}_r, \mathcal{L}_\theta)^T$ where $\mathcal{L}_r$ denotes the magnitude component and $\mathcal{L}_\theta$ denotes the angle component, which results in two terms for the Laplacian mesh energy as follows:

$$\begin{aligned} E_{Lap}(\mathbf{w}) = \lambda \int_\Omega \Psi(\|\mathcal{L}_{r\cdot 2}(\mathbf{X}+\mathbf{w}) - \mathcal{L}_{r\cdot 1}(\mathbf{X})\|^2)\mathbf{dX} \\ +\lambda \int_\Omega \Psi(\|\mathcal{L}_{\theta\cdot 2}(\mathbf{X}+\mathbf{w}) - \mathcal{L}_{\theta\cdot 1}(\mathbf{X})\|^2)\mathbf{dX} \end{aligned} \quad (7)$$

where both the terms $\mathcal{L}_{*\cdot 1}$ and $\mathcal{L}_{*\cdot 2}$ are computed respectively based on $\mathcal{M}_1^k$ and $\mathcal{M}_2^k$ (Sec. 3.2.1) on level k. Note that the terms are applied to each pixel of the input image. We go through each triangle and employ bicubic interpolation using the $\mathcal{L}_*$ values on the three vertices of the triangle. This process results in a continuous *Laplacian Mesh Energy* presented in Eq. (7). The term λ is a weight capturing the influence of our Laplacian mesh, and is set to 0.6 in our experiments. The behavior of our algorithm by varying λ values is also considered in the evaluation section. The mathematical abbreviations for the *Laplacian Mesh Energy* are as follows:

$$\begin{aligned} \mathcal{L}_{*\cdot x} &= \partial_x \mathcal{L}_*(\mathbf{X}+\mathbf{w}) \\ \mathcal{L}_{*\cdot y} &= \partial_y \mathcal{L}_*(\mathbf{X}+\mathbf{w}) \\ \mathcal{L}_{*\cdot z} &= \mathcal{L}_{*\cdot 2}(\mathbf{X}+\mathbf{w}) - \mathcal{L}_{*\cdot 1}(\mathbf{X}) \end{aligned}$$

Our energy function $\mathbf{E}(\mathbf{w})$ is highly nonlinear on the terms of $\mathcal{L}_*$, $\mathbf{w}$ and Ψ. We employ two nested *Fixed Point Iterations* on $\mathbf{w}$ after Euler-Lagrange equations are applied.

Fix w for I_*^{k+1} and $\mathcal{L}_*^{k+1}$. In the first fixed point iteration, the algorithm goes through every level of the pyramid starting from the top/coarsest level. We assume that $\mathbf{w}$ converges at the k-th iteration (the k-th level of the pyramid) giving us $\mathbf{w}^k = (\mathbf{u}^k, \mathbf{v}^k)^T, k = 0, 1, \ldots$ with an initialization $\mathbf{w}^0 = (0,0)^T$ at the coarsest level of the pyramid. The flow field $\mathbf{w}^k$ is then propagated to the next finer level for computing the initial flow field $\widehat{\mathbf{w}}^k$ (sec. 3.2). However, the new system reached fixed $\mathbf{w}^k$ is still nonlinear and difficult to solve as it contains terms I_*^{k+1}, $\mathcal{L}_*^{k+1}$ and the nonlinear function Ψ'.

Fix dw for Ψ'. First order Taylor expansion is employed on the terms I_z^{k+1}, I_{xz}^{k+1}, I_{yz}^{k+1}, $\mathcal{L}_{*\cdot x}^{k+1}$, $\mathcal{L}_{*\cdot y}^{k+1}$ and $\mathcal{L}_{*\cdot z}^{k+1}$ in order to remove the nonlinearity of I_*^{k+1} and $\mathcal{L}_*^{k+1}$. We have $I_z^{k+1} \approx I_z^k + I_x^k du^k + I_y^k dv^k$ and $\mathcal{L}_{*\cdot z}^{k+1} \approx \mathcal{L}_{*\cdot z}^k + \mathcal{L}_{*\cdot x}^k du^k + \mathcal{L}_{*\cdot y}^k dv^k$, where we assume that the flow field on level $k+1$ can be estimated by the flow field and the incremental from previous level k, denoted as $\mathbf{w}^{k+1} \approx \widehat{\mathbf{w}}^k + \mathbf{dw}^k$. Two unknown increments du^k, dv^k and two known flow fields $\widehat{u}^k$,$\widehat{v}^k$ can be obtained from the previous iteration. Note that this assumption also applies to the terms $\mathcal{L}_{*\cdot z}^{k+1}$. For removing nonlinearity in Ψ' with unknown increments du^k and dv^k, we apply a nested second fixed point iteration. Here, in every iteration step we assume that both $du^{k,j}$ and $dv^{k,j}$ converge within j iteration steps with initialization of $du^{k,0} = 0$ and $dv^{k,0} = 0$. Therefore, the final linear system is obtained in $du^{k,j+1}$ and $dv^{k,j+1}$ as follows:

$$\begin{aligned} &(\Psi')_{Data}^{k,j} \cdot (I_x^k(I_z^k + I_x^k du^{k,j+1} + I_y^k dv^{k,j+1}) \\ &+\theta\,[I_{xx}^k(I_{xz}^k + I_{xx}^k du^{k,j+1} + I_{xy}^k dv^{k,j+1}) \\ &+I_{xy}^k(I_{yz}^k + I_{xy}^k du^{k,j+1} + I_{yy}^k dv^{k,j+1})]) \\ &+\lambda\,(\Psi')_{Lap\cdot r}^{k,j} \cdot \mathcal{L}_{r\cdot x}^k(\mathcal{L}_{r\cdot z}^k + \mathcal{L}_{r\cdot x}^k du^{k,j+1} + \mathcal{L}_{r\cdot y}^k dv^{k,j+1}) \\ &+\lambda\,(\Psi')_{Lap\cdot \theta}^{k,j+1} \cdot \mathcal{L}_{\theta\cdot x}^k(\mathcal{L}_{\theta\cdot z}^k + \mathcal{L}_{\theta\cdot x}^k du^{k,j+1} + \mathcal{L}_{\theta\cdot y}^k dv^{k,j+1}) \\ &-\xi\,\mathbf{Div}(\Psi')_{Smooth}^{k,j} \cdot \nabla(u^k + du^{k,j+1}) = 0 \end{aligned} \quad (8)$$

$$\begin{aligned} &(\Psi')_{Data}^{k,j} \cdot (I_y^k(I_z^k + I_x^k du^{k,j+1} + I_y^k dv^{k,j+1}) \\ &+\theta\,[I_{yy}^k(I_{yz}^k + I_{xy}^k du^{k,j+1} + I_{yy}^k dv^{k,j+1}) \\ &+I_{xy}^k(I_{xz}^k + I_{xx}^k du^{k,j+1} + I_{xy}^k dv^{k,j+1})]) \\ &+\lambda(\Psi')_{Lap\cdot r}^{k,j} \cdot \mathcal{L}_{r\cdot y}^k(\mathcal{L}_{r\cdot z}^k + \mathcal{L}_{r\cdot x}^k du^{k,j+1} + \mathcal{L}_{r\cdot y}^k dv^{k,j+1}) \\ &+\lambda(\Psi')_{Lap\cdot \theta}^{k,j} \cdot \mathcal{L}_{\theta\cdot y}^k(\mathcal{L}_{\theta\cdot z}^k + \mathcal{L}_{\theta\cdot x}^k du^{k,j+1} + \mathcal{L}_{\theta\cdot y}^k dv^{k,j+1}) \\ &-\xi\,\mathbf{Div}(\Psi')_{Smooth}^{k,j} \cdot \nabla(v^k + dv^{k,j+1}) = 0 \end{aligned} \quad (9)$$

Where $(\Psi')_{Data}^k$ and $(\Psi')_{Lap\cdot *}^k$ provides both robustness against occlusion and sharpness on object boundaries, $(\Psi')_{Smooth}^k$ is defined as diffusivity in the global smoothness terms [5] as below:

$$\begin{aligned} (\Psi')_{Data}^k &= \Psi'((I_z^k + I_x^k du^k + I_y^k dv^k)^2 \\ &+\theta[(I_{xz}^k + I_{xx}^k du^k + I_{xy}^k dv^k)^2 + (I_{yz}^k + I_{xy}^k du^k + I_{yy}^k dv^k)^2]) \\ (\Psi')_{Lap\cdot *}^k &= \Psi'(\mathcal{L}_{*\cdot z}^k + \mathcal{L}_{*\cdot x}^k du^k + \mathcal{L}_{*\cdot y}^k dv^k)^2 \\ (\Psi')_{Smooth}^k &= \Psi'(\left\|\nabla(u^k + du^k)\right\|^2 + \left\|\nabla(v^k + dv^k)\right\|^2) \end{aligned}$$

In our implementation, an n-level image pyramid is constructed by using a down sampling factor of 0.75 and *Bicubic Interpolation* on each pyramid level. Furthermore, the first fixed point iterations are set based on both the down sampling factor and the image size while the nested second fixed point iterations are fixed to 5 steps. Finally, the large linear systems (Eq. (8) and (9)) are solved using *Conjugate*

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
MDP-Flow2 [74]	4.2	0.08 4	0.21 2	0.07 9	**0.15** 1	**0.48** 1	**0.11** 1	**0.20** 1	**0.40** 1	**0.14** 1	0.15 8	0.80 15	0.08 6	0.63 6	0.93 6	0.43 6	**0.26** 1	**0.76** 1	0.23 3	0.11 9	0.12 6	0.17 11	**0.38** 1	**0.79** 1	**0.44** 1
ADF [71]	9.5	0.08 4	0.22 6	**0.06** 1	0.18 4	0.62 9	0.14 9	0.29 14	0.71 18	0.17 8	0.16 17	0.91 28	**0.07** 1	0.69 13	1.03 13	0.47 8	0.43 13	0.91 4	0.28 6	0.12 15	0.12 6	0.20 20	0.43 2	0.88 2	0.63 7
LME [76]	9.7	0.08 4	0.22 6	**0.06** 1	**0.15** 1	0.49 2	**0.11** 1	0.30 17	0.64 9	0.31 51	0.15 8	0.78 13	0.09 13	0.66 8	0.96 8	0.53 16	0.33 3	1.18 17	0.28 6	0.12 15	0.12 6	0.18 14	0.44 3	0.91 5	0.61 5
IROF++ [61]	9.8	0.08 4	0.23 9	0.07 9	0.21 18	0.68 16	0.17 19	0.28 10	0.63 8	0.19 18	0.15 8	0.73 10	0.09 13	0.60 3	0.89 4	0.42 5	0.43 13	1.08 11	0.31 14	0.10 4	0.12 6	0.12 4	0.47 7	0.98 9	0.68 13
⋮	⋮																								
TV-L1-improved [17]	37.6	0.09 17	0.26 25	0.07 9	0.20 15	0.71 20	0.16 16	0.53 40	1.18 46	0.22 34	0.21 39	1.24 47	0.11 29	0.90 37	1.31 42	0.72 31	1.51 61	1.93 57	0.84 57	0.18 56	0.17 56	0.31 58	0.73 36	1.62 41	0.87 33
⋮	⋮																								
Brox et al. [5]	40.5	0.11 33	0.32 43	0.11 49	0.27 42	0.93 43	0.22 45	0.39 30	0.94 29	0.24 38	0.24 41	1.25 48	0.13 36	1.10 59	1.39 54	1.43 66	0.89 47	1.77 52	0.55 44	0.10 4	0.13 20	0.11 2	0.91 47	1.83 51	1.13 49

Figure 2. Snapshot of *Average Endpoint Error* (AEE) in *Middlebury* Evaluation (Captured on October 2^{nd}, 2012). Our proposed method is *LME* with automatic Edge-Aware mesh initialization.

(a) Ground Truth (b) Closeup (c) EA. Mesh (d) Manual Mesh

Figure 3. Visual comparison of our method on the *Army* and *Mequon* sequences of *Middlebury* dataset using different mesh initialization schemes, i.e. (c) Automatic Edge-Aware Meshing and (d) Manual Segmented Mesh.

Gradients with 45 iterations. The other parameter setting can be found in the evaluation section (Sec. 4).

4. Evaluation

In this section we evaluate the performance of our approach and compare its performance with existing state-of-the-art techniques. We use quantitative metrics to demonstrate the performance of our approach against the highest performing existing methods on the *Middlebury* dataset [2] and a synthetic benchmark dataset with ground truth introduced by Garg *et al.* [7]. As our method is designed to be particularly suitable for non-rigid scenarios, we therefore compare our approach against a number of the best performing (state-of-the-art) publicly available non-rigid optical flow algorithms, details are as below: The first non-rigid algorithm we have used for comparison is Garg *et al.*'s spatiotemporal optical flow algorithm. Their approach exploits correlations between 2D trajectories of neighboring pixels to improve optical flow estimation. In addition, We compare our results with the *Improved TV-L1* (ITV-L1) algorithm [15] and Brox *et al.* [5]. The former has a similar optimization framework and preprocessing steps to that of Garg *et al.* and ranks in the reasonable midfield of the *Middlebury* evaluation based on overall average. The latter is proposed by Brox *et al.* to overcome the issues caused by large pixel displacements with the help of integrating the image pyramid and warping technique in a variational model. Finally, We compare our method with the state-of-the-art keypoint-based non-rigid image registration method proposed by Pizarro *et al.* [12].

In summary, our results show that the *Laplacian Mesh Energy* greatly improves algorithm performance while our algorithm outperforms all publicly available non-rigid optical flow techniques. It also performs in the top tier of all the *Middlebury* criteria, and strongly overall - especially compared to the aforementioned specialist non-rigid optical flow techniques.

4.1. Middlebury Dataset

We first performed an evaluation on the *Middlebury* benchmark dataset using default parameter setting as follows: $\theta = 0.6$, $\lambda = 0.6$ and $\xi = 0.75$ are set for the energy function $E(\mathbf{w})$ while $\eta = 0.25$ is applied in vector displacement candidate selection (Sec. 3.2). These parameter setting remains consistent in all experiments in our paper.

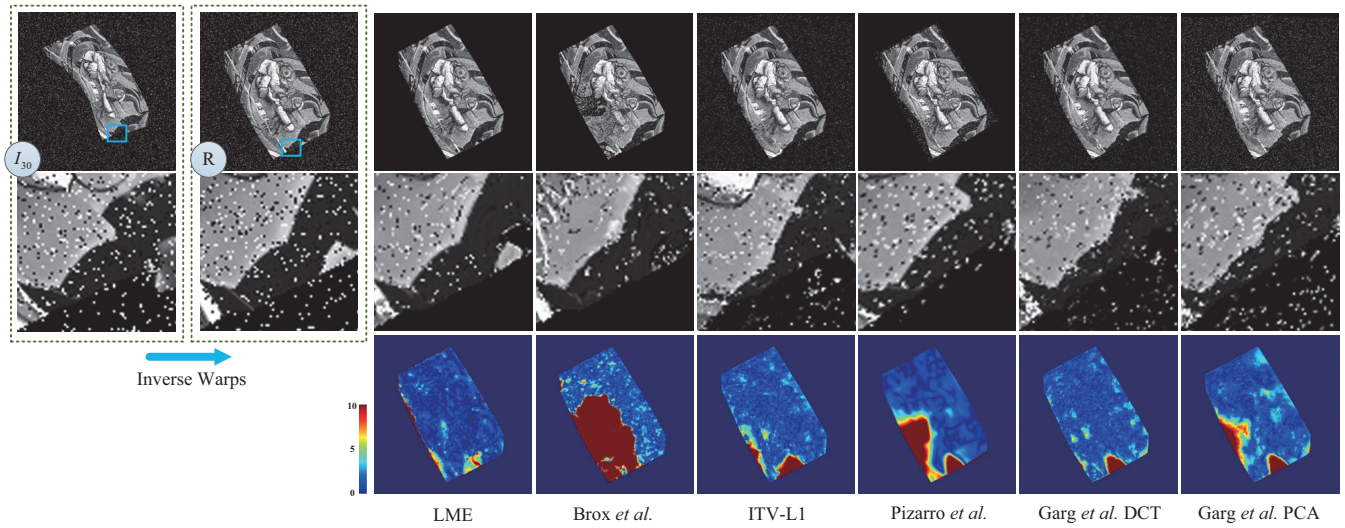
As shown in Figure 2, Our implementations is denoted by *LME* with the automatic *Edge-Aware Mesh Initialization* (Sec. 3.1). We observe that *LME* ranks among the top three algorithms and significantly outperforms most methods in the *Average Endpoint Error* (AEE) test with an overall average rank 9.7. Moreover, our implementation provides sharp flow estimation on the boundaries against the relative motion on sequence *Mequon* which contains multiple objects and a non-rigid deformable background (Figure 3). However, *Middlebury* results against other non-rigid approaches (Garg *et al.*'s method and Pizarro *et al.*'s) are not available. We therefore compare our approach against theirs using a specific non-rigid ground truth dataset (Sec. 4.2). Our approach also ranks in the top three overall for the *Average Normalized Interpolation Error* (ANIE) test which represents the quality of local image detail during the warping. Particularly strong performance is observed on *Middlebury* sequences captured using the high-speed camera – *Backyard*, *Basketball*, *Dumptruck* and *Evergreen*. Moreover, Figure 3 shows the visual comparison of our method on *Army* and *Mequon* by varying the meshing strategies w.r.t. the automatic *Edge-Aware Mesh* and the manually segmented mesh. The latter provides the sharper motion boundary.

4.2. MOCAP Benchmark Dataset

In this section we compare against a recently popular optical flow dataset specifically designed for non-rigid evaluation. In order to quantitatively evaluate their optical flow algorithm, Garg *et al.* proposed benchmark sequences

Methods	Average Endpoint Error				R 1.0 Endpoint Error				A 75 Endpoint Error			
	Original	Occlusion	Guass.N	S&P.N	Original	Occlusion	Guass.N	S&P.N	Original	Occlusion	Guass.N	S&P.N
Ours, LME	**0.39**	**0.65**	1.20	**0.87**	**0.04**	**0.06**	**0.24**	**0.19**	**0.37**	**0.39**	**0.97**	**0.83**
Garg *et al.*, PCA [7]	0.58	0.70	1.62	1.20	0.12	0.16	0.61	0.41	0.69	0.77	1.98	1.42
Garg *et al.*, DCT [7]	0.57	0.73	1.85	1.52	0.11	0.14	0.68	0.52	0.63	0.69	2.19	1.81
Pizarro *et al.* [12]	0.76	0.78	**0.95**	0.95	0.2	0.21	0.24	0.24	0.88	0.91	0.97	0.97
ITV-L1 [15]	0.56	0.69	1.81	1.37	0.09	0.11	0.68	0.45	0.50	0.53	2.23	1.58
Brox *et al.* [5]	12.62	13.55	13.73	13.32	0.28	0.32	0.72	0.69	1.83	9.38	4.99	4.52

(a) Endpoint error comparison of different methods on Garg *et al.* benchmark dataset [7].



(b) Visual comparison on the alignment from the frame 30 to the reference frame in the sequence *S&P.Noise*.

Figure 4. Quantitative analysis (*Endpoint Error*) and the visual comparison on the Garg *et al.* benchmark dataset [7]. (a): *Average Endpoint Error* and two robustness tests (R 1.0 and A75 [2]) are applied on results by varying methods. (b): **Top-left Boxes**: those include the chosen frame, the reference frame and their closed up. **The Rest**: the first row is the alignment results; the second row is the closeups; the third row is the error map against the ground truth flow field.

accompanying with ground truth [7]. A continuous dense 3D surface is obtained by interpolating sparse motion capture (MOCAP) data from real deformations of a waving flag [16]. They then project the dense textured 3D surface synthetically onto the image plane resulting in a sequence of 60 images (500×500 pixel dimension) along with optical flow ground truth motion. Our evaluation is performed on both the original captured sequence and three other degraded sequences from the Garg *et al.* benchmark dataset, which includes: **Synthetic occlusions** – Two black dots with radius of 20 pixels orbit the deformable object. **Gaussian noise** – Added with standard deviation of 0.2 relative to the range of image gray value intensities. **Salt & pepper noise** – Added with a density of 10%.

When comparing against the other methods, we use the same parameters cited by other authors. That is, for both Garg *et al.* and ITV-L1, the weights α and β are set to 30 and 2 respectively; and we also use 5 warp iterations and 20 alternation iterations [15]. According to parameter setting in [7], *Principal Components Analysis* (PCA) and *Discrete Cosine Transform* (DCT) are used for the 2D trajectory motion basis of Garg *et al.*. In addition, Brox *et al.* [5] is applied with their default parameter setting.

Figure 4(a) shows AEE (in pixel) on the four benchmark

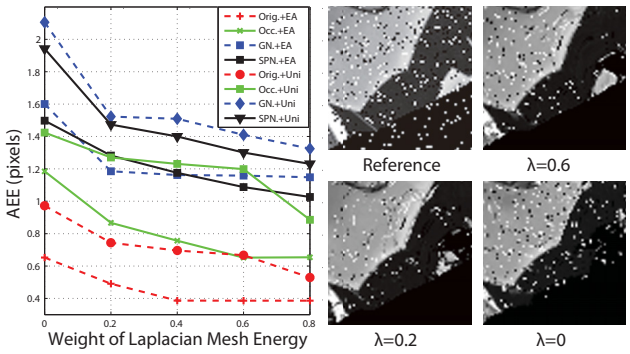


Figure 5. AEE measures on Garg *et al.* [7] benchmark sequences by varying the weighting λ (Edge-Aware (+EA) v.s. Uniform (+Uni) meshes). **Right**: Visual comparison of *LME + Edge-Aware* mesh on alignment from frame 30 to a reference in the sequence *S&P.Noise* by varying the weight λ.

sequences of Garg *et al.*. *LME* displays the best AEE measurements on the *Original*, *Occlusion* and *S&P.Noise* sequences and outperforms Garg *et al.* (both PCA and DCT basis), ITV-L1 and Brox *et al.* algorithms on all four sequences. Pizarro *et al.* has comparable performance (slightly outperforming us by 0.25 RMS) to our method on the *Guass.Noise* sequence.

In addition, we compute two robustness comparisons (R

1.0 and A 75) using identical approaches to those in [2]. *LME* yields the best performance in both R 1.0 and A 75 tests on all trials. We also observe that *LME* obtains the same score as Pizarro *et al.* and yields comparable performance over the other methods on the *Guass.Noise* sequence. We believe that this is because the large amount of Gaussian noise weakens anchor control points in the *Detail-Aware Flow Field Enhancement* step (Sec. 3.2): the accuracy of SIFT feature detection and matching is thus reduced. This issue may cause inaccurate deformation of mesh $\mathcal{M}_2$ which could result in incorrect energy calculation within the function $E(w)$. One possible solution to this would be to use features more robust against noise or to use a low pass filter, which is left for future work.

Figure 4(b) shows comparative *Inverse Image Warping* results between *LME* and five other state-of-the-art algorithms on the Garg *et al.* *Original* and *S&P.Noise* benchmark sequences. Examination of the images illustrates that *Laplacian Mesh Energy* can generate a sharper and less distorted image after warping. This provides some insight into the algorithm's strong performance in the *Middlebury* interpolation result, as images warped using our computed flow appear to preserve local visual detail.

We also evaluate the effect of varying the weight of the *Laplacian Mesh Energy* on the Garg *et al.* dataset where λ is varied with discrete values between 0 and 1. As shown in Figure 5, AEE is improved as the value λ increases on all trails. We observe that even provided with a small weight (e.g. 0.2), *Laplacian Mesh Energy* still contributes a stronger preservation of the local flow structure and hence better preserved image detail during warping. We also demonstrate how different input meshes may affect performance. Figure 5 shows a quantitative analysis of our method using *Edge-Aware* meshing and a uniform grid mesh (5-pixel vertex distances) on the Garg *et al.* dataset. The former outperforms the uniform grid mesh in all four trails.

5. Conclusion

In this paper we have presented a novel optical flow approach which uses *Laplacian Mesh Energy* to preserve local continuity of optical flow estimated on non-rigid deformations. Adapted from computer graphics, our novel energy achieves this property by minimizing differentials in Laplacian coordinates. In our evaluation we have compared our method to several state-of-the-art optical flow approaches on two well known evaluation sets. It has been demonstrated that our algorithm is capable of providing accurate flow estimation and also preserving local image detail – evident through high scores in Middlebury evaluation, and comparison to Garg *et al.*. For future work we are interested in more intelligently creating the underlying mesh to better approximate the image of interest.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. on PAMI*, 34(11):2274–2282, 2012. 3

[2] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 92:1–31, 2011. 1, 6, 7, 8

[3] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. A. Beardsley, C. Gotsman, R. W. Sumner, and M. H. Gross. High-quality passive facial performance capture using anchor frames. *ACM Trans. Graph.*, 30(4):75, 2011. 4

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on PAMI*, 23(11):1222–1239, 2001. 1

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. of ECCV*, pages 25–36, 2004. 1, 2, 3, 5, 6, 7

[6] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005. 1

[7] R. Garg, A. Roussos, and L. Agapito. Robust trajectory-space tv-l1 optical flow for non-rigid sequences. In *Proc. of EMMCVPR*, pages 300–314. Springer, 2011. 1, 6, 7

[8] B. Glocker, T. Heibel, N. Navab, P. Kohli, and C. Rother. Triangleflow: Optical flow with triangulation-based higher-order likelihoods. In *Proc. of ECCV*, pages 272–285, 2010. 1

[9] B. Horn and B. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1, 2

[10] W. Li, D. Cosker, and M. Brown. An anchor patch optimisation framework for reducing optical flow drift in long image sequences. In *Proc. of ACCV*, 2012. 3, 4

[11] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3(7):34–57, 2002. 1, 2

[12] D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *IJCV*, 97:54–70, 2012. 1, 6, 7

[13] O. Sorkine. Laplacian mesh processing. In *Eurographics State-of-the-Art Report*, pages 53–70, 2005. 1, 2, 4

[14] W. Trobin, T. Pock, D. Cremers, and H. Bischof. Continuous energy minimization via repeated binary fusion. In *Proc. of ECCV*, pages 677–690, 2008. 1

[15] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l 1 optical flow. *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45, 2009. 6, 7

[16] R. White, K. Crane, and D. Forsyth. Capturing and animating occluded cloth. *ACM Trans. on Graphics (TOG)*, 26(3):34, 2007. 7

[17] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. on PAMI*, 34(9):1744–1757, 2012. 4

[18] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H. Seidel. Complementary optic flow. In *Proc. of EMMCVPR*, pages 207–220, 2009. 1