Direct Estimation of Non-Rigid Registrations with Image-Based Self-Occlusion Reasoning

Accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence in October 2008

Vincent Gay-Bellile[1,2], Adrien Bartoli[1], Patrick Sayd[2]

[1] LASMEA, CNRS / Université Blaise Pascal, Clermont-Ferrand, France

[2] CEA, LIST, Boite Courrier 94, Gif-sur-Yvette, F-91191 France

Vincent.Gay-Bellile@univ-bpclermont.fr

Adrien.Bartoli@gmail.com

Abstract

The registration problem for images of a deforming surface has been well studied. External occlusions are usually well-handled. In 2D image-based registration, self-occlusions are more challenging. Consequently, the surface is usually assumed to be only slightly self-occluding.

This paper is about image-based non-rigid registration with self-occlusion reasoning. A specific framework explicitly modeling self-occlusions is proposed. It is combined with an intensity-based, 'direct' data term for registration. Self-occlusions are detected as shrinkage areas in the 2D warp.

Experimental results on several challenging datasets show that our approach successfully registers images with self-occlusions while effectively detecting the self-occluded regions.

I. INTRODUCTION

Registering monocular images of a deforming surface is important for tasks such as video augmentation by texture editing, non-rigid Structure-from-Motion and deformation capture. Image registration tries to find the geometric and photometric transformations that make two or several images of the same scene as similar as possible. The geometric transformation or 'warp' changes the pixels position while the photometric one changes their value (gray level or color).

Image registration is a difficult problem since the appearance of imaged surfaces varies due to several phenomena such as camera pose, surface deformation, lighting, motion blur, occlusions. Occlusions is one of the main issues in 2D image registration. They can be classified into external and self-occlusions. External occlusions are caused by an object entering the space between the camera and the surface of interest. Self-occlusions are due to the surface being bent in such a way that a part of it occludes another one.

There are two main registration approaches: the direct one and the feature-based one. The former uses the color discrepancy as a registration criterion, while the latter minimizes some distance between matched features (interest points, lines, etc). Dealing with extreme self-occlusions such as those of figures 1 is very challenging. The number of features might not be large enough, in particular in the neighborhood of the self-occlusion boundary, to recover the correct warp. We thus use a direct method which provides dense correspondences and is very accurate.

Deformation estimation in the presence of self-occlusions could be formulated in 3D. Detecting self-occlusions would be achieved by a z-buffer technique or as the solution of an ordinary differential equation if an implicit representation of 3D shapes were employed, see *e.g.* [19].

However, recovering a 3D surface, its deformations and the camera pose from a monocular video sequence is intrinsically ill-posed. While strong prior information can be used to disambiguate the problem, see *e.g.* [11], [29], it is common to avoid a full 3D model by using image-based deformation models, *e.g.* [6], [13], [21].

Previous work on 2D deformation estimation via image registration, *e.g.* [6], [21], [22], [24], [31], usually deals with external and self-occlusions within an outliers rejection framework, *e.g.* with an M-estimator [2], [9] wrapped around the data terms. These methods handle external occlusions and a limited amount of self-occlusion, as shown in figure 1. Lin *et al.* [22] explicitly tackle self-occlusions for tracking a Near Regular Texture (NRT). A visibility map is used for dealing with external and self-occlusions. It is based on geometric and appearance properties of NRTs. Dealing with self-occlusions in purely 2D image-based registration is a very challenging problem that has not yet received a commonly agreed solution in the scientific community.

We propose a specific framework for non-rigid registration in spite of self-occlusions. The basic idea is to consider self-occluded pixels not as outliers, but as points at which the unknown warp is surjective. This is implemented with two main components: the warp is constrained to shrink rather than to 'fold' and self-occlusions are detected as warp shrinkage areas. This enables accurate registration in spite of self-occlusions. The proposed algorithm does not require any specific pattern and is general in the sense that it only needs a warp smoothness prior, which is a quite generic assumption. This article proposes the implementation of these 'self-occlusion-capable' warps to retexture videos of deformable surfaces. Extensive experimental results on realistic videos are reported to demonstrate the efficiency of this framework to deal with self-occlusions.

Roadmap. Previous and background on direct non-rigid registration are given in §II. In §III, the problem statement and the solution we propose are presented. The detection of self-occlusions is explained in §IV. Direct registration with self-occlusion reasoning is proposed in §V. Means for realistic self-occluded surface retexturing are proposed in §VI . Experimental results on real data are reported in §VII. Finally, we give our conclusions and discuss future work in §VIII.

Notation. The images to be registered are written $\mathcal{I}_i$ with $i = 1, \ldots, n$. The template, *e.g.* the occlusion-free region of interest in the first image, is denoted $\mathcal{I}_0$. The warp is written $\mathcal{W}$. It depends on a parameter vector $\mathbf{u}_i$ for image $\mathcal{I}_i$, and maps a point $\mathbf{q}$ in the template to the corresponding point $\mathbf{q}_i$ in image i as: $\mathbf{q}_i = \mathcal{W}(\mathbf{q}; \mathbf{u}_i)$.

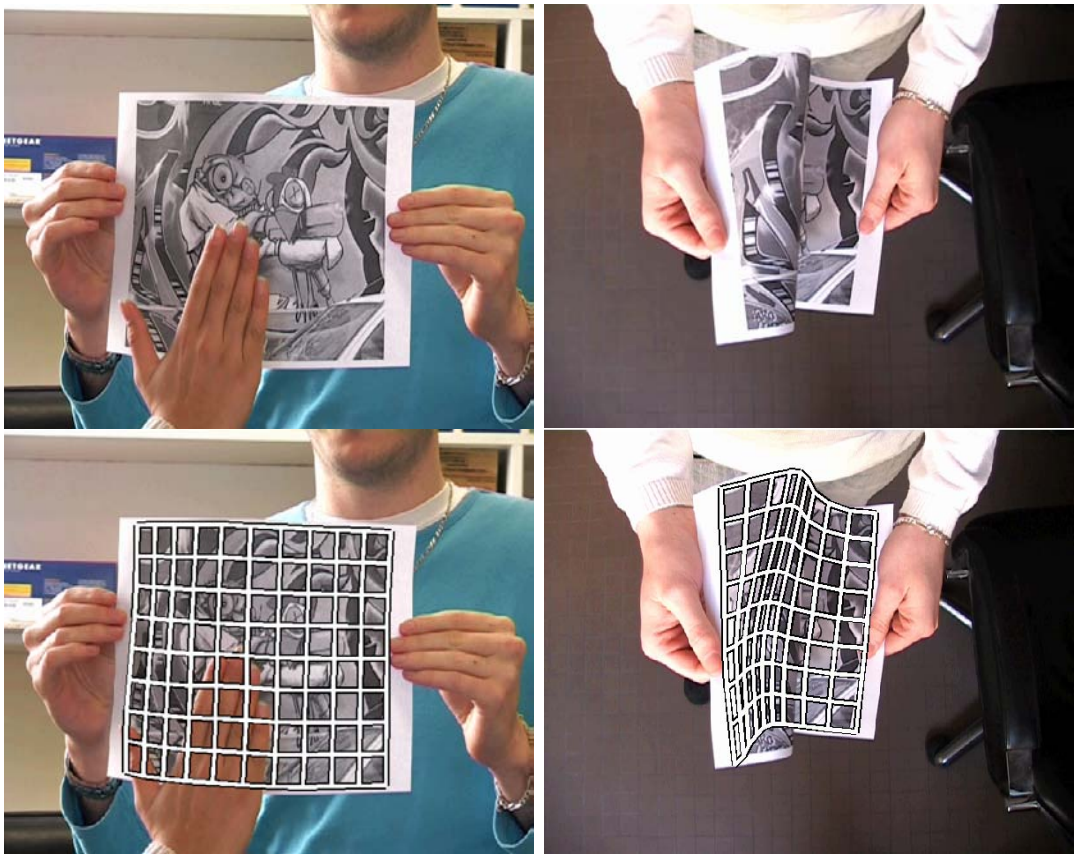


Fig. 1. **Classical methods fail on extreme self-occlusions.** Example of image registration using a classical outlier rejection algorithm (see §VII for details). Left: success on an external occlusion. Right: failure on an extreme self-occlusion.

We drop the image index i for clarity reasons in most of the paper. We define $\mathcal{R}$ to be the set of pixels of interest (in the template) and $\text{vect}_{\mathcal{R}}(\mathcal{M})$ the operator that vectorizes the elements of $\mathcal{M}$ indicated in $\mathcal{R}$. We write $\frac{\partial \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q};\mathbf{u})$ and $\frac{\partial_{\mathbf{d}} \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q};\mathbf{u})$ for the partial derivatives and directional partial derivatives (along direction $\mathbf{d} \in \mathbb{S}^1$) of the warp, respectively, where $\mathbb{S}^1$ is the unit circle. We respectively denote $\mathbf{E}_c(\mathbf{d};\mathbf{q};\mathbf{u}), \mathbf{E}_l(\mathbf{d};\mathbf{q};\mathbf{u})$ and $\mathbf{E}_r(\mathbf{d};\mathbf{q};\mathbf{u})$ the central, left and right finite difference approximations of the directional derivatives of the warp along direction $\mathbf{d} \in \mathbb{S}^1$, *e.g.* $\mathbf{E}_c(\mathbf{d};\mathbf{q};\mathbf{u}) = \frac{\mathcal{W}(\mathbf{q}+\epsilon\mathbf{d};\mathbf{u})-\mathcal{W}(\mathbf{q}-\epsilon\mathbf{d};\mathbf{u})}{2\epsilon}$. We denote v^x and v^y the x-component and the y-component of a two-dimensional vector $\mathbf{v}$. Finally, we use $\&$ for the elementwise 'and' operator and $\text{diag}(\mathcal{A},\mathcal{B})$ the block diagonal concatenation.

II. Preliminaries on Direct Non-rigid Registration

We use a parametric approach, *i.e.* parameters of a deformation model are estimated for each frame of a sequence, as opposed to the variational approach which estimates a function. The framework we propose for self-occlusions can also be used within the variational approach, though.

A. Choosing a Deformation Model

A large variety of parametric warps are used in the literature. The most well-known ones are Radial Basis Functions (RBF) and Free Form Deformations (FFD). The proposed framework is independent of the chosen warp. Thereafter, we present an implementation of an FFD warp. The basic idea of an FFD warp is to deform an object by manipulating a mesh of control points. We use an incremental FFD warp defined from [18] by:

$$\mathcal{W}(\mathbf{q};\mathbf{u}) = \sum_{k=0}^{3}\sum_{l=0}^{3} \mathcal{B}_k(v)\mathcal{B}_l(w)(\mathbf{s}_{i+k,j+l} + \mathbf{u}_{i+k,j+l}),$$

where $\mathbf{s}_{i,j}$ are the reference position of the control points, $\mathcal{B}_k, \mathcal{B}_l$ are B-spline interpolation coefficients[1] evaluated at the normalized coordinates v, w. The parameter vector $\mathbf{u}$ includes the displacement of the control points $\mathbf{s}_{i,j}$: $\mathbf{u}^\mathsf{T} = \left(\ldots u^x_{i,j} \ldots, \ldots u^y_{i,j} \ldots\right)$. We denote $\mathcal{S}$ the set of control points $\mathbf{s}_{i,j}$.

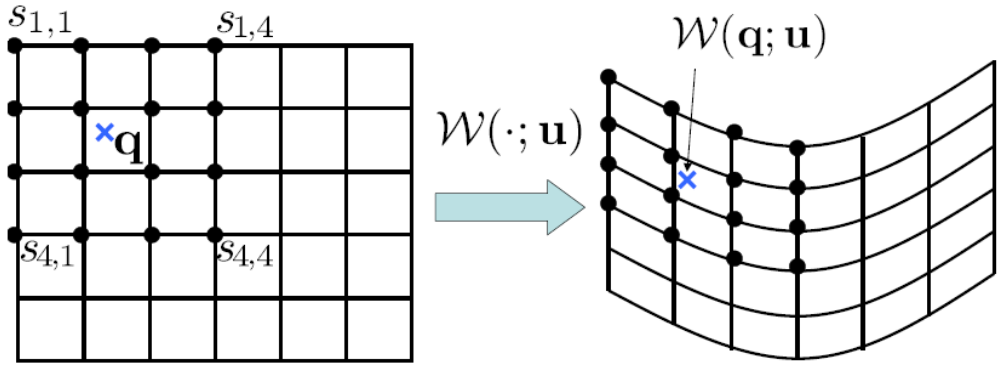


Fig. 2. The FFD warp model we use is based on the cubic B-spline: the displacement of a pixel $\mathbf{q}$ is a linear combination of 16 neighboring control points displacements.

[1]Any smooth basis function can be used. We use the cubic B-spline since it provides continuous second derivatives and local support.

B. The Cost Function

Direct non-rigid registration algorithms usually use as data term $\mathcal{E}_d$ the two-norm of the difference image $\mathcal{D}$ between the template and the current image warped toward the template, *i.e.* $\mathcal{D}(\mathbf{q};\mathbf{u}) = \mathcal{I}_0(\mathbf{q}) - \mathcal{I}(\mathcal{W}(\mathbf{q};\mathbf{u}))$, giving:

$$\mathcal{E}_d(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{R}} \mathcal{D}^2(\mathbf{q};\mathbf{u}). \tag{1}$$

Photometric changes such as those due to lighting variations can be dealt with in different ways. A model can be explicitly embedded in $\mathcal{E}_d$ as in [4] for global illumination changes and in [30] for local ones (piecewise affine and spline models). In [25], the images are projected to shadow invariant space. This requires one to estimate additional photometric calibration parameters along with the registration parameters.

For purely twodimensional registration, as is the case in this paper, smoothness constraints are used to make the motion estimation well-posed. These soft constraints can be implicitly incorporated in a parameterized warp as *e.g.* in Thin-Plate Spline warps. We use a simple smoother $\mathcal{E}_s$ that is added to the error function. Several ones exist in the literature. A common one is the first order quadratic smoothness constraint of Horn and Schrunk [17]. We use the second order smoothness constraint, called the bending energy, related to the Thin-Plate Spline [10]:

$$\int_{\mathcal{R}}\int_{\mathcal{R}} \left(\frac{\partial \mathcal{W}^2}{\partial^2 x}\right)^2 + 2\left(\frac{\partial \mathcal{W}^2}{\partial x \partial y}\right)^2 + \left(\frac{\partial \mathcal{W}^2}{\partial^2 y}\right)^2 \mathrm{d}x\mathrm{d}y.$$

It has empirically been proved to be well suited for the case of smooth surfaces, see *e.g.* [6].

The bending energy can be expressed in discrete form as a quadratic function of $\mathbf{u}$ using a finite difference approximation of the first and second derivatives on the surface:

$$\mathcal{E}_s(\mathbf{u}) = \mathbf{u}^\mathsf{T}\mathcal{K}\mathbf{u} = \mathbf{u}_x^\mathsf{T}\mathcal{C}\mathbf{u}_x + \mathbf{u}_y^\mathsf{T}\mathcal{C}\mathbf{u}_y,$$

where $\mathcal{K} = \mathrm{diag}(\mathcal{C},\mathcal{C})$ and $\mathcal{C} = \mathcal{C}_{xx}^\mathsf{T}\mathcal{C}_{xx} + 2\mathcal{C}_{xy}^\mathsf{T}\mathcal{C}_{xy} + \mathcal{C}_{yy}^\mathsf{T}\mathcal{C}_{yy}$. $\mathcal{C}_{xx}, \mathcal{C}_{xy}, \mathcal{C}_{yy}$ are finite difference operators approximating the second partial derivatives $\frac{\partial^2 .}{\partial^2 x}$, $\frac{\partial^2 .}{\partial x \partial y}$, $\frac{\partial^2 .}{\partial^2 y}$ respectively. More details for computing $\mathcal{K}$ are given in [27].

The compound cost function is then given by:

$$\mathcal{E}(\mathbf{u}) = \mathcal{E}_d(\mathbf{u}) + \lambda_s \mathcal{E}_s(\mathbf{u}), \tag{2}$$

where λ_s is a weighting parameter which defines the trade-off between the data term and the smoother. It is usually fixed experimentally [28] or via cross validation [5].

Dealing with occlusions. It is usually done by embedding a robust kernel ρ in the data term. A one is the Huber kernel [16]:

$$\rho(t) = \begin{cases} \frac{1}{2}t^2 & \text{if } \|t\| \leq \sigma \\ \sigma\|t\| - \frac{1}{2}\sigma^2 & \text{otherwise,} \end{cases} \tag{3}$$

but many others exist such as the Lorentzian kernel [14] or the Tuckey function [20] for example. The parameter σ is called the 'scale parameter'. It is usually fixed as a known constant but may also be estimated throughout the registration [2]. The robust kernel limits the influence of outliers.

C. Minimization of the Direct Criterion

a) The Gauss-Newton approximation: using an additive update of the parameter vector, *i.e.* $\mathbf{u} \leftarrow \mathbf{u} + \boldsymbol{\delta}_u$, Gauss-Newton can be used in a straightforward manner for minimizing equation (1) or in conjunction with complexity tuning schemes as in [6], [21] for non-rigid warps. The local Gauss-Newton approximation to $\mathcal{E}_d$ is given by first order Taylor expansion in $\boldsymbol{\delta}_u$ of each term being squared in equation (1):

$$\mathcal{E}_d(\mathbf{u} + \boldsymbol{\delta}_u) \approx \sum_{\mathbf{q}\in\mathcal{R}} \|\mathcal{D}(\mathbf{q}) + \mathbf{g}(\mathbf{q};\mathbf{u})^\mathsf{T}\boldsymbol{\delta}_u\|^2.$$

The gradient vector $\mathbf{g}$ factors as the product of the image gradient vector and the Jacobian matrix $\mathcal{L}$ of the warp[2], *i.e.* $\mathbf{g}(\mathbf{q};\mathbf{u}) = \nabla\mathcal{I}^\mathsf{T}\mathcal{L}$ (we drop the arguments $\mathbf{q}$ and $\mathbf{u}$ for clarity). The Gauss-Newton approximation induces a Linear Least Squares minimization problem in $\boldsymbol{\delta}_u$. Defining $\mathcal{J}$ as the Jacobian matrix of the error, obtained by stacking the gradient vectors $\mathbf{g}(\mathbf{q};\mathbf{u})^\mathsf{T}$ for all pixels $\mathbf{q}$ in $\mathcal{R}$, and $\mathbf{D} = \text{vect}_\mathcal{R}(\mathcal{D})$ the residual error vector, the solution is obtained through the normal equations:

$$\mathcal{H}\boldsymbol{\delta}_u = -\mathbf{b} \qquad \text{with} \qquad \mathcal{H} = \mathcal{J}^\mathsf{T}\mathcal{J} \qquad \text{and} \qquad \mathbf{b} = \mathcal{J}^\mathsf{T}\mathbf{D}.$$

Matrix $\mathcal{H}$ is the Gauss-Newton approximation to the Hessian matrix. The Jacobian matrix $\mathcal{J}$ must be recomputed at each iteration, implying that $\mathcal{H}$ must be recomputed and inverted as well.

[2]The Jacobian matrix of an FFD warp is constant. It is also the case for a TPS warp.

b) Efficient Second order Minimisation: a second order approximation of $\mathcal{E}_d$ called ESM (Efficient Second-order Minimization), theoretically better than the Gauss-Newton one, is proposed in [7]:

$$\mathcal{E}_d(\mathbf{u}+\boldsymbol{\delta}_u) \approx \sum_{\mathbf{q}\in\mathcal{R}} \|\mathcal{D}(\mathbf{q}) + \frac{1}{2}(\mathbf{g}(\mathbf{q};\mathbf{u}) + \mathbf{g}(\mathbf{q};\mathbf{u}_0))^\mathsf{T}\boldsymbol{\delta}_u\|^2.$$

It has been shown to improve the convergence rate compared to Gauss-Newton, without increasing the computation time per iteration, since the gradient vectors $\mathbf{g}(\mathbf{q};\mathbf{u}_0) = \nabla\mathcal{I}_0{}^\mathsf{T}\mathcal{L}$ are constant. It thus yields a more efficient algorithm. ESM however requires that the warp has a group structure, but [23] reports good convergence properties for non-rigid warps with the use of a Jacobian approximation.

c) The Inverse Compositional framework: the Inverse Compositional Algorithm (ICA), proposed by Baker *et al.* [3], is a more efficient solution for direct registration. It is based on a compositional (instead of the classical additive) update of the parameters $\mathcal{W}(\cdot;\mathbf{u}) \leftarrow \mathcal{W}(\cdot;\mathbf{u}) \circ \mathcal{W}^{-1}(\cdot;\boldsymbol{\delta}_u)$ and leads to a constant Hessian matrix. It assumes that the warp is invertible and is thus limited in its initial version to warps which have a group structure. An extension of the ICA to non-groupwise warps such as the FFD is proposed in [15]. One drawback of the Inverse Compositional paradigm is that the Hessian matrix should in principle not be kept constant in the presence of outliers. In [2] the spatial coherence of the outliers is used as a prior. The Hessian matrix is not constant but can be computed very efficiently.

D. Hierarchical Framework

A hierarchical, coarse-to-fine framework is often used in image registration. It allows one to register images with large displacements. The key idea is to estimate the motion at a low resolution initially and then to propagate it to higher resolutions to improve its accuracy. The additional steps implied by the hierarchical framework are the image pyramid construction and the motion propagation through pyramid levels. As for the former, several levels are constructed, each containing the same image as the previous level, with lower dimensions. The result is a pyramid, where the lowest level is the initial image, and each level above is the same image at $\frac{1}{4}$ of its size. Different approaches can be used to create a lower resolution image from the initial one. One possibility is to compute each pixel value at level k as a weighted average of values

in level $k-1$ within a given window. A Gaussian kernel is used as the 'weighting pattern'. This operation is called reduction. The coarse-to-fine propagation of the estimated motion at a higher level is achieved via expansion [12].

E. The 'classical' approach

Thereafter, we compare our proposed approach with what we call the classical one. By classical we mean that it does not explicitly models self-occlusions. It minimizes the following cost function with a hierarchical Gauss-Newton algorithm:

$$\mathcal{E}(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{R}} \rho\left(\mathcal{D}(\mathbf{q};\mathbf{u})\right) + \mathbf{u}^{\mathsf{T}}\mathcal{K}\mathbf{u}, \tag{4}$$

where $\mathcal{K}$ is the matrix representing the bending energy (see above). The Huber function (3) is used as M-estimator, The scale parameter σ is estimated throughout the iterations using the median absolute deviation (MAD) of the residuals.

III. Problem Statement and the Proposed Solution

Existing direct non-rigid registration algorithms are well adapted to deal with external occlusions. Minimizing equation (4) makes the smoother dominate for pixels detected as outliers. This behavior is well suited for external occlusions *i.e.* when an object enters the space between the camera and the surface of interest as shown in figure 1: whose pixels belonging to the hand are not taken into account in the minimization, the estimated warp is smooth in this area. In the case of self-occlusions, a part of the surface disappears due to its bending. The warp must make the self-occluded area disappear, thereby creating discontinuities in the warp, which equation (4) does clearly not allow.

Our approach is to enforce the warp to shrink along the self-occlusion boundary. We add a new penalty term that we call the 'shrinker' to the cost function, as figure 3 illustrates. The shrinker prevents the warp to fold *i.e.* discourages warp smoothness in the self-occluded areas. We note that a robust kernel could be incorporated in the smoother term $\mathcal{E}_s$ to prevent smoothness in those areas. This however cannot be used since neither the data term nor the smoother would constrain the warp at the self-occlusion boundary anymore. The warp would thus not shrink properly.

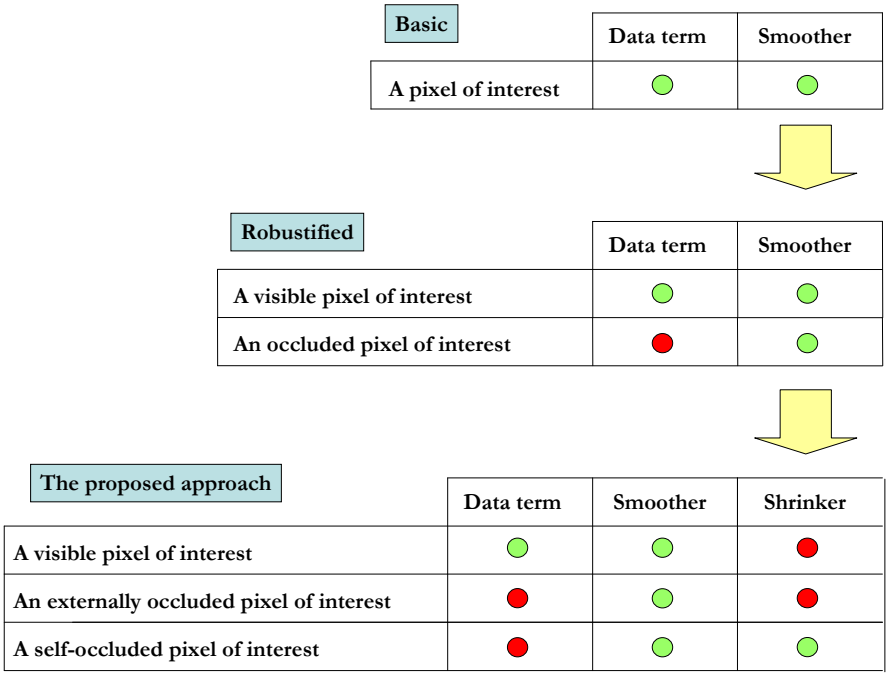


Fig. 3. The proposed approach classifies each pixel as being visible, externally or self-occluded. It appears to be clearly more adapted to self-occlusions than the other approaches. Green, Red (or light gray, dark gray on the gray-scale printed paper) circles mean that the associated term is activated or disactivated in the cost function, respectively.

So as to activate the shrinker, we have to find the self-occluded pixels. Thresholding the data terms, does obviously not perform well. We propose to use the warp behavior to detect those pixels. This is possible when the warp correctly shrinks at the self-occlusion boundary as explained in details in §IV. A natural idea that comes to mind is to use the warp's directional derivatives. This does not work if the warp is not shrunk along the occlusion boundary, as shown on figure 4. Warp derivatives nevertheless give some cues: foldings are characterized by a sign change of the warp derivatives along some direction $\mathbf{d}$.

To summarize, visible, externally and self-occluded pixels have to be handled independently, as the proposed approach does. The warp has to be smoothed for both visible and externally occluded pixels, while it must shrink in self-occluded areas. In this case, warp derivatives can be used to detect self-occluded pixels. The detection of externally occluded pixels is achieved by thresholding the data term.

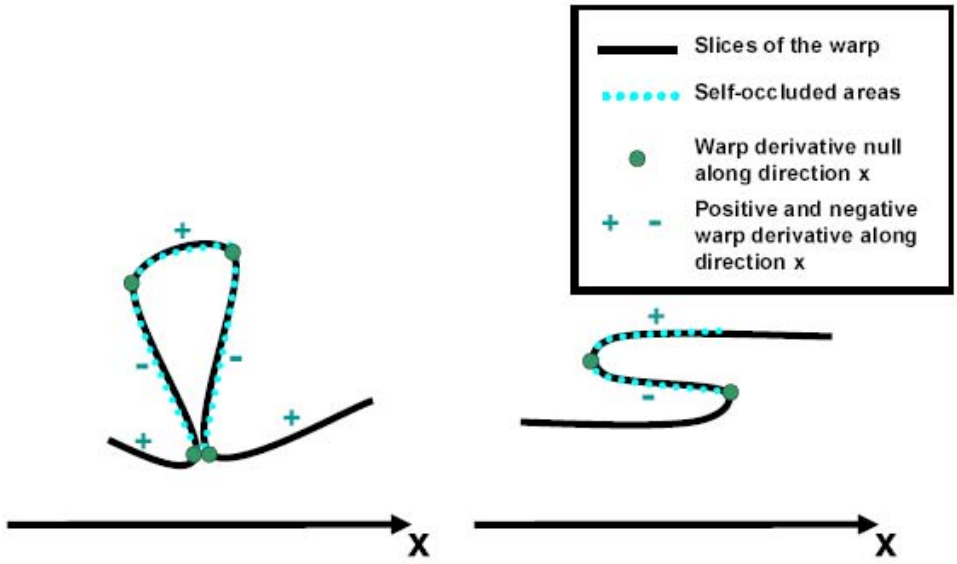


Fig. 4. Warp folds and ambiguities in 2D self-occlusion detection. The 1D loops we show are slices of the 2D warp. Left: example where the self-occluded area is characterized by two sign changes in some directional warp derivative. Right: example where the self-occluded area is characterized by a single sign change only. The diversity of possible folds makes self-occluded area detection difficult.

IV. Self-Occlusion Detection Framework

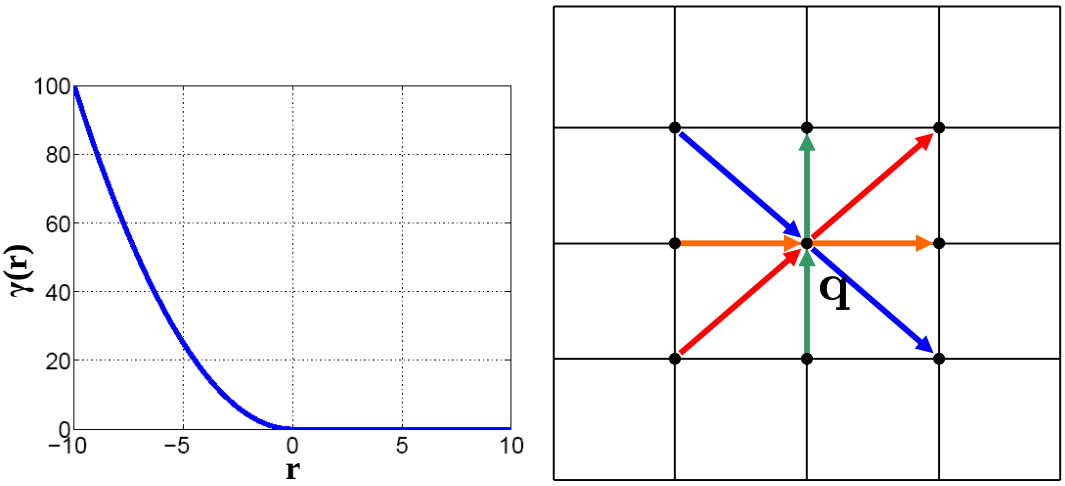
The proposed approach to detect the self-occluded pixels outputs two versions of the self-occlusion map: a binary $\mathcal{H}(\mathbf{q};\mathbf{u})$ and a probabilistic $\widehat{\mathcal{H}}(\mathbf{q};\mathbf{u})$ ones. The former presents several drawbacks that are described below. We use the probabilistic map in most of our experiments.

A. The Shrinker: Preventing Folds, Forcing Shrinkage

The goal of the Shrinker is to enforce the warp to shrink rather than to fold at the self-occlusion boundary. This behavior is required by the self-occlusion detection modules described in §IV-B. A regularized warp naturally folds onto itself in case of an extreme self-occlusion. Folds make the warp many-to-many in visible parts. These configurations are characterized by a variation in the sign of the partial derivatives of the warp along some direction $\mathbf{d}$, as shown in figure 4. The shrinker has to penalize sign variations of the warp derivatives and to force the warp to collapse along this direction. We note that diffeomorphic warps are proposed in [13]. They enforce one-to-one correspondences by preventing warp folds. Diffeomorphic warps are unadapted in this context since self-occluded areas are modeled by a locally many-to-one warp.

The shrinker is built via the function γ shown on figure 5(a):

$$\gamma(r) = \begin{cases} 0 & \text{if } r \geq 0 \\ r^2 & \text{otherwise.} \end{cases}$$



(a) (b)

Fig. 5. (a) The function γ used in the shrinker term. (b) The set of directions $\mathcal{F}$.

It is applied to the element-wise product between the left and the right derivatives of the warp evaluated at points $\mathbf{q} \in \mathcal{R}$ and integrated over the directions $\mathbf{d} \in \mathbb{S}^1$. In practice the integral is discretized on a set of directions $\mathcal{F}$ shown in figure 5(b). The shrinker is given by:

$$\mathcal{E}_f(\mathbf{u}) = \sum_{\mathbf{q}\in\mathcal{R}} \sum_{\mathbf{d}\in\mathcal{F}} \sum_{c\in\{x,y\}} \gamma(\mathbf{E}_l^c(\mathbf{d},\mathbf{q},\mathbf{u})\mathbf{E}_r^c(\mathbf{d},\mathbf{q},\mathbf{u})), \tag{5}$$

Minimizing equation (5) imposes the above describe constraints: it only penalizes points for which the right and left derivatives have opposite signs, since function γ returns null values for positive inputs. The minimum of $\mathcal{E}_f$ is obtained when the warp does not fold.

B. The Binary Self-Occlusion Map

The warp collapses at point $\mathbf{q}$ along direction $\mathbf{d}$ if points $\mathcal{W}(\mathbf{q}+\epsilon\mathbf{d};\mathbf{u})$ and $\mathcal{W}(\mathbf{q}-\epsilon\mathbf{d};\mathbf{u})$ are quasi identical: $\mathcal{W}(\mathbf{q}+\epsilon\mathbf{d};\mathbf{u}) \approx \mathcal{W}(\mathbf{q}-\epsilon\mathbf{d};u)$, as illustrated on figure 6. This implies that the directional derivative along $\mathbf{d}$ evaluated at point $\mathbf{q}$ vanishes: $\frac{\partial_{\mathbf{d}}\mathcal{W}}{\partial\mathbf{q}}(\mathbf{q};\mathbf{u}) = 0$. One consequence

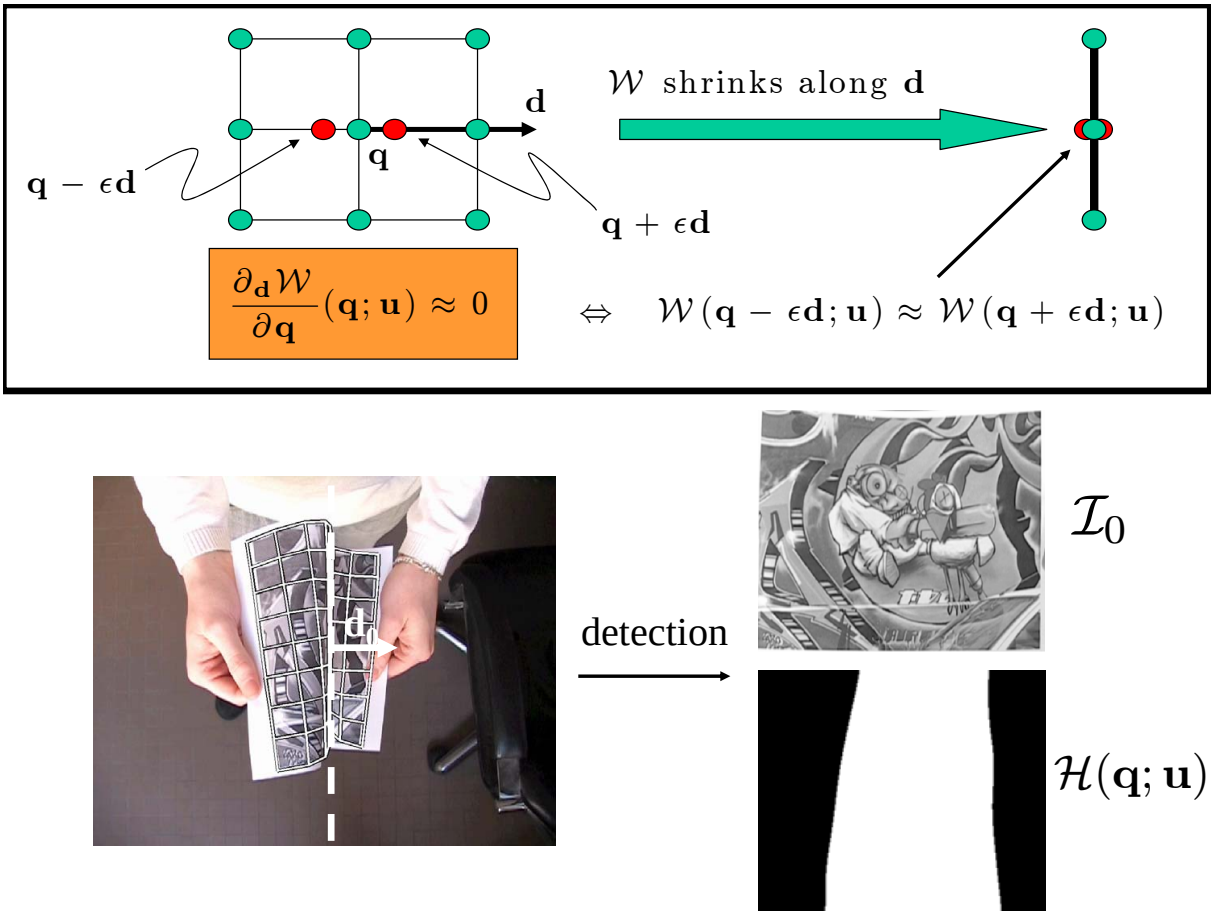


Fig. 6. Example of self-occluded area detection. Top: the warp shrinkage along direction $\mathbf{d}_x$ makes the warp derivative along this direction vanish. The search of near zero warp derivative values allows us to detect shrunk areas. Bottom: The warp shrinks along the self-occlusion boundary. It causes the warp derivative to vanish along direction $\mathbf{d}_0$. Associated pixels are detected as self-occluded via the proposed detection modules (white areas of $\mathcal{H}(\mathbf{q};\mathbf{u})$).

of the 'shrinking' property is that for a self-occluded pixel $\mathbf{q}$, there exists at least a direction $\mathbf{d} \in \mathbb{S}^1$ such that the partial derivative of the warp at $\mathbf{q}$ in direction $\mathbf{d}$ vanishes.

We thus define the binary self-occlusion map $\mathcal{H}(\mathbf{q};\mathbf{u})$ as:

$$\mathcal{H}(\mathbf{q};\mathbf{u}) = \begin{cases} 1 & \exists \mathbf{d} \in \mathbb{S}^1 \;\mid\; \|\mathbf{E}_c(\mathbf{d};\mathbf{q};\mathbf{u})\| < rd \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\mathbf{E}(\mathbf{d};\mathbf{q};\mathbf{u})$ is the finite difference approximation of $\frac{\partial_{\mathbf{d}}\mathcal{W}}{\partial\mathbf{q}}(\mathbf{q};\mathbf{u})$. We fix the threshold rd slightly above 0, *e.g.* 0.1, in order to tolerate noisy warps and to anticipate[3] self-occlusions.

[3]We mean to detect them slightly before they occur

In practice, the exhaustive search of the direction $\mathbf{d}$ required by (6) is replaced by a minimization problem:

$$\left(\min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}_c(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2 \right) < rd.$$

This minimization problem has a closed-form solution. Let L be the (finite difference) Jacobian matrix of $\mathcal{W}$ evaluated at $(\mathbf{q}; \mathbf{u})$. We have $\mathbf{E}_c(\mathbf{d}; \mathbf{q}; \mathbf{u}) = \mathsf{L}\mathbf{d}$, and thus:

$$\sigma_0 = \min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}_c(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2 = \min_{\mathbf{d} \in \mathbb{S}^1} \mathbf{d}^\mathsf{T} \mathsf{L}^\mathsf{T} \mathsf{L} \mathbf{d}.$$

Spectral decomposition of the symmetric matrix $\mathcal{O} = \mathsf{L}^\mathsf{T}\mathsf{L}$ gives:

$$\sigma_0 = \frac{1}{2} \left(\mathcal{O}_{1,1} + \mathcal{O}_{2,2} - \sqrt{(\mathcal{O}_{1,1} - \mathcal{O}_{2,2})^2 + 4\mathcal{O}_{1,2}^2} \right).$$

C. The Probabilistic Self-Occlusion Map

Using a probabilistic map has several assets: it allows one to keep some incertitude for those pixels lying near the self-occlusion boundary. It is also well adapted to a hierarchical minimization framework (more details are given in §V-C). Thereafter, the probabilistic self-occlusion map is used.

Upgrading the binary map to a probabilistic one is achieved by smoothing the hard transition related to the binary decision. We use the smooth Heaviside function ψ but any other sigmoid function could be employed as well. This function is given by:

$$\psi(x; r) = \frac{\exp(2k(x - r))}{1 + \exp(2k(x - r))},$$

where k and r are two thresholds encoding the slope and the position of the transition respectively. The impact of those two thresholds are illustrated on figure 7.

The probabilistic map is then obtained by:

$$\widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) = \psi \left(\min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2, rd \right), \tag{7}$$

typical values for the two thresholds are: $k = 40$ et $rd = 0.1$.

Examples of binary and probabilistic self-occlusion maps are shown on figure 8. They are obtained at frame 120 of the paper sequence shown in figure 14 (§VII).

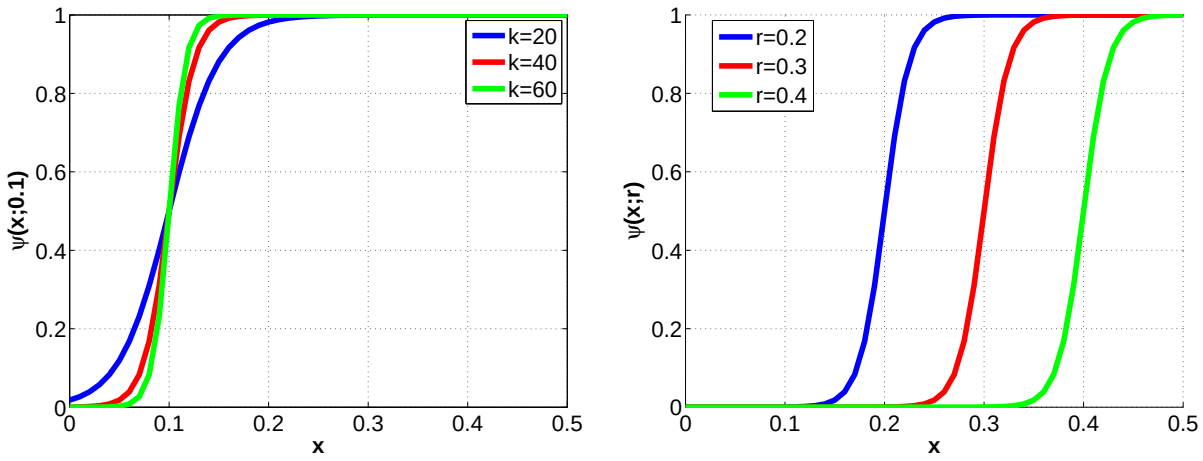


Fig. 7. The smooth Heaviside function used to build the probabilistic self-occlusion map. Left: varying k, fixing $r = 0.1$. Right: varying r, fixing $k = 40$.

We note that warp shrinking is the natural behavior of the warp when pixels vanish due to foreshortening under perspective projection. In these cases, the detection module still give correct results. The shrinking constraint does not however activates since the warp does not fold. The proposed method thus naturally deals with these configurations.

V. Image Registration with Self-Occlusions Reasoning

A. The Cost Function

The cost function has three terms: $\mathcal{E}_t(\mathbf{u}) = \mathcal{E}_{hd}(\mathbf{u}) + \lambda_s \mathcal{E}_s(\mathbf{u}) + \lambda_f \mathcal{E}_f(\mathbf{u})$. One possible data term is:

$$\sum_{\mathbf{q}\in\mathcal{R}} (1 - \widehat{\mathcal{H}}(\mathbf{q};\mathbf{u}))\mathcal{D}^2(\mathbf{q};\mathbf{u}). \tag{8}$$

So as to prevent the warp fully shrink, one could add a penalty such as $\lambda_d \widehat{\mathcal{H}}(\mathbf{q};\mathbf{u})$. We however take a different approach. Indeed, simultaneously estimating the self-occlusion map and the warp is highly nonlinear and thus likely to be subject to many local minima.

We instead use a two-step minimization scheme:

- Set $\mathbf{u} \leftarrow \tilde{\mathbf{u}}$ where $\tilde{\mathbf{u}}$ is the latest parameter estimate
- **First step.** The parameter vector $\mathbf{u}$ is updated by using the current estimate of the probabilistic self-occlusion map $\widehat{\mathcal{H}}(\mathbf{q};\tilde{\mathbf{u}})$. The following global energy is minimized:

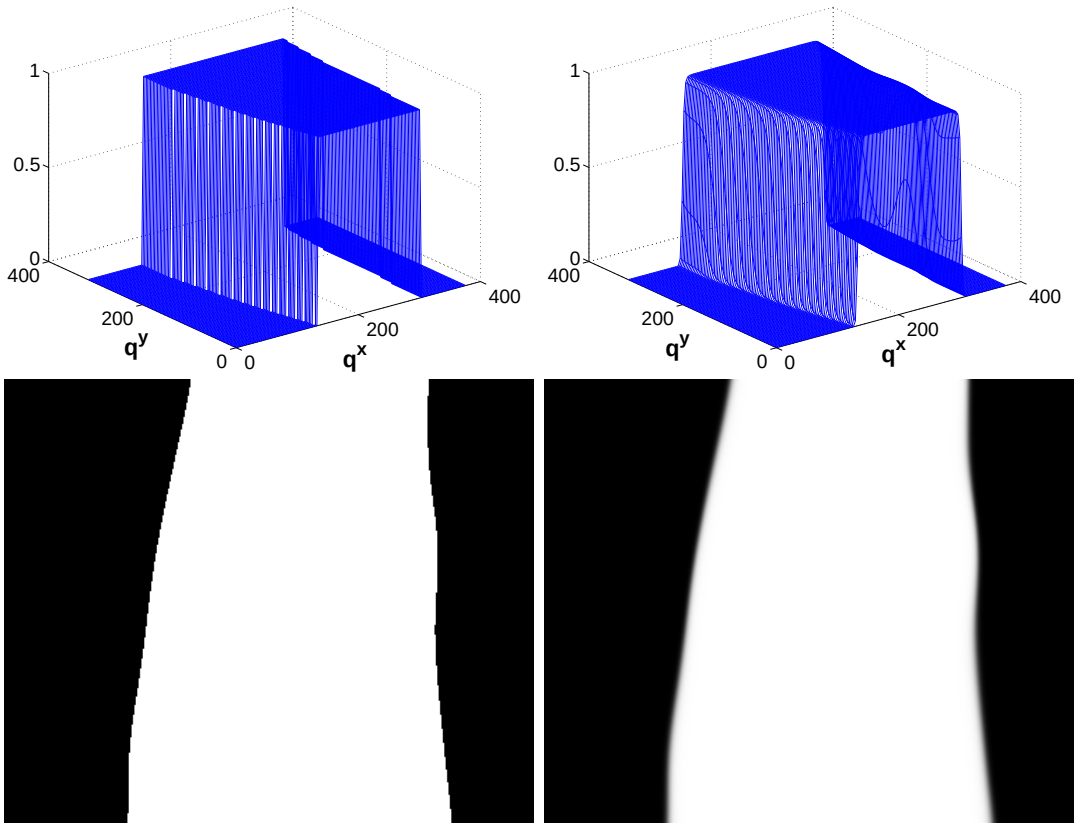


Fig. 8. Illustration of the binary self-occlusion map $\mathcal{H}(\cdot;\mathbf{u})$ (left) and the probabilistic one $\widehat{\mathcal{H}}(\cdot;\mathbf{u})$ (right). Top: 2.5D representation of the maps. Bottom: White and black pixels are labeled as self-occluded and visible respectively. The two thresholds are chosen as $k = 40$ and $r = rd = 0.1$.

$$\min_{\mathbf{u}} \mathcal{E}(\mathbf{u}) = \overbrace{\sum_{\mathbf{q}\in\mathcal{R}} (1 - \widehat{\mathcal{H}}(\mathbf{q};\tilde{\mathbf{u}}))\mathcal{D}^2(\mathbf{q};\mathbf{u})}^{\mathcal{E}_{hd}(\mathbf{u})} + \lambda_s \mathbf{u}^\mathsf{T}\mathcal{K}\mathbf{u} + \lambda_f \sum_{\mathbf{q}\in\mathcal{R}} \sum_{\mathbf{d}\in\mathcal{F}} \sum_{c\in\{x,y\}} \gamma(\mathbf{E}_l^c(\mathbf{d},\mathbf{q},\mathbf{u})\mathbf{E}_r^c(\mathbf{d},\mathbf{q},\mathbf{u})). \quad (9)$$

- **Second step.** Update the probabilistic self-occlusion map:

$$\widehat{\mathcal{H}}(\mathbf{q};\mathbf{u}) \leftarrow \psi\left(\min_{\mathbf{d}\in\mathbb{S}^1} \|\mathbf{E}(\mathbf{d};\mathbf{q};\mathbf{u})\|^2, rd\right)$$

B. External Occlusions

The probabilistic self-occlusion map $\widehat{\mathcal{H}}$ may be replaced by a probabilistic visibility map $\widehat{\mathcal{H}}_v$. It is made by combining the self-occlusion map described in §IV-C and a probabilistic external occlusion map $\widehat{\mathcal{H}}_e(\mathbf{q};\mathbf{u})$, obtained by using the intensity discrepancy $\mathcal{D}(\mathbf{q};\mathbf{u})$ and the

smooth Heaviside function $\psi(\cdot; r_e)$. The threshold r_e can be set fixed or estimated throughout the minimization procedure as for an M-estimator.

Given this definition of an externally occluded pixel, it seems obvious that self-occluded pixels could also be detected as externally occluded. External occlusion detection is thus only applied to those pixels which are not already detected as self-occluded:

$$\widehat{\mathcal{H}}_e(\mathbf{q}, \mathbf{u}) = \begin{cases} 0 & \widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) > 0.5 \\ \psi(\mathcal{D}(\mathbf{q}; \mathbf{u}), r_e) & \text{otherwise,} \end{cases} \tag{10}$$

where we make the hypothesis that a pixel cannot be self-occluded and occluded by an external object at the same time.

The accuracy of the externally occluded pixel detection is improved by enforcing spatial constraints: erosion and dilation operators are applied on $\widehat{\mathcal{H}}_e(\mathbf{q}, \mathbf{u})$ in turn. This penalizes local occlusion detections that may be introduced by noise or error in the registration and enforces the detection of dense regions. Experiments with both external and self-occlusions are reported in §VII.

The probabilistic visibility map is obtained as the element-wise product between the probabilistic external and self-occlusion maps:

$$\widehat{\mathcal{H}}_v(\mathbf{q}; \mathbf{u}) = \widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) \odot \widehat{\mathcal{H}}_e(\mathbf{q}; \mathbf{u}). \tag{11}$$

C. Minimization Based on Gauss-Newton

The global error function (9) is minimized using the Gauss-Newton algorithm. The sparse structure of the Jacobian matrix is taken into account in the minimization. A hierarchical approach [8] is used. It is mainly required at the disocclusion stage. The coarse-to-fine refinement step propagates the displacement field and the self-occlusion map $\widehat{\mathcal{H}}$ (or the visibility map $\widehat{\mathcal{H}}_v$) using pyramid expansion operations [12].

We note that if binary maps were used, expansion operations would spoil the binary property. An additional step to recover a binary map could be one of:

- non 1 labels are fixed at 0
- labels between 0 and 1 are rounded to the nearest integer
- non zero labels are fixed at 1

The last solution should be used. It over-estimates the binary self-occlusion map, preventing misalignment at the boundaries. However, probabilistic maps resolve this issue naturally.

How to find the Jacobian matrix for each term in the error function is described below.

1) The data term: The data energy is a weighted $\mathcal{L}_2$ norm of the intensity discrepancy. The Jacobian matrix is given by: $\mathcal{J}_{hd} = \mathrm{diag}((1 - \widehat{\mathcal{H}}(\mathbf{q}; \tilde{\mathbf{u}})))\mathcal{J}_{\mathcal{I}}$. $\mathcal{J}_{\mathcal{I}}$ is obtained by stacking the gradient vectors $\mathbf{g}_{\mathcal{I}}(\mathbf{q}; \mathbf{u})^{\mathsf{T}}$ for all pixels in $\mathcal{R}$. The gradient vector $\mathbf{g}_{\mathcal{I}}$ factors as the product of the image gradient vector and the Jacobian matrix of the warp: $\mathbf{g}_{\mathcal{I}}(\mathbf{q}; \mathbf{u}) = \nabla\mathcal{I}^{\mathsf{T}}\mathcal{L}$ respectively evaluated at $\mathcal{W}(\mathbf{q}; \mathbf{u})$ and at $\mathbf{q}$.

2) The smoother: The Jacobian and the residual error for the smoother are simply given by:

$$\mathcal{J}_s = \sqrt{\lambda_s}\mathcal{Z} \text{ and } \mathcal{D}_s = \sqrt{\lambda_s}\mathcal{Z}\mathbf{u},$$

where $\mathcal{Z}^{\mathsf{T}}\mathcal{Z} = \mathcal{K}$ and $\mathcal{K}$ represents the bending energy as described in §II.

3) The shrinker: The Jacobian matrix of the shrinker is given by:

$$\mathcal{J}_f = \lambda_f \mathcal{J}_{\mathbf{E}}.$$

$\mathcal{J}_{\mathbf{E}}$ is obtained by stacking for each triplet $(\mathbf{d}; \mathbf{q}; c) \in (\mathcal{F}; \mathcal{R}; \{x, y\})$ the gradient vectors $\mathbf{g}_f$:

$$\mathbf{g}_f(\mathbf{d}; \mathbf{q}; \mathbf{u}) = \mathbf{E}_r^c \nabla \mathbf{E}_l^c + \mathbf{E}_l^c \nabla \mathbf{E}_r^c,$$

for which the associated product $\mathbf{E}_l^c(\mathbf{d}, \mathbf{q}, \mathbf{u})\mathbf{E}_r^c(\mathbf{d}, \mathbf{q}, \mathbf{u})$ is negative.

4) Parameter update: The parameter increment $\boldsymbol{\delta}_u$ is obtained by solving the normal equations using a sparse solver:

$$(\mathcal{J}_t^{\mathsf{T}}\mathcal{J}_t)\boldsymbol{\delta}_u = -\mathcal{J}_t^{\mathsf{T}}\mathcal{D}_t, \tag{12}$$

where:

$$\mathcal{J}_t^{\mathsf{T}} = \begin{pmatrix} \mathcal{J}_{hd}^{\mathsf{T}} & \mathcal{J}_s^{\mathsf{T}} & \mathcal{J}_f^{\mathsf{T}} \end{pmatrix} \text{ and } \mathbf{D}_t^{\mathsf{T}} = \begin{pmatrix} \mathbf{D}_{hd}^{\mathsf{T}} & \mathbf{D}_s^{\mathsf{T}} & \mathbf{D}_f^{\mathsf{T}} \end{pmatrix}.$$

The sparse structure of $\mathcal{J}_{hd}$ results from the local influence of the warp control points. The residual associated to a pixel $\mathbf{q}$ depends only on the vertex displacements (along x and y) in its vicinity. It yields a band diagonal structure against displacements along x ($\mathbf{u}^x$) and along y ($\mathbf{u}^y$) as shown on figure 9. The width of the bands depend on the interpolation kernel. For cubic

B-splines, 16 control points affect the position of one pixel, yielding 32 non-zero components per pixel, *i.e.* per row, in $\mathcal{J}_{hd}$. The Jacobian of the smoother $\mathcal{J}_s$ has a band diagonal structure against the whole displacements $\mathbf{u}$. Finally, the Jacobian of the shrinker $\mathcal{J}_f$ is also very sparse. The sparse structure of the global Jacobian and the Gauss-Newton approximation to the Hessian matrix, *i.e.* ${\mathcal{J}_t}^\top \mathcal{J}_t$, are shown in figure 9. The former has no specific structure such as the well-organized block structure introduced in rigid bundle adjustment. Using a sparse method is necessary to solve the normal equations since the Hessian matrix is very large. The resolution is otherwise untractable. We use the sparse Cholesky factorization package [1] induced in the Matlab routines[4].

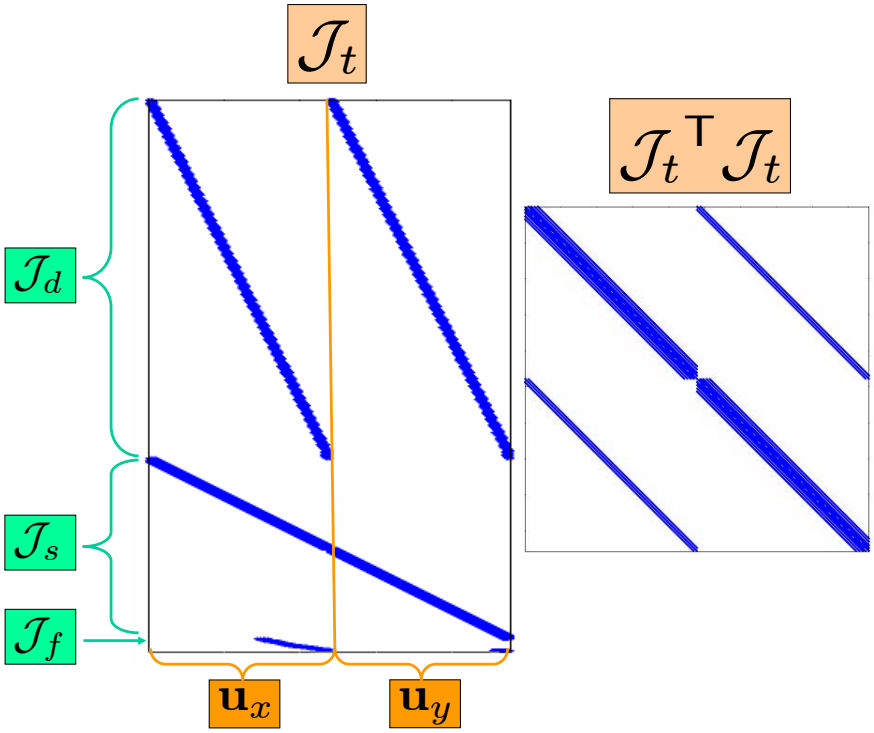


Fig. 9. The Jacobian and Hessian (Gauss-Newton approximation) matrices have a very sparse structure. Note that only the non-zero rows of $\mathcal{J}_f$ are shown. These matrices have been obtained for the paper sequence shown in figure 14.

[4]More precisely, we use the 'mldivide' matlab function

VI. Retexturing Self-Occluded Surfaces

Once the warp is estimated, realistic self-occluded surface retexturing is not trivial. First, for extreme self-occlusions, jittering usually appears in the registration, since remaining visible parts of the surface might be very small. We propose to use a median filter to enforce temporal smoothness. It is compared with linear filtering and cubic spline regularization. Second, pixels on the self-occlusion boundary are challenging to retexture. The method depends directly on the representation used for the self-occlusions.

A. Temporal Smoothness

One possibility is to convolve the trajectories $\mathbf{t}_{i,j}$, *i.e.* the temporal displacement of FFD's control points $\mathbf{s}_{i,j}$ during the video, with a low-pass filter: $\mathbf{t}_{i,j}^f = \mathbf{t}_{i,j} * f$. We tested classical linear filters such as the Gaussian and mean filters and the nonlinear median filter. Another approach fits a cubic spline on the control points trajectories $\mathbf{t}_{i,j}$. The obtain trajectories $\mathbf{t}_{i,j}^r$ are regularized.

Figure 10 shows the results obtained with the different approaches applied on one control point trajectory (x component) of the paper sequence. Jittering occurs between frames 100 and 150 when the self-occlusion is extreme. Regularization and temporal filtering (linear and nonlinear) correct jittering well. However linear filters modify the vertex trajectories globally. They shift the mesh away from the ideal position. The nonlinear median filter and regularization attenuate high frequencies without degrading the registration accuracy. Retexturing is dramatically improved. A window size between 5 and 7 pixels gives the best visual results with the median filter. The latter might also be performed online with a few frames delay.

B. Retexturing and Self-Occlusions.

One possibility to retexture a surface is to retexture pixels $\mathbf{q}_i$ with the color of the nearest point on the template, after warping: $\mathcal{I}_i(\mathbf{q}_i) \leftarrow \mathcal{I}_i(\arg\min_{\mathbf{q}} \|\mathcal{W}(\mathbf{q};\mathbf{u}) - \mathbf{q}_i\|)$. This approach is not adapted with our self-occlusion representation. Pixels on the self-occlusion boundary have several points in their neighborhood warped to the same location. Artifacts might thus appear. We rather use the inverse warp *i.e.* the warp which maps a point $\mathbf{q}_i$ in the input image to the corresponding point $\mathbf{q}$ in the template. The warp we estimated might not be invertible. Using only the visible part, a piecewise homographic warp is used as an approximation. It is computed

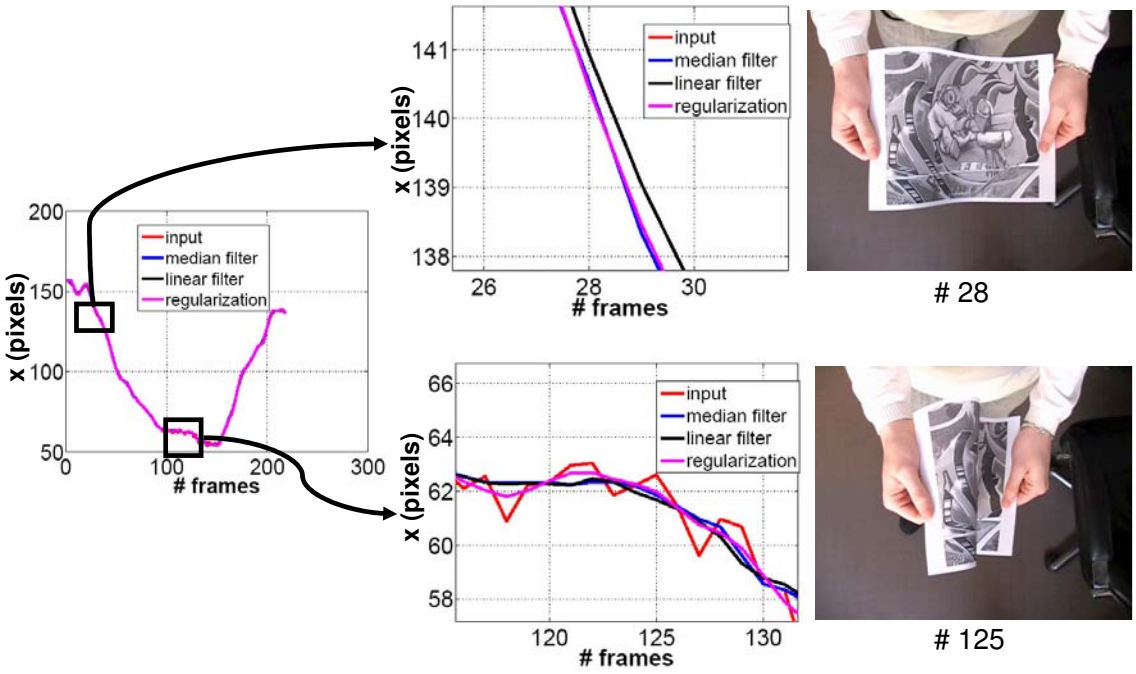


Fig. 10. Example of temporal smoothness on control point trajectories. The size of the window is fixed to 7 pixels for the filtering approaches. Left: the global trajectory. Top right: zoom on a trajectory portion without jittering and an associate frame. Note that the input trajectory and the median filtered one are undistinguishable on this part. Bottom right: zoom on a trajectory part with jittering and one associated frame.

between the control points $\mathbf{s}_{i,j}$ on the template and those evaluated in the input images: $\mathbf{s}_{i,j}+\mathbf{u}_{i,j}$. It is well adapted with our self-occlusion representation. Estimated vertices in self-occluded areas are compressed so no pixel belongs to the associated mesh tiles. The corresponding texture in the template is thus naturally avoided. Self-occluded surface retexturing is illustrated on figure 11.

VII. Experimental Results

We tested our approach on several videos with different kinds of surfaces (paper, fabric, rug). A 2-level pyramid is used. Various experiments show that the hierarchical framework is necessary to recover the warp at disocclusion but that additional levels do not significantly improve the registration. The image registration algorithm described in §V takes about 25 seconds per frame with our Matlab code for a texture image of size 316×378 (we take every pixel into account) and a grid of 64×76 control points on a 2 years old laptop. The major computation time is

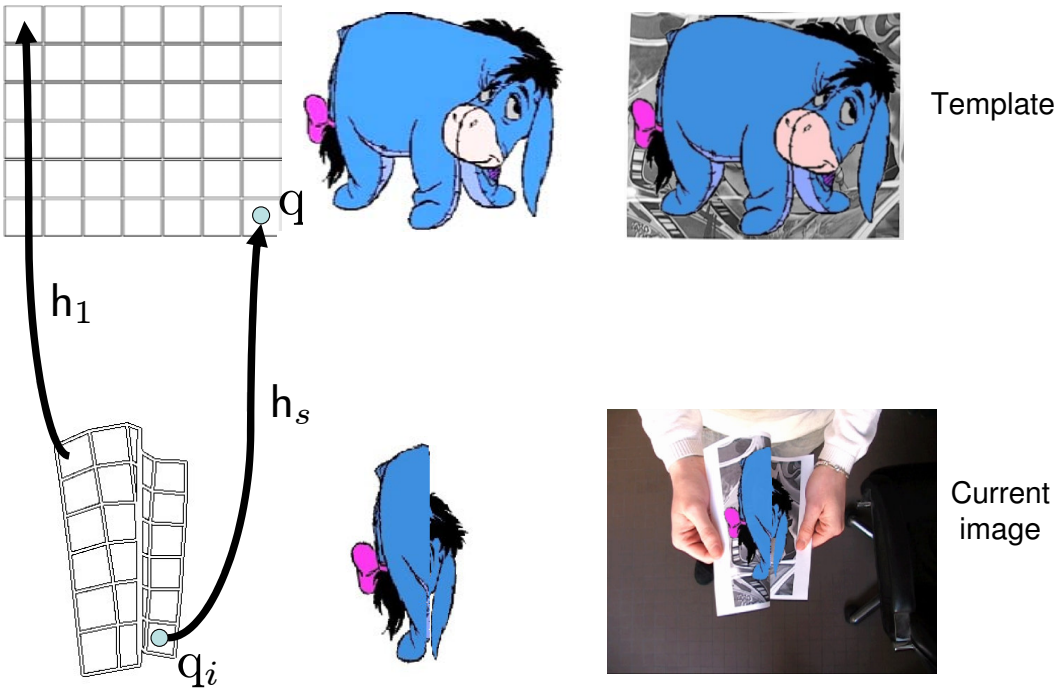


Fig. 11. Illustration of the piecewise homographic warp for self-occluded surface retexturing. Left: the reference grid and the one estimated with our registration algorithm described in §V. For each tile of the control point grid, an homography $\mathsf{h}_v$with $v \in \{1..s\}$ is computed. They map pixels from the actual image to the template. Note that the grids shown are very sparse for clarity reasons. Middle: The retexturing logo. The self-occlusion boundary is well handled with the piecewise homographic warp. Right: The template and the augmented input image $\mathcal{I}_i$.

spend to form the Jacobian matrix, the linear system is solve very efficiently due to the sparse solver: about 0.2 seconds.

A regular grid has been defined for visualization purposes. It is less dense than the one used to parameterize the warp. The detection of self-occluded regions is represented by retexturing in white the associated pixels on the texture image:

$$\mathcal{I}_0 \leftarrow \text{trunc}(\mathcal{I}_0 + 255\widehat{\mathcal{H}}; 255),$$

with

$$\text{trunc}(x; 255) = \begin{cases} x & \text{if } x \leq 255 \\ 255 & \text{otherwise.} \end{cases}$$

There are few thresholds used in the cost function: the weights λ_s and λ_f that control the influence of each term, and those associated with the self-occlusion detection module, rd and k.

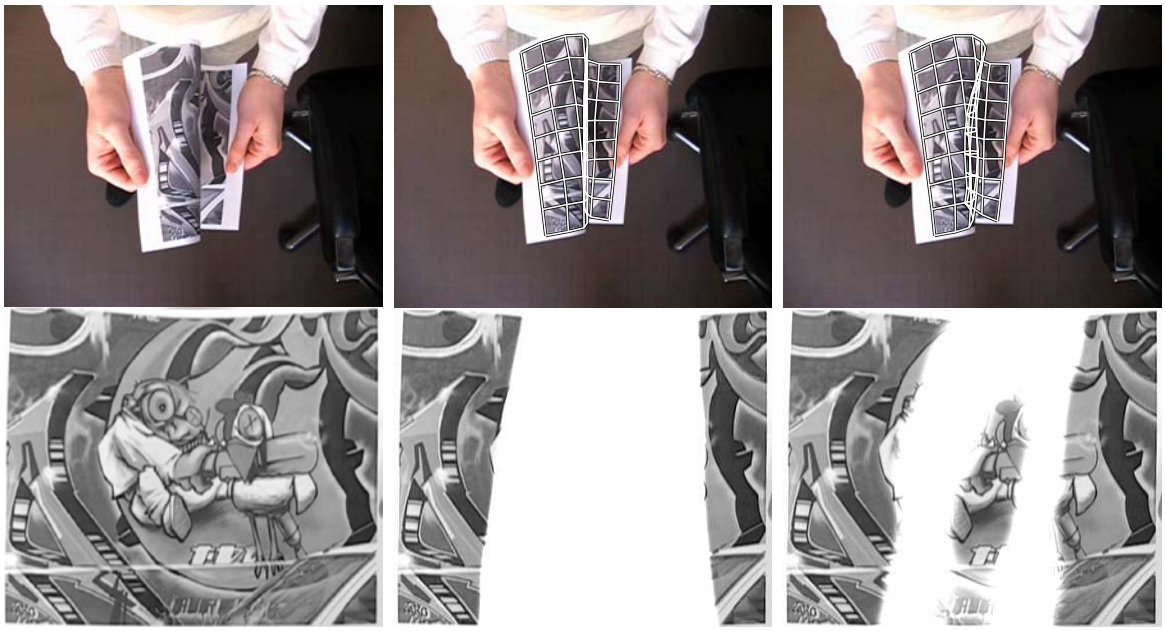


Fig. 12. **Example of image registration with and without the shrinker.** Self-occluded areas are shown in white. Left: the input $\mathcal{I}_i$ and the reference $\mathcal{I}_0$ images. Middle: erroneous registration results and self-occlusion detection without the shrinker. Right: successful registration results and self-occlusion detection with the shrinker.

Only the parameter λ_s related to the smoothness constraint is tuned for the different sequences, the others remaining constant. The value of λ_s is however not critical. Solutions have been proposed to automatically select this smoothness term [5] (they use a feature-based criterion). Their extension to a direct approach is not trivial and would involve additional study.

1) The Shrinker: As said above, loops and folds are non-admissible warp configurations. Without the shrinker, they naturally appear when the surface is extremely self-occluded, as shown in figure 12. The self-occlusion detection is wrong since the warp derivative is not null for the whole self-occluded area. This defeats the registration process. Adding the shrinker to the error function forces the warp not to fold but rather to shrink. Self-occlusion detection is then successful.

2) The Paper Sequence: We describe the processing of this sequence in details.

Description. Figure 14 shows some images of the first paper sequence. It has 215 frames of size 720×576. A self-occlusion appears near frame 50 and disappear near frame 180. It is extreme between frames 80 and 160. About half of the surface disappears. The size of the template $\mathcal{I}_0$ is 316×378. The number of control points is 4864, organized on a grid of size

64×76. Their large number makes the warp flexible. This is required in self-occluded areas. A strong 'smoother coefficient' $\lambda_s \approx 4000$ is used to make the warp well-constrained.

The Jacobian matrix. The size of the Jacobian matrix associated to the data term $\mathcal{J}_d$ is $119,448 \times 9,728$, the number of non zero entries is 3,815,936. For the smoother Jacobian $\mathcal{J}_s$, the size is $9,728 \times 9,728$ and the number of non zero entries is 123,672. Finally, the Jacobian of the shrinker is non null at the self-occluded pixels as seen on figure 13. The dotted area means that the shrinker is activated for the associated frames *i.e.* between frames 61 and 170.

Self-occlusion detection. We use the previously discussed derivative-based module. Figure 13 shows the evolution of the percentage of self-occluded pixels per frame. The self-occlusion map is not null between frames 46 and 184, which approximately matches our visual estimation. The self-occlusion detection appears to slightly anticipate them (due to the detection threshold being slightly over 0). The maximum percentage of self-occluded pixels is around 50 percent when the self-occlusion is the most extreme. This agrees with our visual estimation. Figure 13 also shows some detection examples. They are visually accurate.

Registration and retexturing. Registration of visible parts is accurate while the warp shrinks well in self-occluded areas. Recovering the true warp at disocclusion is done without misalignment. Jitters appear when the self-occlusion is extreme as previously discussed. They are attenuated through the temporal consistency priors described VI-A, but not totally removed. Surface retexturing for this sequence is very satisfying, see figure 14. The head, some paws and a body part of the 'Eeyore' logo disappear and reappear with the surface. Limited inaccuracies appear along the boundary of extreme self-occlusions.

3) Comparison with State-of-the-Art: We compared our approach with a classical robust direct registration method on the second paper sequence. It has 253 frames. The size of the images is 720×560. The surface is self-occluded between frames 89 and 191. The self-occlusion is extreme since more than half of the surface disappears.

The classical robust method uses a hierarchical Gauss-Newton algorithm to minimize equation (4) (see §II-E for more details). Results are shown on figure 15. The Huber function [16] is employed as the robust kernel. Results on figure 1 are obtained with this procedure. Our approach successfully deals with self-occlusions while robust methods fail since they do not constrain the warp to shrink in self-occluded regions and yield inaccurate registration of visible parts when the surface is extremely self-occluded. It is unlikely that they manage to keep track of the surface

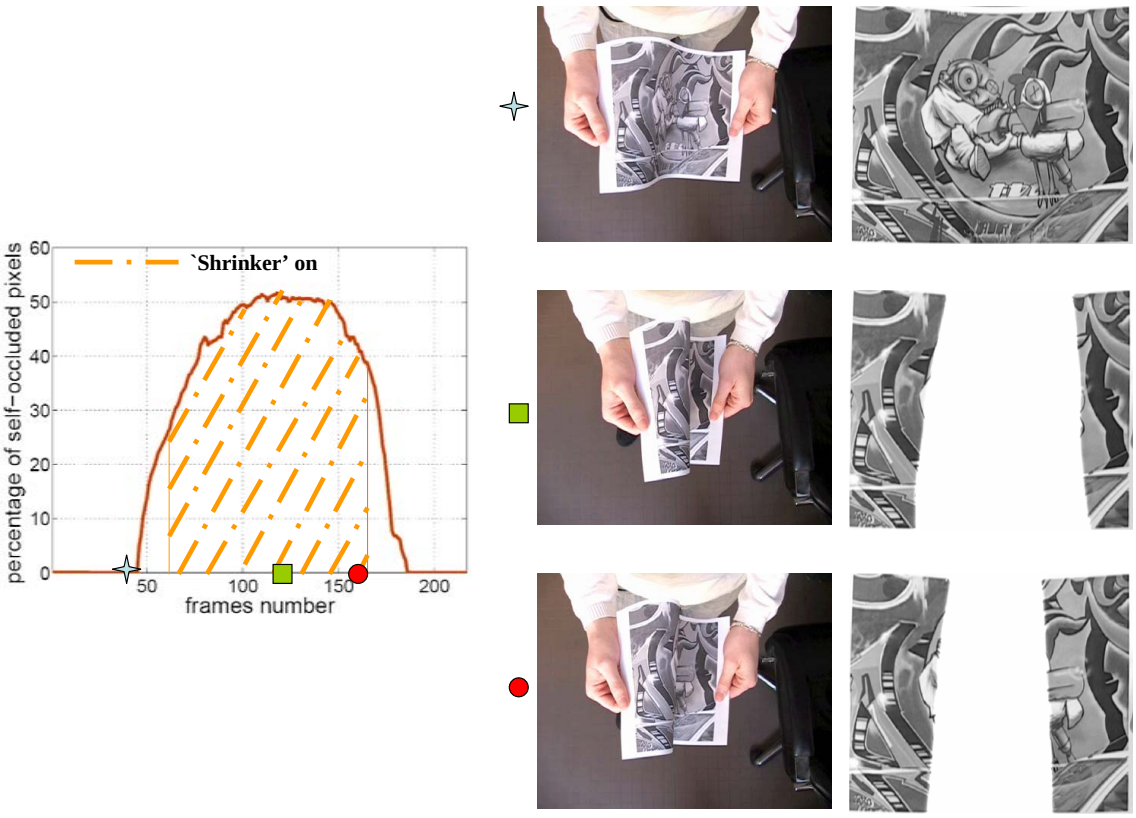


Fig. 13. **Self-occluded pixel detection:** Left: the percentage of pixels detected as self-occluded. Right: some input images and corresponding self-occlusion detection results.

at the disocclusion stage, in general.

We note that a re-initialization procedure such as non-rigid surface detection [24] might be used when the registration is lost. The proposed approach allows one to recover the warp at disocclusion without misalignment. A re-initialization procedure is required to recover the warp at disocclusion with classical robust approaches.

4) Self-Occluded Boundaries: We tested the behavior of the proposed approach for self-occluded boundaries on two sequences. The first one, called boundary I, has 204 frames and the second one, called boundary II, has 277 frames. For both sequences the self-occlusion appears at a surface boundary. They are strong ones *i.e.* approximatively a fourth of the surface is self-occluded.

Self-occluded surface boundaries are particular cases. Classical approaches and the one we

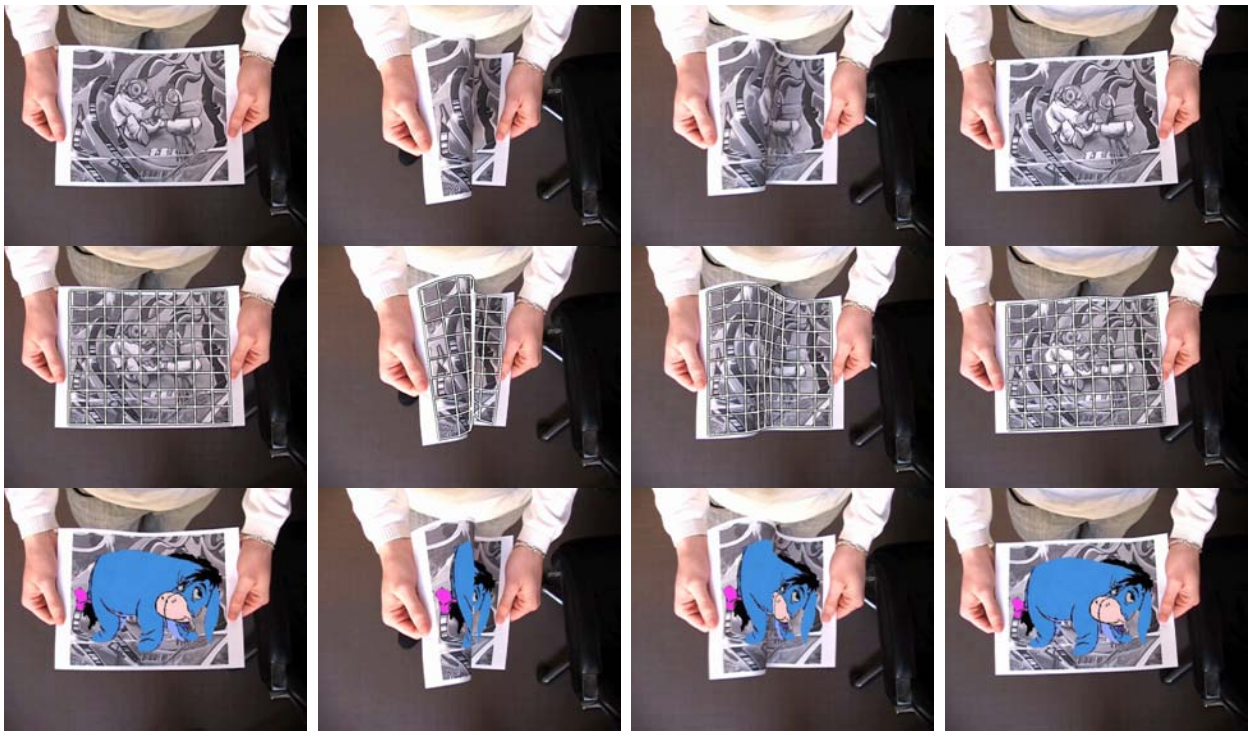


Fig. 14. **Registration and retexturing results on the first paper sequence.** Top: the input images. Middle: a grid illustrating the warp on the current frame. Bottom: surface retextured with the 'Eeyore' logo.

propose are both valid *i.e.* non-visible pixels are correctly detected and visible ones are correctly registered. However, with our self-occlusion approach, the occlusion boundary is tracked as shown on figure 16. Minimizing equation (9) enforces the warp to shrink along the self-occlusion boundary. The self-occlusion boundary is lost with the classical method since the smoother makes the warp nearly affine.

The Root Mean Squares errors on the intensity residual (expressed in intensity unit) for these two sequences (Top: boundary I, Down: boundary II.) are shown on figure 17. The values obtained with our algorithm are typical ones in spite of the strong deformations and extreme self-occlusion. For the boundary I and boundary II sequences, the mean RMS errors on the intensity residual are around 11 and 16 respectively. Small errors for the last frames, less than 10 for the boundary II sequence and around 12 for the boundary I sequence, show that the warp is correctly recovered at disocclusion.

5) External Occlusions: The proposed method is also robust to external occlusions. It is mainly due to the smoother and the external occlusion map. This is illustrated on the third paper

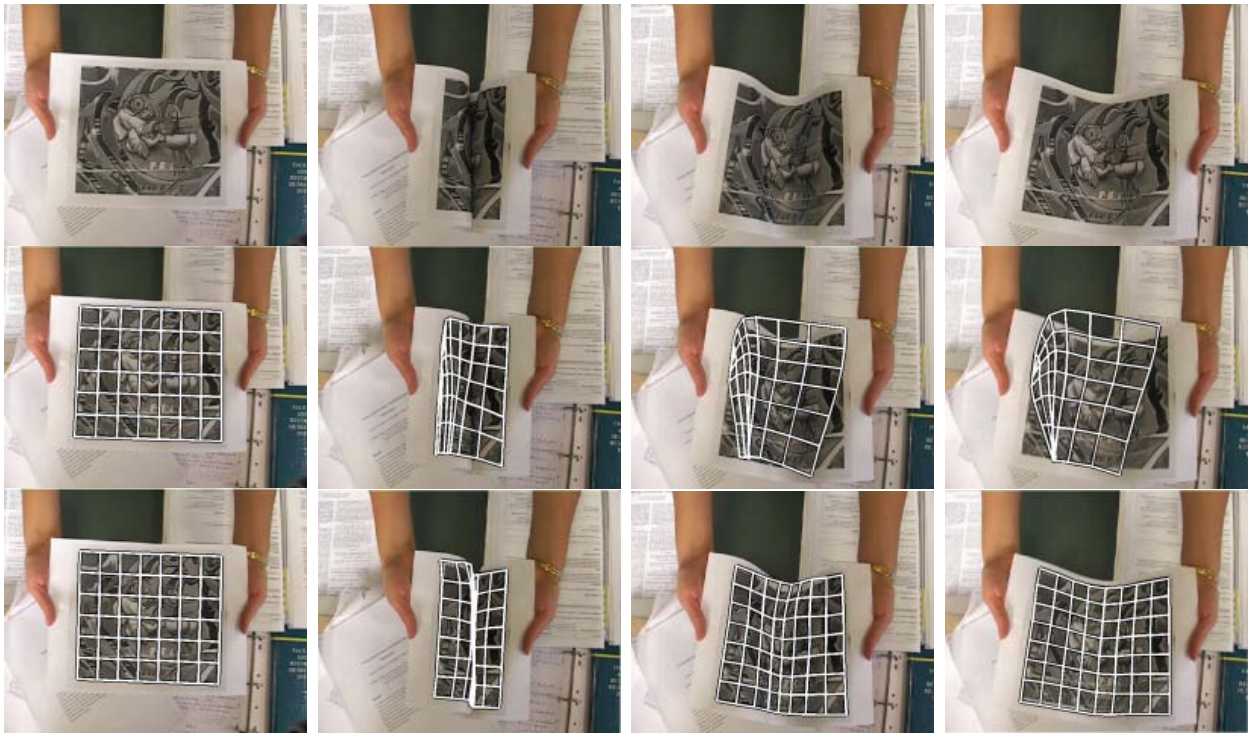


Fig. 15. **Registration results on the second paper sequence**. Top: the input images. Middle: erroneous registration results with a classical outlier rejection. Bottom: successful registration results with the proposed method.

sequence, see figure 18, for which both external and self-occlusion occur. It has 217 frames. External and self-occlusion appear and disappear simultaneously at frames number 37 and 160, respectively.

We classify each pixel as visible, externally occluded or self-occluded. For the visible pixels, the warp is smooth and $\widehat{\mathcal{H}}_v(\mathbf{q};\mathbf{u}) < 0.5$, for the externally occluded pixels the warp is also smooth and $\widehat{\mathcal{H}}_e(\mathbf{q};\mathbf{u}) > 0.5$ (*i.e.* green points in figure 18). Finally, for the self-occluded pixels, the warp is shrunk and $\widehat{\mathcal{H}}(\mathbf{q};\mathbf{u}) > 0.5$ (*i.e.* white points in figure 18).

6) The Comic Strip Sequence: This sequence has 255 frames. The page totally disappears at the end of the video, as shown on figure 19. The proposed algorithm convincingly retextures the small remaining visible part, with the 'Cars' logo. At the end of the sequence the warp is completely shrunk. The 'Cars' logo naturally disappears. We note that detection algorithms such as [24], allow one to augment a 'new page' when one totally disappears. Combining this approach with the proposed one could be powerful.

The RMS values for this sequence are shown in figure 17. The RMS error on the intensity

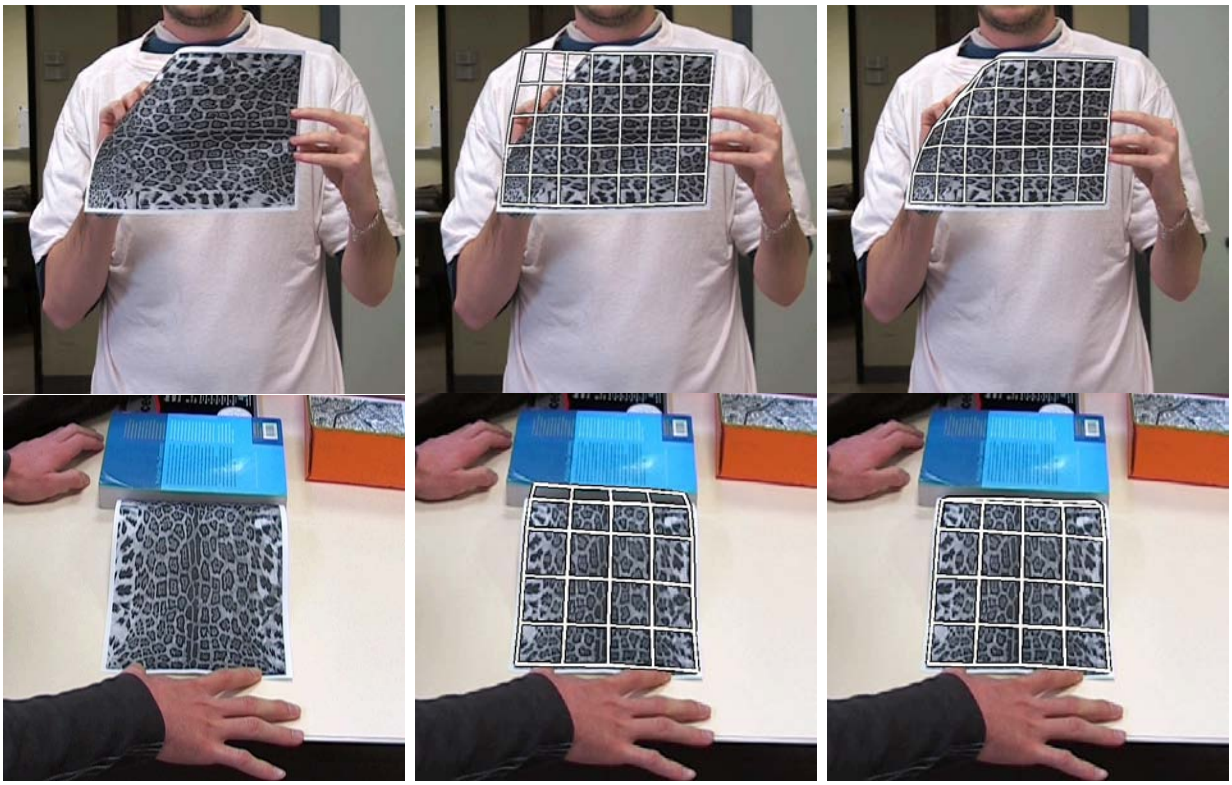


Fig. 16. **Self-occlusion on the surface boundary.** Left: input images. Middle: registration results with classical outlier rejection. Right: registration results with the proposed method.

residual between frames 100 and 200 is slightly over 10 while the surface is self-occluded up to 50 percents. At the end of the sequence *i.e.* from frame 210 the RMS is over 20. The remaining visible parts are less accurately registered since the surface is extremely self-occluded (more than 60% of the surface is hidden). Lighting variations also appear due to the strong deformations, partially explaining the large RMS values. For the last frames, the RMS error on the intensity residual does not exist anymore since the self-occlusion map covers the whole template.

7) Other Kinds of Surfaces: Experiments on a rug, figure 20, and on a fabric, figure 21, show that the specific framework we propose deals with self-occlusions for many kinds of surfaces.

8) Failure Cases: The proposed approach fails when the smoothness constraint, *i.e.* the bending energy, is violated. For example, videos of a paper being teared or those presented a T-shirt with several bumps defeat the method. Figure 22(a) and 22(b) illustrate the deformations that the proposed algorithm fails to register.

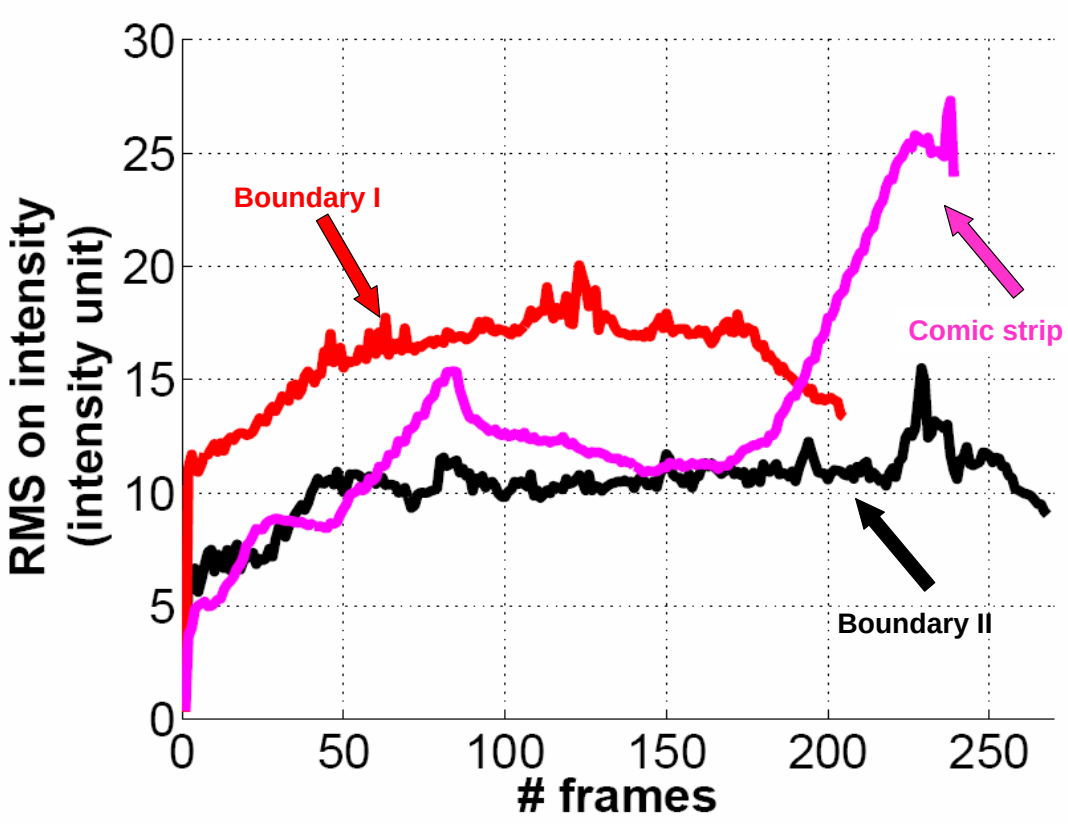


Fig. 17. **RMS on intensity residuals for different sequences.** Those are typical values.

VIII. Conclusion

We addressed the important and little explored issue of self-occlusions in non-rigid registration. A specific framework is proposed. The main idea is to constrain the warp to shrink in self-occluded regions while detecting them through the directional derivatives of the warp. Experimental results on real videos show that our approach successfully deals with extreme self-occlusions while classical robust methods fail. We applied our approach to surface retexturing, which gave very convincing results.

The illumination issue must be tackled in future work. Using an explicit local illumination model would probably cause instabilities: local intensity variations would both be explained by the warp and the illumination model. A solution might be to use invariant images as proposed in [25] and very recentlied to Active Appearance Models in [26]. The efficiency of the proposed approach could also be improved by using the ESM algorithm for the minimization. However,

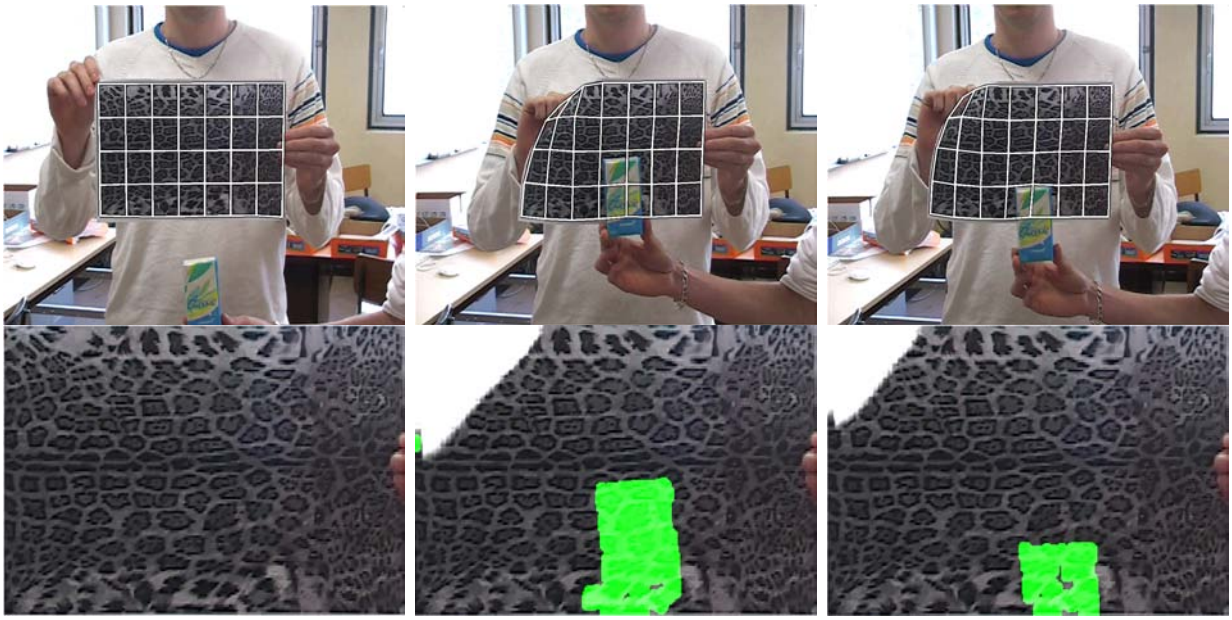


Fig. 18. **Registration results on the third paper sequence**. Top: registration results for a video with external and self-occlusions. Bottom: the template $\mathcal{I}_0$ with detected self-occlusions shown in white and external ones shown in green (or gray on the gray-scale printed paper).

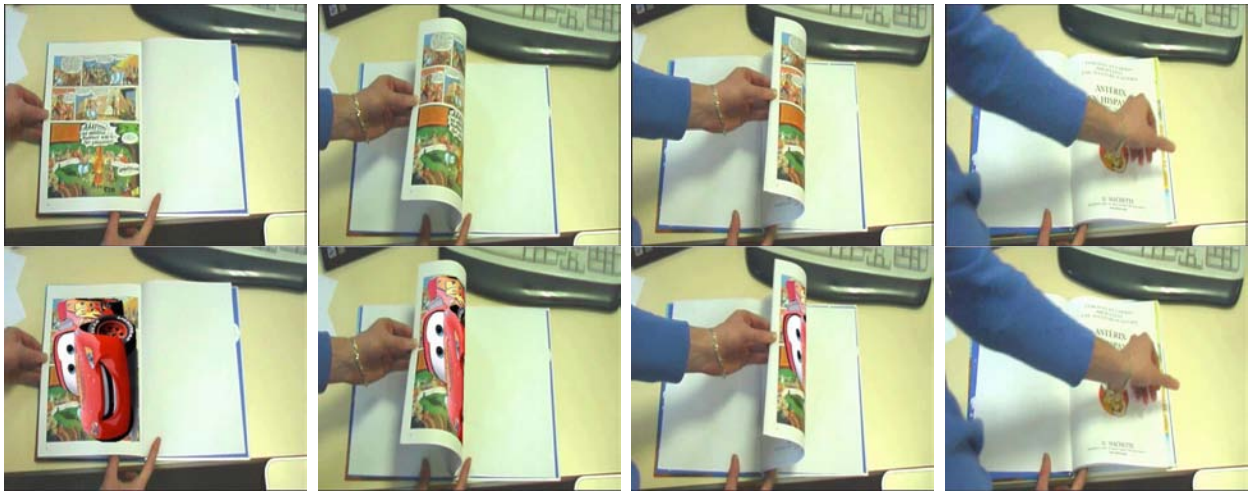


Fig. 19. **Retexturing results on the comic strip sequence.** Top: input images. Bottom: surface retextured with the 'Cars' logo.

Fig. 20. **Registration results on the rug sequence.** Top: the input images $\mathcal{I}_i$. Bottom: a grid illustrating the warp.

Fig. 21. **Retexturing results on the fabric sequence.** Top: input images $\mathcal{I}_i$. Bottom: surface retextured with a 'Mickey'.

the validity of the Jacobian approximation (caused by the non-groupwise warp) for strong deformations such as those presented in this article should be carefully studied.

REFERENCES

[1] http://www.cise.ufl.edu/research/sparse/cholmod.

[2] S. Baker, R. Gross, I. Matthews, and T.Ishikawa. Lucas-kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, 2003.

[3] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 2004.

(a) (b)

Fig. 22. Deformations that will defeat the proposed approach: (a) a paper being teared (b) a T shirt with several bumps. For these examples smoothness priors are not valid anymore.

[4] A. Bartoli. Groupwise geometric and photometric direct image registration. In *Proceedings of the British Machine Vision Conference*, 2006.

[5] A. Bartoli. Maximizing the predictivity of smooth deformable image warps through cross-validation. *Mathematical Imaging and Vision*, 2007.

[6] A. Bartoli and A. Zisserman. Direct estimation of non-rigid registrations. In *Proceedings of the British Machine Vision Conference*, 2004.

[7] S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004.

[8] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings of the European Conference on Computer Vision*, 1992.

[9] M. J. Black and P Anandan. Robust dynamic motion estimation over time. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1991.

[10] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 1989.

[11] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2000.

[12] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 1983.

[13] T. F. Cootes, S. Marsland, C. J. Twining, K. Smith, and C. J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *Proceedings of the European Conference on Computer Vision*, 2004.

[14] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Proceedings of the British Machine Vision Conference*, 2001.

[15] V. Gay-Bellile, A. Bartoli, and P. Sayd. Feature-driven non-rigid image registration. In *Proceedings of the British Machine Vision Conference*, 2007.

[16] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[17] K.P. Horn and G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[18] X. Huang, N. Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[19] S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *International Journal of Computer Vision*, 2007.

[20] V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 2005.

[21] J. Lim and M.-H. Yang. A direct method for non-rigid motion with thin-plate spline. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[22] W. C. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusions and rapid movements. In *Proceedings of the European Conference on Computer Vision*, 2006.

[23] E. Malis. An efficient unified approach to direct visual tracking of rigid and deformable surfaces. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2007.

[24] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 2007.

[25] D. Pizarro and A. Bartoli. Shadow resistant direct image registration. In *Proceedings of the Scandinavian Conference on Image Analysis*, 2007.

[26] D. Pizarro, J.Peyras, and A. Bartoli. Light-invariant fitting of active appearance models. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.

[27] M. Prasad, A. Zisserman, and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.

[28] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: Application to breast mr images. *IEEE Transactions on Medical Imaging*, 1999.

[29] M. Salzmann, S. Ilic, and P. Fua. Physically valid shape parameterization for monocular 3-D deformable surface tracking. In *Proceedings of the British Machine Vision Conference*, 2005.

[30] G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

[31] R. White and D. Forsyth. Retexturing single views using texture and shading. In *Proceedings of the European Conference on Computer Vision*, 2006.