



HAL
open science

Surface Flow from Visual Cues

Benjamin Petit, Antoine Letouzey, Edmond Boyer, Jean-Sébastien Franco

► **To cite this version:**

Benjamin Petit, Antoine Letouzey, Edmond Boyer, Jean-Sébastien Franco. Surface Flow from Visual Cues. VMV 2011 - Vision, Modeling and Visualization Workshop, Oct 2011, Berlin, Germany. pp.1-8, 10.2312/PE/VMV/VMV11/001-008 . inria-00631476

HAL Id: inria-00631476

<https://inria.hal.science/inria-00631476v1>

Submitted on 8 Mar 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Surface Flow from Visual Cues

B. Petit¹, A. Letouzey¹, E. Boyer¹, J.-S. Franco^{1,2}

¹LJK / INRIA Rhône-Alpes
²Grenoble INP

Abstract

In this paper we study the estimation of dense, instantaneous 3D motion fields over a non-rigidly moving surface observed by multi-camera systems. The motivation arises from multi-camera applications that require motion information, for arbitrary subjects, in order to perform tasks such as surface tracking or segmentation. To this aim, we present a novel framework that allows to efficiently compute dense 3D displacement fields using low level visual cues and geometric constraints. The main contribution is a unified framework that combines flow constraints for small displacements with temporal feature constraints for large displacements and fuses them over the surface using local rigidity constraints. The resulting linear optimization problem allows for variational solutions and fast implementations. Experiments conducted on synthetic and real data demonstrate the respective roles of flow and feature constraints as well as their ability to provide robust surface motion cues when combined.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Computer Graphics]: Scene Analysis — Motion

1. Introduction

Recovering dense motion information is a fundamental intermediate step in the image processing chain upon which higher level applications can be built, such as tracking or segmentation. For that purpose, pixel observations in the image provide useful motion cues through temporal variations of the intensity function. In the monocular case these variations allow us to recover a dense 2D motion field in the image: the *optical flow*. The estimation of the optical flow has been a subject of interest in the vision community for decades and numerous methods [BFB94, HS81, LK81] have been proposed. In the multiocular case, the integration over different viewpoints allow us to consider 3D motions of points on the observed surfaces and to estimate dense 3D vector fields: the *scene flow* [VBR*05, NA02]. However, in both 2D and 3D cases, the motion information cannot be determined independently at a point from intensity variations only, and so additional constraints between points must be introduced, smoothness for example. Moreover, as a result of finite difference approximations of derivatives, flow estimates are known to be limited to small motions. While several approaches have been proposed in 2D to cope with these limitations [XJM10], less effort has been devoted to the 3D case.

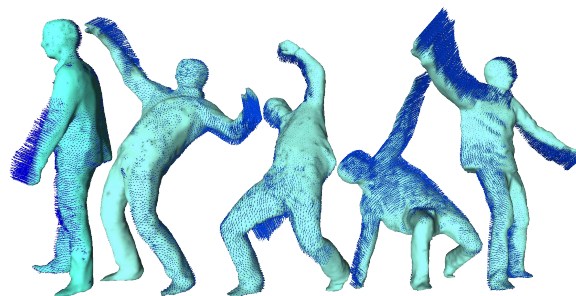


Figure 1: Example of dense scene flow (in blue) from sparse 2D and 3D features and dense normal flow constraints (as for the rest of the paper, figures are best viewed in color).

In this paper we study how to incorporate, in an efficient way, various constraints when estimating dense motion information over 3D surfaces from temporal variations of the intensity function in several images. Our primary motivation is to provide robust motion cues that can be directly used by an application, e.g. interactive applications, or that can be fed into more advanced tasks such as surface tracking or segmentation, e.g. into rigid parts. The approach is however not limited to a specific scenario and applies to any application

that can benefit from low-level motion information. Most existing approaches that estimate scene flow assume small motions between time instants, for which finite difference approximations of temporal derivatives are valid. However this assumption is often violated with actual acquisition systems and real moving objects. In addition, flow constraints are usually plugged into specific resolution schemes that are not necessarily grounded on physical principles nor easily allow for constraints of different types to be taken into account.

We propose a unified framework that links visual constraints from consecutive images with surface deformation constraints in a consistent way. In order to handle large motions, it allows local temporal matching constraints, which are obtained from image features. Such features act as anchor points in surface regions with larger displacements and where pixel intensity variations are not informative. All visual constraints are *diffused* over the surface through a Laplacian scheme that regularizes the estimated motion vectors between neighboring surface points. A key feature of the proposed framework is that it leads to linear optimization problem, which can be solved efficiently.

The remainder of this paper is organized as follows. Section 4 presents the visual constraints obtained from consecutive images. Section 5 explains how to integrate these constraints over the surface. Experimental results on both synthetic and real data are then presented in section 6 before discussing the approach in section 7.

2. Related Work

In a seminal work on scene flow, Vedula *et al.* [VBR*05] explicated the normal flow constraint that links the intensity function derivatives in images to the scene flow of 3D points. As mentioned before, such constraints do not allow us to estimate the scene flow independently at a surface point and additional constraints must be introduced. Instead of using the normal flow constraint, an algorithm is proposed that linearly estimates the scene flow given the surface geometry and 2D optical flows. Optical flow better constrains the scene flow than the normal flow, however their estimation is based on smoothness assumptions that seldom hold in the image planes, whereas they often do on surfaces.

In [NA02], Neumann and Aloimonos introduced an elegant subdivision surface model that allows normal flow constraints to be integrated over the surface with regularization constraints. Nevertheless, this global solution still assumes small motions and can hardly deal with challenging datasets as used in this paper.

Another strategy is followed by Pons *et al.* [PKF05] who presented a variational framework that optimizes a photo-consistency criterion instead of the normal flow constraints. The interest is that both spatial and temporal consistency can be enforced but at the price of a computationally expensive

optimization. In contrast, our focus is not on shape optimization but more on providing low level motion information in an efficient way. Several works [ZK01, IM06, WRV*08] consider the case where the scene structure is described by stereo disparities and propose the combined estimation of spatial disparity and temporal 3D motion. We consider a different situation where the shape surface is given, e.g. a mesh obtained using a multi-view approach, thus allowing for a regularization of the motion field over a domain where smoothness assumptions hold.

It is worth also mentioning recent approaches on temporal surface tracking [SH07b, VZBH08, NTR*08, CBI10] that can also provide velocity fields as a by-product of the matching between consecutive frames. Our purpose is different since our method does not make any assumption on the observed shape, and only weak assumptions on the deformation model in the form of local smoothness assumptions. It provides information at a lower level, instantaneous motion, that can in turn be used as input data by a surface tracking or matching approach.

Our contributions with respect to the aforementioned approaches are twofold: (i) Following works on robust optical flow estimation [LYT*08, XJM10], we take advantage of robust initial displacement values as provided by image features tracked over consecutive time instants. Such features allow for large surface motions while normal flow constraints better model small motions. (ii) A linear framework that combines visual constraints with surface deformation constraints and allows for iterative resolutions (variational approach) as well as coarse to fine refinement.

3. Preliminaries and Definitions

Our method deals with the output of any multi-camera system capable of producing a stream of non-rigidly moving surfaces, each independently reconstructed from a set of N calibrated views, using a 3D reconstruction technique such as [FB08] or [FP06].

The surface at time t is denoted $S^t \subset \mathbb{R}^3$ and associated with the set of images $\mathcal{I} = \{\mathbf{I}_c^t \mid c \in [1..N]\}$. A 3D point \mathbf{P} on the surface is described by the 3D vector $(x, y, z)^T \in \mathbb{R}^3$. Its projection in the image \mathbf{I}_c^t is the 2D image point \mathbf{p}_c with coordinates $(u_c, v_c)^T \in \mathbb{R}^2$ computed using the 3x4 projection matrix $\Pi_c : \mathbb{R}^3 \mapsto \mathbb{R}^2$ of camera c (see figure 2). The 2D image region corresponding to the visibility of S^t in \mathbf{I}_c^t is denoted by $\Omega_c^t = \Pi_c S^t$.

Our method is looking for the 3D motion field of the surface between time t and $t+1$ described by $V^t : S^t \mapsto \mathbb{R}^3$ with $V^t(\mathbf{P}) = \frac{d\mathbf{P}}{dt} \forall \mathbf{P} \in S^t$. This motion field is constrained by: (i) the input data, i.e. the set of calibrated images \mathcal{I}^t and \mathcal{I}^{t+1} and the surface S^{t+1} and (ii) a deformation model.

The projection of the 3D motion field on \mathbf{I}_c^t is denoted by v_c^t . The relation between a small displacement on the surface

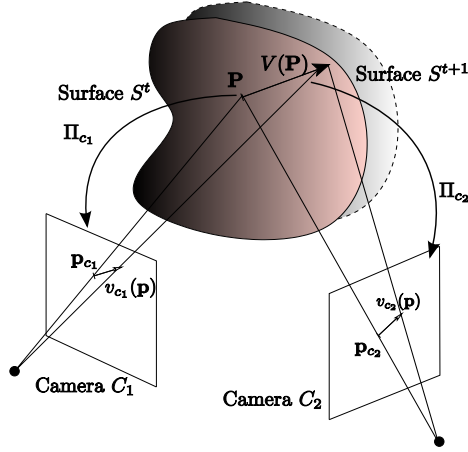


Figure 2: Projection from scene flow $V(P)$ into optical flows v_c^t in different images of a multi-camera system.

S^t and its image taken by the camera c is described by the 2×3 jacobian matrix $J_{\Pi_c}(\mathbf{p}_c) = \frac{\partial \mathbf{p}_c}{\partial \mathbf{p}}$ such that $v_c^t = J_{\Pi_c}(\mathbf{p}_c)V^t$.

4. Visual Constraints

Our method can use three types of visual constraints to estimate 3D displacements:

1. Dense image flow constraints.
2. Sparse 2D features correspondences.
3. Sparse 3D features correspondences.

Each of these constraints will lead to a term in the error functional (see section 5.2), describing how the computed 3D motion field relates to the observations. Notice that we do not include spatial or temporal photo-consistency constraints as they yield non-linear terms in the error and better suited to shape optimization problems than to direct low level motion cue estimation.

4.1. Dense 2D Normal Flow

Dense information on V^t can be classically obtained using the 2D optical flow information available in the images. Indeed, assuming brightness constancy between \mathbf{p}_c^{t+1} and \mathbf{p}_c^t , projection of the same surface point on two consecutive frames, one can write the *Normal Flow Equation* [BFB94] as:

$$\begin{aligned} \nabla I_c^t \cdot v_c^t + \frac{dI_c^t}{dt} &= 0, \\ \text{or } \nabla I_c^t \cdot [J_{\Pi_c} V^t] + \frac{dI_c^t}{dt} &= 0, \end{aligned}$$

as expressed from 3D surface velocities [VBR*05]; ∇I_c^t is the spatial gradient of the image intensity and $\frac{dI_c^t}{dt}$ is the temporal gradient of image intensity. We can then define an er-

ror term measuring the discrepancy between the computed 2D motion field v_c^t and the normal flow constraints:

$$\mathbf{E}_{flow} = \sum_{c=1}^N \int_{\Omega_c^t} \|\nabla I_c^t \cdot [J_{\Pi_c} V^t] + \frac{dI_c^t}{dt}\|^2 d\mathbf{p}_c. \quad (1)$$

This term is the most common among scene flow methods and well suited for small image displacements, but has important limitations: it only constrains the image displacements in the direction of the image gradient ∇I_c^t , or the normal component of the optical flow. This is the *aperture problem* in 2D that extends to 3D as will be discussed in 5. Also, linearization based on the image gradient is typically invalid for large displacements.

4.2. Sparse 2D Features

In some situations, e.g. slow motion or high frame rates, motion field recovery can rely on dense normal flow constraints alone. However, in a more general context, additional constraints must be considered. To this purpose, we propose the use of sparse 2D correspondences between the set of images \mathcal{I}^t and \mathcal{I}^{t+1} as 2D anchor points to guide the flow estimation. Such features are easily obtained using one of various popular techniques, e.g. SIFT [Low04]. Importantly, we choose to match features among subsequent frames of the same camera and not between views. This is for two reasons: First, this eliminates any need for inter-camera exposure and color calibration. More importantly, the inlier/outlier ratio between such images are substantially more favorable than for inter-camera matching. This is especially true for the challenging data targeted: general subjects with low-to-average texture, and object-centered setups exhibiting wide baselines by nature. Any remaining outliers can thus be easily eliminated using a conservative matching threshold, as validated in our experiments.

We compute SIFT descriptors for \mathcal{I}^t and \mathcal{I}^{t+1} , then match features between \mathbf{I}_c^t and \mathbf{I}_c^{t+1} , with $c \in [1..N]$. This yields a set of sparse 2D displacements $v_{c,s}^t$ for some 2D points $\mathbf{p}_{c,s} \in \Omega_c^t$, those points form a subset of Ω_c^t called $\Omega_{c,s}^t$ (see figure 3). The following error term measures the discrepancy between the computed 2D motion field v_c^t and the sparse 2D displacements $v_{c,s}^t$:

$$\begin{aligned} \mathbf{E}_{2D} &= \sum_{c=1}^N \sum_{\Omega_{c,s}^t} \|v_c^t - v_{c,s}^t\|^2, \text{ or} \\ \mathbf{E}_{2D} &= \sum_{c=1}^N \sum_{\Omega_{c,s}^t} \|J_{\Pi_c} V^t - v_{c,s}^t\|^2, \end{aligned} \quad (2)$$

where (2) is the linearization we use. Unlike the normal flow equation, this approximation is still valid for moderate displacements as it doesn't involve image gradients.

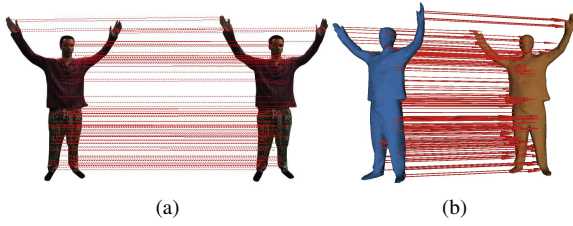


Figure 3: Example of sparse 2D features obtained from image matching (a), and 3D feature correspondences between two surfaces (b).

4.3. Sparse 3D Features

3D features can also easily be included in our framework to guide flow estimation in the presence of large displacements. They provide sparse displacement information for a set of salient 3D points lying on \mathcal{S}^t , obtained by detecting features on \mathcal{S}^t and \mathcal{S}^{t+1} and matching them across time, based on a geometric or photometric surface descriptor. These correspondences can be obtained using various recent methods, such as [SH07a], or the MeshDOG 3D feature detector and the MeshHOG descriptor [ZBVH09], and can provide complementary information to the 2D terms previously described in the form of robustness to occlusions. On the other hand, they are sensitive to different issues, such as topology changes of the observed surface, which sometimes occur in the sequence.

We have found that an interesting and efficient way to obtain 3D features is to back-project matching 2D feature correspondences between \mathcal{I}^t and \mathcal{I}^{t+1} onto their respective surfaces \mathcal{S}^t and \mathcal{S}^{t+1} . This yields a 3D point-pair whose match was based on intra-view 2D SIFT. This is not entirely equivalent to the sparse 2D feature term previously proposed, as it assumes availability of \mathcal{S}^{t+1} , whereas the previous method could be used without \mathcal{S}^{t+1} , if required by the application. Also this type of match could be influenced by the error in surface estimation, dependent on the reconstruction method used. The advantage in having 3D constraints is that the term is valid for arbitrarily large displacements as it doesn't involve linearization. We have found this scheme to work well in practice and use it in stages of the final algorithm described in section 5.3.

Regardless of how they are obtained, let V_m^t be the displacements of the detected feature points $\mathbf{P}_m \in \mathcal{S}^t$ (see figure 3). These points form a discrete subset of \mathcal{S}^t called \mathcal{S}_m^t . Being measured directly as a 3D distance, the error between the computed 3D motion field V^t and the target 3D displacements V_m^t can be written without linearization:

$$\mathbf{E}_{3D} = \sum_{\mathcal{S}_m^t} \|V^t - V_m^t\|^2. \quad (3)$$

5. Regularization

The sparse set of 2D and 3D correspondences only constrains the displacement of the surface for specific 3D points and for their re-projection on the images. To find a dense motion field over the surface we need to propagate those constraints through a regularization term.

Furthermore, as mentioned earlier, dense 2D normal flow constraints do not provide enough information to estimate 3D displacements. In fact it can be shown that the normal flow equations at different image projections of a 3D point \mathbf{P} are linearly dependent, and can only be used to solve for 2 of the 3 dofs. Vedula *et al.* [VBR*05] mentioned two regularization strategies to cope with this limitation. The regularization can be performed in the image planes to estimate optical flows which provide then full constraints on the scene flow, or the regularization can be performed on the 3D surface.

Since we are given the 3D surface and that sparse constraints from 2D or 3D features need to be integrated, a natural choice in our context is to regularize in 3D. In addition regularization in the image space suffers from artifacts and incoherences resulting from depth discontinuities and occlusions that contradict the smoothness assumption whereas such assumption holds on the 3D surface.

5.1. Deformation Model

Smoothness assumptions on 3D displacement fields over a surface constrain the surface deformations locally. They thus define a deformation model of the surface, e.g. local rigidity. In 2D, numerous regularization schemes have been proposed for the optical flow estimation that fall into 2 main categories: local and global regularizations. They can be extended to 3D. For example, the 2D Lucas and Kanade method, which uses a local spatial neighborhood, was applied in 3D by Devernay *et al.* [DMG06]. However, the associated deformation model of the surface has no real meaning since deformation constraints only propagate locally, yielding inconsistencies between neighborhoods. On the other hand, the global strategy introduced by Horn and Schunck [HS81] is well suited to our context. Though less robust to noise than local methods such as Lucas-Kanade, it allows sparse constraint propagation over the whole surface. In addition the associated surface deformation model has proved to be efficient in the computer graphics domain [SA07].

The extension of Horn and Schunck deformation model to 3D points is described by the following error function which enforces a local rigidity of the motion field:

$$\mathbf{E}_d = \int_{\mathcal{S}} \|\nabla V\|^2 d\mathbf{P}. \quad (4)$$

5.2. Energy Functional Minimization

We find the best displacement that satisfies all the aforementioned constraints by minimizing the following error functional:

$$\arg \min_V \left[\lambda_{3D}^2 \mathbf{E}_{3D} + \lambda_{2D}^2 \mathbf{E}_{2D} + \lambda_{flow}^2 \mathbf{E}_{flow} + \lambda_d^2 \mathbf{E}_d \right],$$

where the different λ coefficients are parameters that can be set to give more weight to a particular constraint.

This functional can be minimized by solving its associated Euler-Lagrange equation:

$$\sum_{c=1}^N \left[\lambda_{flow}^2 \left[\nabla_{I_c}^t \cdot [J_{\Pi_c} V^t] + \frac{dI_c^t}{dt} \right] + \lambda_{2D}^2 \delta_{\Omega_{c,s}^t} J_{\Pi_c} [V^t - V_{c,s}^t] \right] + \lambda_{3D}^2 \delta_{S_m^t} [V^t - V_m^t] + \lambda_d^2 \nabla^2 V^t = 0, \quad (5)$$

where δ is the Kronecker symbol, denoting that this constraint is only defined for 3D points in S_m^t or $\Omega_{c,s}^t$.

The discretized Euler-Lagrange equation for each 3D points \mathbf{P} of the surface has the form:

$$\mathbf{A}_P \mathbf{V}_P + \mathbf{b}_P - \Delta \mathbf{V}_P = 0, \quad (6)$$

where Δ is the normalized Laplace-Beltrami operator over the surface.

The combination of equation (6) for all 3D points $\mathbf{P} \in S^t$ creates a simple linear system of the form:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{A} \end{bmatrix} \mathbf{V}^t + \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \end{bmatrix} = \mathbf{0}, \quad (7)$$

where \mathbf{L} is the Laplacian matrix as defined in [SA07]. This is a sparse linear system which can be solved using a sparse solver such as *Taucs*.

Note that, interestingly, this formulation revisits the Laplacian mesh editing in an *as-rigid-as-possible* way of the computer graphics community [SA07]. While the deformation model is similar, the difference lies in the constraint used: anchor points in [SA07] and visual constraints in our approach. In both cases, it is known that this deformation model does not handle explicitly rotations of the surface. Although this is an issue when deforming the surface under a small number of constraints, as usual in graphic applications, the density of the normal flow constraints in our case helps to recover rotations without the need for nonlinear optimizations.

Equation (5) can also be solved iteratively using the Jacobi method applied to this large sparse system. In this case one could solve the linear system for each point independently, and repeat the process iteratively using the updated solution of the neighborhood points. This variational approach also allows for coarse to fine refinements.

5.3. Selection of Weights and 2-Pass Refinement

In equation (5), the parameters λ_{2D} , λ_{3D} , λ_{flow} and λ_d indicate the strengths of 2D and 3D features, 2D normal flow constraints and the Laplacian respectively. High values of the parameters imply that the influence of each of the respective components is larger.

In our context, similarly to [XJM10] in 2D, we trust our 2D and 3D features to be robust even under wide displacements, while we know that the 2D flow constraints are not reliable when the reprojected displacement is greater than a few pixels on the images. Consequently, we propose a method that performs two consecutive minimizations of the energy functional using two sets of weights. The steps of the corresponding algorithm are as follows:

1. We start by computing the sparse 2D and 3D feature correspondences between S^t and S^{t+1} and between \mathcal{I}^t and \mathcal{I}^{t+1} . We also compute the Laplacian matrix \mathbf{L} of our discretized surface.
2. We solve equation (7), with $\lambda_{flow} = 0$ and high values of λ_{3D} and λ_{2D} with respect to λ_d . We obtain a first approximation of V^t denoted V'' which recovers wide displacements on the surface.
3. We create a deformed surface $S'' = S^t + V''$ that we reproject in all cameras using the initial texture information coming from the projection of \mathcal{I}^t on S^t . We obtain a new set of images \mathcal{I}'' .
4. We compute the visibility of the surface S'' on each camera and the dense normal flow constraints between \mathcal{I}'' and \mathcal{I}^{t+1} for each visible points. We thus have several constraints by sampled points of the surface.
5. As in step 2., we solve equation (7) using the flow computed in step 4. and the 2D and 3D features previously computed in step 1. as anchor points. For this step we use high values for λ_{3D} and λ_{2D} and lower values for λ_{flow} and λ_d . We obtain the displacement between S'' and S^{t+1} denoted V''' and thus a refined version of $V^t = V'' + V'''$. This second step allows us to recover smaller displacements, which is handled well by the flow constraint.

We see from our results that, in practice our approach can handle both large and small displacements. This is because we use sparse features to attain large displacements, and the normal flow to recover the details.

6. Evaluation

For our evaluation we used both synthetic and real data:

1. Synthetic data were obtained by deforming a model over time to create sequences. We rasterized this sequences into virtual cameras of resolution 1 MPixels, distributed on a sphere around the models. We used two different models and camera setups to create different sequences. (i) A triangular mesh with 7k vertices representing an articulated human model, deformed to generate a sequence

- of 200 frames viewed by ten cameras. (ii) A rigidly moving sphere model of 640 vertices viewed by 34 cameras, used for quantitative evaluation and comparison.
- Real data are taken from publicly available datasets (or soon to be). We present experiments on the *flashkick* sequence from the *SurfCap* project [SH07b] of the University of Surrey. This sequence uses height 2-MPixels cameras, and produces smooth meshes of $\sim 140k$ vertices. The other sequences were taken from 32 2-MPixels cameras and provide visual-hull based meshes of $\sim 10k$ vertices.

6.1. Quantitative Evaluation on Synthetic Data

Using the algorithm described in section 5.3 we computed the motion fields on the synthetic dance sequence. Figure 4-a) shows the motion field on one frame of the sequence. Red vectors denote the initial sparse 3D and reprojected 2D constraints while blue vectors denote the sampled 3D motion field.

Figure 4-b) shows the motion field accumulated over few frames from a top view. This result can be compared to the one from Varanasi *et al.* [VZBH08], indeed their method is able to provide velocity field, albeit as a by-product of the matching between two consecutive meshes.

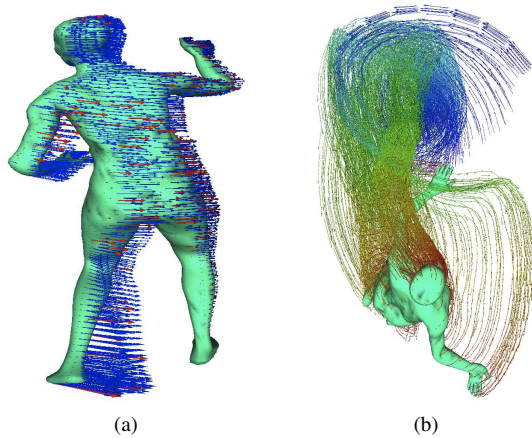


Figure 4: (a) Motion field on one frame of our synthetic dance data and (b) motion history from a top view (color indicates frame number).

Since the meshes are consistent over time, we were able to obtain the ground truth displacements and to evaluate our results quantitatively. Figure 5 shows the error on the angle of the motion vectors after each regularization step of our algorithm. We can clearly see the advantages of using the normal flow constraints to refine the motion field.

The graphs in Figure 6 show quantitative results on synthetic data. We tested our algorithm on two 15 frame sequences of a sphere seen by 34 cameras.

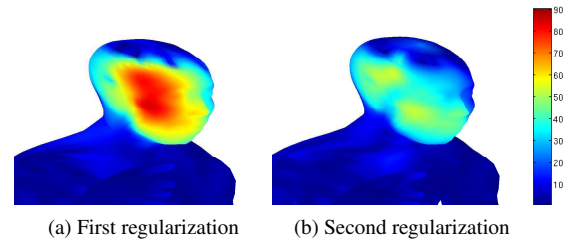


Figure 5: Close-up on the angular error, in degrees, for the dancer's face. These images show how the second step of our method helps recovering motion details.

In the first sequence the motion is a translation and in the second the sphere is rotating around its center. We can see on Figure 6 that the second regularization step (in green) always gives roughly the same level of quality increment. This is due to the fact that our first step (in red) can recover large displacement in such a way that the remaining motion is at sub-pixel level, which is exactly where normal flow information is reliable. Those graphs also show that the quality of our results does not depend on the amplitude of the motion, unlike many other methods.

We also tested our method on a second sequence with only rotational motion, with up to 12 degrees of rotation between two consecutive frames, yielding plots with strictly identical characteristics (not plotted to preserve space). Even if our deformation model does not handle explicitly rotations, as mentioned in section 5.2, we were still able to properly recover the surface motion.

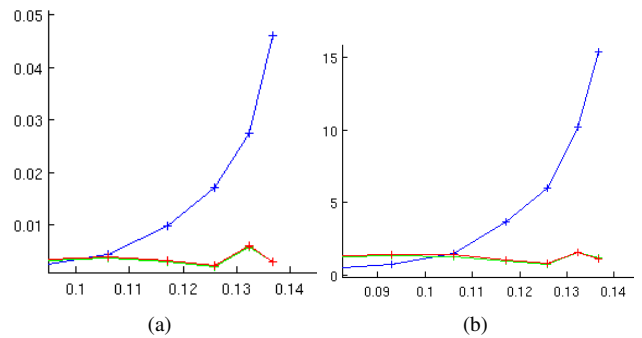


Figure 6: (a) Norm (in meters) and (b) Angular (in degrees) error of the recovered motion with respect to the amount of motion of the surface (in meters). In blue: Vedula *et al.*, in red: our method after the first regularization, and in green: after the second regularization.

6.2. Comparison

In order to compare our approach with the state of the art we implemented Vedula *et al.* method presented in [VBR*05]. Since this paper explains three different ways of computing scene flow, we implemented the one which uses the same input information as we do : "Multiple cameras, known scene geometry". We used the latest OpenCV implementation of the Lukas-Kanade optical flow computation with standard parameters and performed scene flow computation as explained. The graphs in Figure 6 shows the quality of the motion flow computed using Vedula *et al.* (in blue), and compared with our method. As expected our approach, clearly outperforms the other method as soon as the motion of the object is bigger than the pixel size in the images.

Note that the quality of our results is correlated with the resolution of the model used. While Vedula *et al.* are performing regularization in image space, we are performing our regularization on the discretized surface. Thus we could improve our results by using higher resolution models (at a higher tessellation level).

6.3. Experiments on Real Data

We computed 3D motion fields on the popular *flashkick* sequence. In this challenging sequence the subject is wearing loose clothes with poor texture information. Furthermore, the amplitude of the motion is very high between two frames. Fewer reliable 2D/3D correspondences are available, but they are essential to recover the wide displacement.

nonetheless, we were able to compute a coherent motion field on most of the frames (see Figures 7-a-b)). On a few frames where our algorithm did not find any features on the legs or feet of the dancer, the computed motion field shows the good direction but not the correct norm of the vectors. Lack of visual constraints results in incomplete first estimation of the motion field, and so the remaining displacement cannot be recovered completely by the normal flow constraints. Figure 7-c) shows a problematic frame where the motion of the right leg of the dancer is not properly computed. To visualize this error, we displayed the input surfaces at time t and $t + 1$ (respectively cyan and dark blue), while the 'flowed' surface is shown in yellow dots. Finally figure 7-d) shows the motion history over a few frames. Note that we only compute dense motion over the surface and not a deformed mesh. Thus we do not have a consistent connectivity over time and cannot perform any vertex tracking. Therefore the quantitative evaluation of the data is not possible, but visualization of the results are very satisfactory.

We also used our own sequences. One shows a subject performing a simple action, moving both hands from hips to head. The subject is wearing loose and highly textured clothes, which allows us to compute a high number of reliable 2D and 3D features, see Figure 8-a)-b) for examples of motion fields on this sequence. Figure 8-c) shows the motion

field accumulated over the whole sequence. Instantaneous motion field results are shown in Figure 1. They were computed on another of our sequences in which the subject falls and stands back up. This sequence involves large motions of the arms, which were properly recovered as shown in the motion history in Figure 8-d).

We used a naive hybrid Matlab/C++ implementation of our method and computation times are of the order of a few seconds for each frame on a typical Intel Core 2 duo computer, using a set of 32 2-MPpixels images, and meshes of 10k vertices.

7. Conclusion

We have presented a unified framework which allows us to combine various photometric constraints with the aim of computing dense motion information over a surface. This framework is based on an iterative method that allows us to handle arbitrary large displacements while still recovering small details. Experiments on real datasets demonstrate the robustness of the approach

In order to handle images with lower texture, the method could be improved by adding more constraints, for example a photometric consistency criterion such as the one used by Pons *et al.* in [PKF05]. Additional perspectives include interactive applications, such as collision-based interactions between the observed object and any virtual object, as well as real-time action recognition.

References

- [BFB94] BARRON J., FLEET D.-J., BEAUCHEMIN S.: Performance of Optical Flow Techniques. *International Journal of Computer Vision* (1994).
- [CBI10] CAGNIART C., BOYER E., ILIC S.: Probabilistic Deformable Surface Tracking From Multiple Videos. In *European Conference on Computer Vision* (2010).
- [DMG06] DEVERNAY F., MATEUS D., GUILBERT M.: Multi-Camera Scene Flow by Tracking 3-D Points and Surfels. In *Computer Vision and Pattern Recognition* (2006).
- [FB08] FRANCO J.-S., BOYER E.: Efficient Polyhedral Modeling from Silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008).
- [FP06] FURUKAWA Y., PONCE J.: Carved Visual Hulls for Image-Based Modeling. In *European Conference on Computer Vision* (2006).
- [HS81] HORN B., SCHUNCK B.: Determining Optical Flow. *Artificial Intelligence* (1981).
- [IM06] ISARD M., MACCORMICK J.: Dense Motion and Disparity Estimation via Loopy Belief Propagation. In *Asian Conference on Computer Vision* (2006).
- [LK81] LUCAS B., KANADE T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In *International Joint Conference on Artificial Intelligence* (1981).
- [Low04] LOWE D.: Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision* (2004).

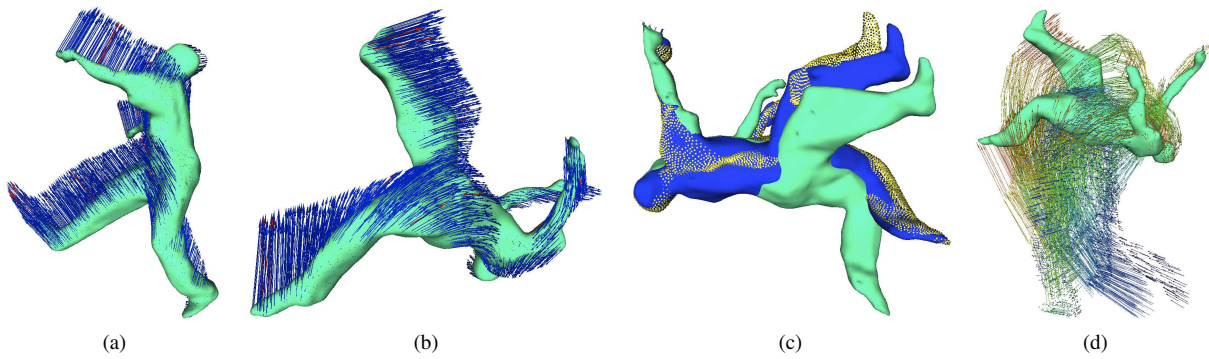


Figure 7: Motion field on challenging frames of the flashkick sequence (a) and (b), partially recovered motion (c) and motion history over this sequence (d) (color indicates frame number).

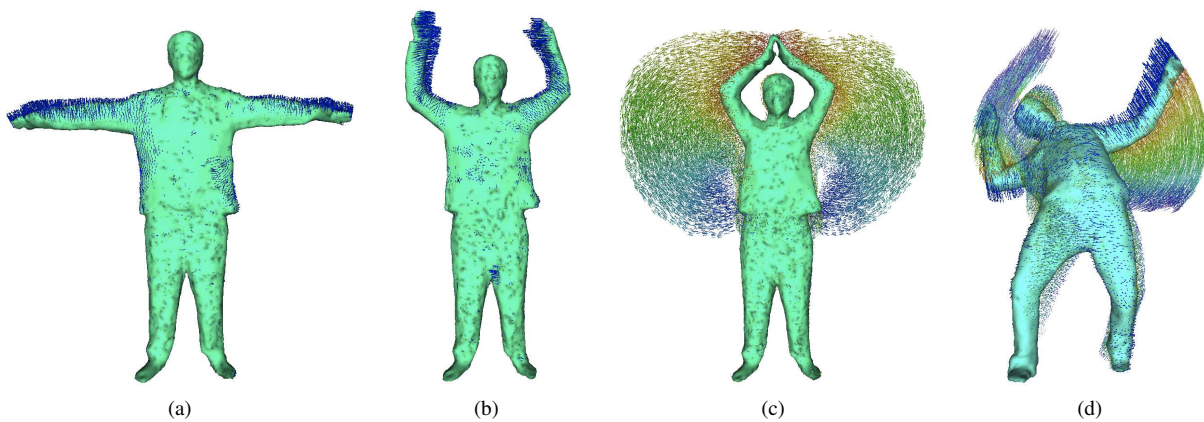


Figure 8: Motion fields on several frames of our real data (a) and (b) and motion history over the sequences (c) and (d) (color indicates frame number).

[LYT*08] LIU C., YUEN J., TORRALBA A., SIVIC J., FREEMAN W.: SIFT Flow: Dense Correspondence across Different Scenes. In *European Conference on Computer Vision* (2008).

[NA02] NEUMANN J., ALOIMONOS Y.: Spatio-Temporal Stereo Using Multi-Resolution Subdivision Surfaces. *International Journal of Computer Vision* (2002).

[NTR*08] NAVEED A., THEOBALT C., ROSSL C., THURN S., SEIDEL H.: Dense Correspondence Finding for Parametrization-free Animation Reconstruction from Video. In *Computer Vision and Pattern Recognition* (2008).

[PKF05] PONS J.-P., KERIVEN R., FAUGERAS O.: Modelling Dynamic Scenes by Registering Multi-View Image Sequences. In *Computer Vision and Pattern Recognition* (2005).

[SA07] SORKINE O., ALEXA M.: As-Rigid-As-Possible Surface Modeling. In *Eurographics Symposium on Geometry Processing* (2007).

[SH07a] STARCK J., HILTON A.: Correspondence Labeling for Wide-Timeframe Free-Form Surface Matching. In *European Conference on Computer Vision* (2007).

[SH07b] STARCK J., HILTON A.: Surface Capture for Performance-Based Animation. *IEEE Computer Graphics and Applications* (2007).

[VBR*05] VEDULA S., BAKER S., RANER P., COLLINS R., KANADE T.: Three-Dimensional Scene Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2005).

[VZBH08] VARANASI K., ZAHARESCU A., BOYER E., HORRAUD R. P.: Temporal Surface Tracking Using Mesh Evolution. In *European Conference on Computer Vision* (2008).

[WRV*08] WEDEL A., RABE C., VAUDREY T., BROX T., FRANKE U., CREMERES D.: Efficient Dense Scene Flow from Sparse or Dense Stereo Data. In *European Conference on Computer Vision* (2008).

[XJM10] XU L., JIA J., MATSUSHITA Y.: Motion Detail Preserving Optical Flow Estimation. In *Computer Vision and Pattern Recognition* (2010).

[ZBVH09] ZAHARESCU A., BOYER E., VARANASI K., HORRAUD R. P.: Surface Feature Detection and Description with Applications to Mesh Matching. In *Computer Vision and Pattern Recognition* (2009).

[ZK01] ZHANG Y., KAMBHAMETTU C.: On 3D Scene Flow and Structure Estimation. In *Computer Vision and Pattern Recognition* (2001).