Robust Deformation Capture from Temporal Range Data for Surface Rendering

Umberto Castellani[1], Vincent Gay-Bellile[2,3] and Adrien Bartoli[2]

[1] Computer Science Department – University of Verona, Italy

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

[3] LIST – CEA, LIST, Boite Courrier 94, 91191 Gif-sur-Yvette

Abstract

Imagine an object such as a paper sheet being waved in front of some sensor. Reconstructing the time-varying 3D shape of the object finds direct applications in computer animation. The goal of this paper is to provide such a deformation capture system for surfaces. It uses temporal range data obtained by sensors such as those based on structured light or stereo. So as to deal with many different kinds of material, we do not make the usual assumption that the object surface has textural information. This rules out those techniques based on detecting and matching keypoints or directly minimizing color discrepancy.

The proposed method is based on a planar mesh that is deformed so as to fit each of the range images. We show how to achieved this by minimizing a compound cost function combining several data and regularization terms, needed to make the overall system robust so that it can deal with low quality datasets. Carefully examining the parameter to residual relationship shows that this cost function can be minimized very efficiently by coupling nonlinear least squares methods with sparse matrix operators.

Experimental results for challenging datasets coming from different kinds of range sensors are reported. The algorithm is reasonably fast and is shown to be robust to missing and erroneous data points.

Keywords: Deformable Models, 3D Registration, Levenberg-Marquardt, Bending Energy, Motion Capture.

1 Introduction

Capturing time-varying surface deformations is an important problem which has been recently tackled thanks to the advances in realtime range sensors engineering [1, 2, 3]. A range sensor gives images where the intensity of a pixel is to be interpreted as its depth. Research on deformable models is active in fields such as computer graphics for 3D morphing and animation [4], medical image processing for data alignment and segmentation [5, 6], and computer vision for e.g. contour detection [7], face synthesis and expression recognition

[8].

The aim in this paper is to provide an effective system that allows one to capture the deformation of a physical surface being smoothly waved, such as the page of a book being turned, in front of a range sensor. As opposed to most of the existing methods [9, 10, 11, 12, 13], we do not use keypoint tracks [9, 10, 12] or optical flow [11, 13]. Instead, we only use the range images. This has the advantage that the method is independent of the appearance of the surface. In particular, those textureless surfaces that defeat existing methods are handled. We have to face the two problems of reconstructing the observed surface at each time instant, and registering the reconstructed surfaces. While most of the existing approaches bypasses the first problem [14, 15, 16, 17], the one we propose solves both simultaneously. We target applications such as 3D data compression, augmented reality and computer animation, all requiring an accurate registration of the range point clouds over time, as well as a reconstruction of the underlying surface. As an example, videos can be synthesized from the captured deformations and a user-provided texture map. The advantage of using range data is that real 3D deformations are obtained, as opposed to a 2D flow field in the monocular image registration case, see e.g. [18]. The input data is a sequence of range images and a coarse boundary of the surface of interest. The whole process is highly robust, filling in possible holes in the range data and detecting erroneous points, while establishing reliable registration even for flat areas that usually defeat shape based registration methods. We model the surface with a mesh as [18, 19] do for 2D image registration.

Our framework is implemented through two main lines of contributions extending our previous work [20]. First, we show that the problem is well-modeled by using a mesh that is deformed to fit each point cloud. This model allows us to write a cost function whose global minimum is the sought after solution. This cost function has several data and penalty terms. The data terms incorporate surface to data points distances and boundary information. Furthermore, it explicitly embeds a $\min$ operator, thus avoiding the traditional two steps in ICP-like (Iterated Closed Point) algorithms through distance transform. The penalty terms include spatial, i.e. surface-related, and temporal smoothness as well as inextensibility of the surface, if applicable. The data terms are robustified in order to deal with missing and erroneous points. Second, following [21], we use the Levenberg-Marquardt algorithm to minimize the error function. A careful analysis reveals that the Jacobian matrix involved in the normal equations to be solved at each iteration is highly sparse, for all the data and penalty terms we use. This makes tractable and fast the estimation of dense deformation fields.

Roadmap. Section 2 describes the state-of-the-art. The problem statement is given in Section 3, and the minimization strategy is described in Section 4. Experimental results are reported in Section 5. Finally, conclusions are drawn in Section 6.

2 Previous Work

We review existing work on surface motion capture and non-rigid shape registration.

Surface motion capture. Several works have been proposed in computer graphics on surface deformation modeling, especially for cloth motion capture [9, 10, 11, 12, 13]. In [10] a calibrated multi-camera setup is proposed to observe real-time cloth deformation. SIFT keypoints are used to identify a pattern printed on the cloth. The keypoints are tracked in the video thanks to a seed-and-grow approach adapted to the deforming geometry. In [9], quad marked surfaces are tracked by using a multi-baseline stereo system. Markov Random Fields models are used to introduce additional assumptions on the surface appearance and neighborhood consistency. In [12] the authors use a color-coded cloth texture for reliably matching circular features between different camera views. In [13], a direct estimation of the deformable motion parameters is proposed for range image sequences. The range flow is estimated by introducing depth constraints to the 2D displacements. Similarly in [11], the optical flow is computed from frame to frame using the depth.

Nonrigid shape registration. The registration of 3D point clouds is a challenging topic mainly tackled in the framework of ICP for rigid scenarios [21, 22]. However, researchers have recently addressed the case of deformable objects. Roughly speaking the literature on nonrigid registration can be divided into two main categories. The first one directly uses the point clouds. The second one abstracts the point clouds with some probabilistic model.

In [14] the authors propose to jointly compute the correspondences and the nonrigid transformation parameters between two point clouds. The algorithm uses the Expectation-Maximization (EM) paradigm. It combines the soft-assign and deterministic annealing within a robust framework. Thin-Plate Splines (TPS) are used for representing the spatial mapping. Nonrigid alignment is proposed in [23] to account for errors in the point clouds, obtained by scanning a rigid object. The authors use TPS to represent the nonrigid warp between a pair of views, that they estimate through hierarchical ICP [22]. Medical applications are proposed in [5, 6]. In [6], MR brain scan registration is performed by a modified Newton method over a hierarchical spline-based optical flow representation. In [5], a localized Radial Basis Function (RBF) is proposed, making a point to depend only on its neighboring centers.

Probabilistic approaches [15, 16, 17] are based on modeling each of the point sets by a kernel density function [24]. The (dis)similarity among such densities is computed by introducing appropriate distance functions. Registration is carried out without explicitly establishing correspondences. In [15], the authors propose a correlation-based approach [24] to point set registration by representing the point sets as Gaussian Mixture Models (GMMs). A closed-form solution for the L_2 norm distance between two Gaussian mixtures makes fast computation possible. In [17], registration is carried out simultaneously for several 3D range datasets. The method proposes an information-theoretic approach based on the Jensen-Shannon divergence measure. In [16], nonrigid registration is treated as a Maximum Likelihood (ML) estimation problem by introducing the Coherent Point Drift (CPD) paradigm. Smoothness constraints are introduced based on the assumption that points close to one another tend to move coherently over the velocity field. The proposed energy function is minimized with the EM algorithm.

The proposed approach. Our method draws on the strengths of some of the above mentioned approaches. It combines a deformable surface represented by a mesh with an ICP-like registration method that takes spatial and temporal smoothness into account, as well as the range image data and boundary information, required to prevent the computed surface to shrink or slide arbitrarily. The optimization is performed very efficiently using a distance transform of the range image.

3 Problem Statement

3.1 Surface Representation

The range sensor provides a sequence of range images that we interpret as 3D point clouds D_i with l_i points each:

$$D_i = \begin{pmatrix} d^x_{i,1} & d^y_{i,1} & d^z_{i,1} \\ \vdots & \vdots & \vdots \\ d^x_{i,l_i} & d^y_{i,l_i} & d^z_{i,l_i} \end{pmatrix} .$$

The reconstructed surface at time i is represented by a *geometry image* [25]. The model M is organized as three $R \times C$ matrices, representing the deformation of a regular flat grid. Each matrix is reshaped in a single vector of size $\mu = RC$, giving M_i as:

$$M_i = \begin{pmatrix} m^x_{i,1} & m^y_{i,1} & m^z_{i,1} \\ \vdots & \vdots & \vdots \\ m^x_{i,\mu} & m^y_{i,\mu} & m^z_{i,\mu} \end{pmatrix} .$$

In practice, the number of data points is much larger than the number of model points, i.e. $l_i \gg \mu$. Upon convergence, our algorithm determines for each model point if there is a corresponding point in the current point cloud. Points may be missing because of occlusions or corrupted sensor output. This approach has the advantage that it naturally gives the reconstructed surface by interpolating the mesh points. Point cloud registration is obtained by composing the deformation fields.

3.2 Cost Function

Our cost function combines two data and three penalty terms:

$$e(M) = e_g(M) + \lambda_b e_b(M) + \lambda_s e_s(M) + \lambda_t e_t(M) + \lambda_x e_x(M), \tag{1}$$

where λ_b, λ_s λ_x and λ_t are weighting parameters. Note that we drop the frame index i for clarity purposes, and denote M_i as M and M_{i-1} as $\tilde{M}$.

The data terms are used to attract the estimated surface to the actual point cloud. The first term e_g is for global attraction, while the second one e_b deals with the boundary. These terms must account for possible erroneous points by using robust statistics. The penalty terms are e_s, e_t and e_x. The two first ones respectively account for *spatial smoothness* and *temporal smoothness* e_s. The third one penalizes the *surface stress* and is related to the non-extensibility of the surface, and therefore to material properties of the surface.

This cost function is minimized in an ICP-like manner, as described in the next section. All the five terms are explained below in details.

Data term: global surface attraction. This term globally attracts the model to the data points in a closest point manner [22]. In order to avoid the traditional two steps arising in ICP-like algorithms, we explicitly embed the $\min$ operator in this data term, as suggested in [21]. Denoting E_M and E_D the sets of boundary points in the model and in the data, we get the following data term, integrating the model to data points matching step:

$$\sum_{m \in M \setminus E_M} \min_{d \in D \setminus E_D} \parallel d - m \parallel^2, \tag{2}$$

where d and m are $3-$vectors respectively representing a data and a model point. It is worth noting that, as opposed to [21], our unknowns are not the rigid motion parameters (i.e., the classical roto-translations) but correspond to the whole *nonrigid motion field* in M.

An *outliers rejection* strategy is introduced by defining a robust function w. Following the $X84$ rule [26], function w discards (i.e., it puts their residual to zero) those correspondences which residual error differs by more than 5.2 MAD (Median Absolute Deviation) from the median. The value 5.2 corresponds to about 3.5 standard deviations, which includes more than 99.9% of a gaussian distribution. Therefore, (2) is modified so as to get the following robustified data term:

$$e_g(M) = \sum_{m \in M \setminus E_M} w \left(\min_{d \in D \setminus E_D} \parallel d - m \parallel^2 \right). \tag{3}$$

Data term: boundary attraction. This term attracts boundary model points to boundary data points. It is defined in a similar manner to the global attraction term (3) except that the sum and min operators are over the boundary points:

$$e_b(M) = \sum_{m \in E_M} w \left(\min_{d \in E_D} \parallel d - m \parallel^2 \right). \tag{4}$$

Note that the detection of boundaries points in the range image is out of scope of the paper. In Section 5 we describe some ad hoc boundary detection methods for the proposed applicative scenarios.

Penalty term: spatial smoothness. This term discourages surface discontinuities by penalizing its second derivatives, as an approximation to its curvature. According to the definition of the geometry image [25], the model M is a displacement field parameterized[1] by (u, v) with $u = [1 \ldots R]$ and $v = [1 \ldots C]$, i.e., $M(u, v) = [M^x(u, v), M^y(u, v), M^z(u, v)]$. The spatial smoothness term can thus the taken as the surface bending energy:

$$e_s(M) = \int_{\mathcal{R}} \int_{\mathcal{R}} \left(\frac{\partial M^2}{\partial^2 u} \right)^2 + 2 \left(\frac{\partial M^2}{\partial u \partial v} \right)^2 + \left(\frac{\partial M^2}{\partial^2 v} \right)^2 \, du \, dv.$$

Using a finite difference approximation for the first and second derivatives [19], the bending energy can be expressed in discrete form as a quadratic function of M:

$$e_s(M) = \mathrm{vect}(M)^\top \mathcal{K} \mathrm{vect}(M), \tag{5}$$

where $\mathcal{K}$ is a $3\mu \times 3\mu$, highly sparse matrix, and $\mathrm{vect}(M)$ is the vectorization operator which rearranges matrix M to a vector.

[1] Recall that the model points lie on a grid.

Penalty term: temporal smoothness. This defines a dependency between the current and the previous point clouds, M and $\tilde{M}$:

$$e_t(M) = \parallel M - \tilde{M} \parallel^2 . \tag{6}$$

This makes the surface deformation smooth over time and can be used within a sequential processing approach. It is obviously not used on the first frame of the sequence.

Penalty term: non-extensibility. This term discourages surface stretching. It favors the mesh vertices to preserve their distance with their local neighborhood [27]:

$$e_X(M) = \sum_{m \in M} \sum_{k \in \mathcal{N}(m)} \left(\parallel m - k \parallel^2 - L_{m,k}^2 \right)^2 , \tag{7}$$

where $L_{m,k}$ are constants which are computed at the first frame after robust initialization and $\mathcal{N}(m)$ is the neighborhood of the mesh vertex m, with $\#\mathcal{N}(m) = 8$.

4 Minimization Procedure

The above described cost function (1) is a sum of squared residuals nonlinearly depending on the unknowns in M. In order to minimize this cost function, we use a nonlinear least squares optimization algorithm, namely the Levenberg-Marquardt (LM) algorithm. We extend the LM-ICP approach proposed in [21] to deformable objects. LM requires one to provide the partial derivatives of the residuals through a Jacobian matrix.

The Hessian matrix[2] $H = J^\top J + \lambda I$ must be inverted at each LM iteration, the problem

[2] We use ‘Hessian matrix’ for the damped Gauss-Newton approximation to the true Hessian matrix.

is not tractable if the number of model points is too high (if the deformation field is too dense). The Jacobian matrix stacks the individual Jacobian matrices for all the five terms in the cost function:

$$J^\top = \begin{pmatrix} J_d^\top & J_b^\top & J_s^\top & J_t^\top & J_x^\top \end{pmatrix}.$$

where $J_d^{\mu\times 3\mu}$, $J_b^{E_B\times 3\mu}$, $J_s^{3\mu\times 3\mu}$, $J_x^{\xi\times 3\mu}$, $J_t^{\mu\times 3\mu}$, are related to the global attraction, boundary attraction, spatial smoothness, temporal smoothness and non-extensibility terms respectively, and $\xi = \#\mathcal{N}(M)$. For instance, a grid with 15×20 points has a Jacobian matrix with $3760 \cdot 900$ elements ($\mu = 300$, $E_B = 66$, $\xi = 2194$). It is worth noting that the most expensive Jacobian computation is J_d since it requires the estimation of closest points at each iteration. As suggested in [21], we solve this step with the *distance transform*: the distance between data points and a set of points lying on a discrete volumetric grid is pre-computed by defining a proper *distance field* data structure. In this fashion, the closest point computation in carried out in one step for all points [21]. Therefore, the Jacobian matrix J_d is estimated by finite differences on such *distance fields* which remain constant through the minimization.

One advantage of the proposed approach is that the Jacobian matrix J is very sparse. We thus use the sparsity to speed up each iteration using the technique in [28]. In particular, we use the sparse Cholesky factorization package [29] included in the Matlab routines[3]. Figure 1 shows a plot of the Jacobian matrix $J^{3760\times 900}$ (left), and the corresponding Hessian matrix H (right). Dark points show non zero entries. The sparsity of the Jacobian and Hessian

[3] More precisely, we use the ‘mldivide’ matlab function

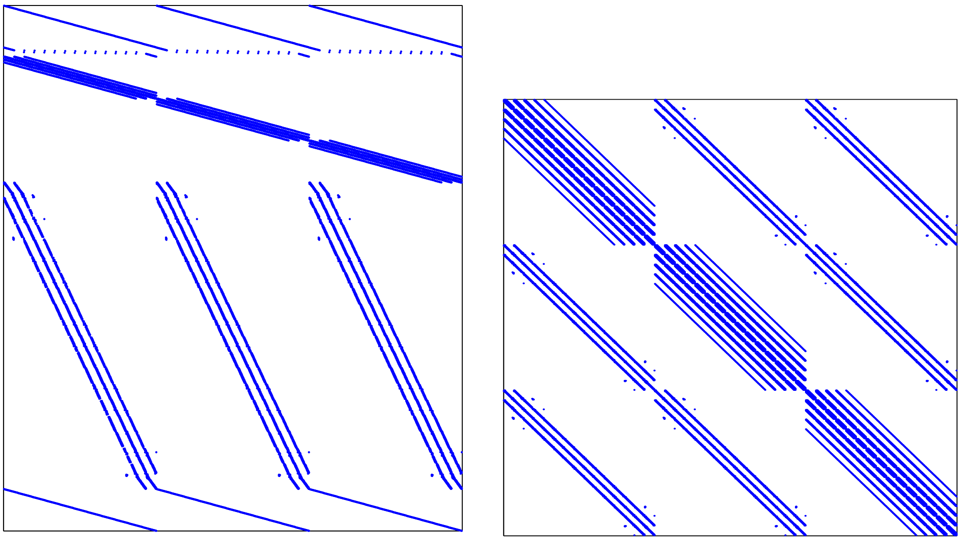


Figure 1: Example of Jacobian (left) and Hessian (right) matrix patterns: dark means non-zero. The size of these matrices is $3,760 \times 900$ and 900×900, so they respectively have $3,384,000$ and $810,000$ entries. They however have only $25,626$ and $46,716$ non-zero entries.

matrices is clearly evidenced.

5 Experiments

Two kinds of experiments have been set up. In the first one, a structured-light 3D scanner[4] is used for scanning different deformations of a paper sheet. In the second experiment, the sensor is a passive-stereo system[5] which allows us to acquire a sequence of 3D point clouds

[4] Data courtesy of Johnny Park from Purdue University (http://web.ics.purdue.edu).

[5] Data courtesy of eVS (http://www.evsys.net).

in realtime. The deformation of a portion of a blanket is modeled.

Initial conditions determine an estimate for both the model position, and the grid size. In practice, a correct starting grid allows LM to converge, as well as to determine the parameters $L_{m,k}$ in (7). In the absence of any texture information, i.e., in experiment 1, we detect them as strong depth discontinuities in the range image. In experiment 2, we use the intensity information to segment the boundary. The result is however quite noisy, but our system handles it thanks to its robustified data terms.

Experiment 1: paper sheet from a structured-light scanner. Several scans have been carried out while bending the paper. The sensor provides accurate and high-resolution 3D point clouds. The initial orientation of the grid is estimated by fitting a plane to the data. By projecting the points to the plane, both the grid size and boundaries are easily computed. There is no temporal dependency between the scans since they show totally different paper deformations. The temporal smoothness constraint e_t is thus inhibited.

Figure 2 shows three examples. Images on the top row visualize the model and data before surface fitting. Boundary points are highlighted. In the first example (Figure 2.a), the deformation is mainly on the horizontal boundary. In the second one (Figure 2.b), the paper is bent from the top-right to the bottom-left corner. In the third one (Figure 2.c), the deformation is basically spread to the whole paper. Images on the central row show the result of our robust fitting. The registration is accurate for both the interior points and the boundary. The recovered meshes are smooth as expected. Finally, three synthetic reconstructions are shown on the bottom row. Any texture can be overlaid onto the model, for rendering realistic paper deformation from arbitrary points of view. Figure 3 shows the paper sheet reconstruction with different textures.

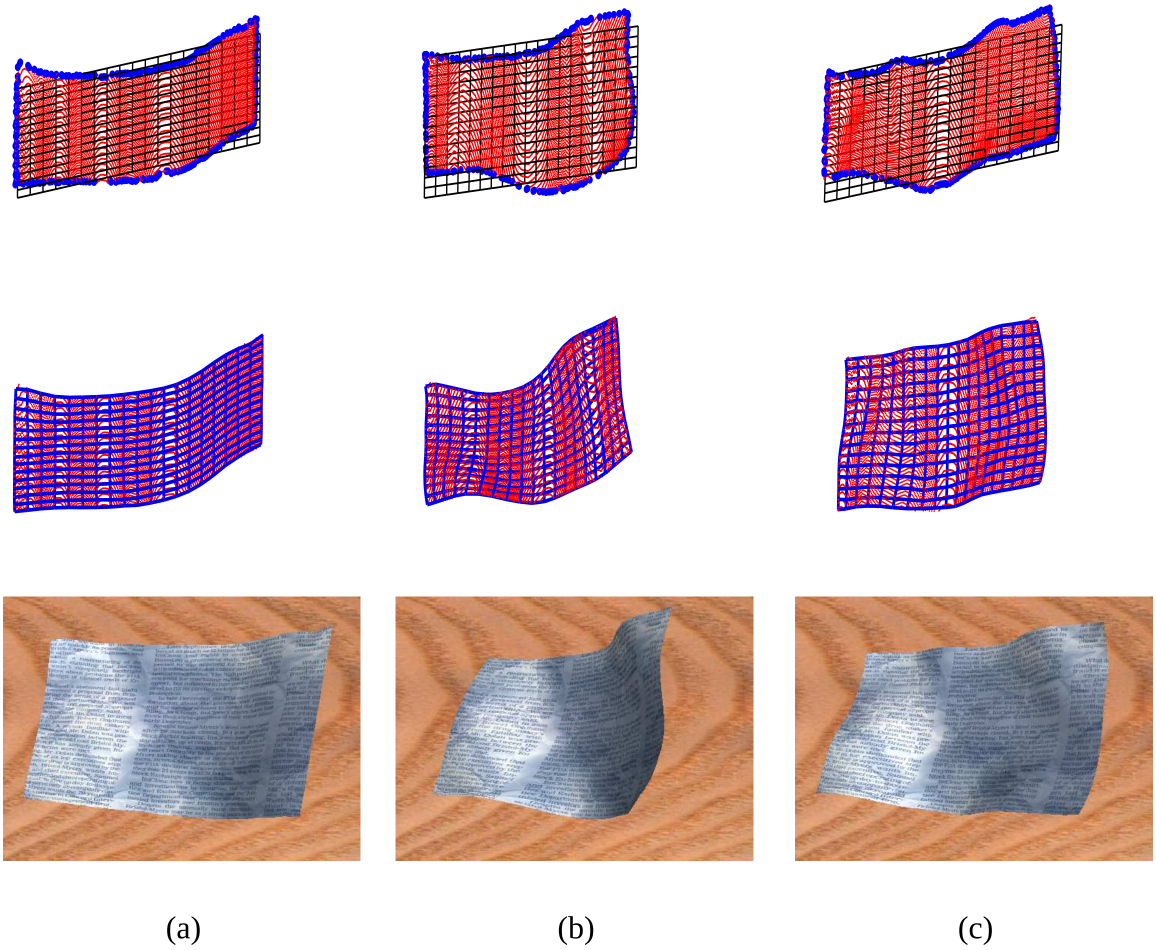


Figure 2: The paper sequence. Three examples of point clouds with the grid mesh superimposed at the starting position (top) and after model fitting (middle). Boundary points are evidenced. The reconstructed model is shown with a new texture (bottom).

Finally, since the same grid is fit on all the images, the correspondences between differ-

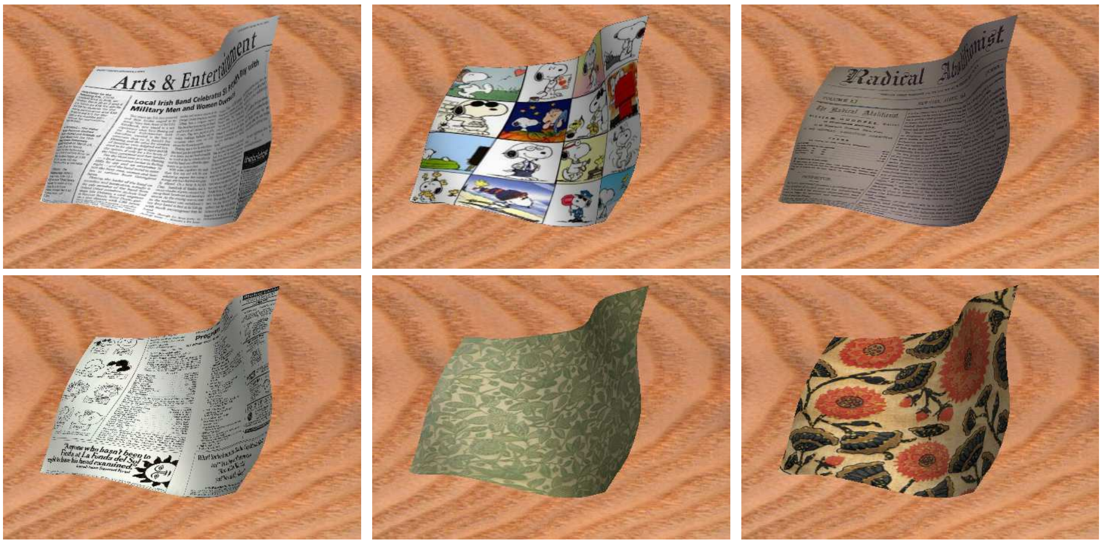


Figure 3: Textured views. Once the surface deformation is captured, any image can be used as a texture.

ent frames are recovered as a by-product (i.e., the corresponding points are those lying on the same grid position). In this manner, it is possible to synthetically interpolate the intermediate frames between two or more reference images. Figure 4 shows some intermediate frames between the actual scanned images shown in Figures 2.a, 2.b and 2.c respectively. Here, intermediate grids are recovered by simple linear interpolation between corresponding mesh vertices. More sophisticated interpolation schemes incorporating the non-extensibility penalty e_x could be used so as to make the surface behave similarly to a real paper sheet.

Experiment 2: blanket from a stereo system. A long sequence of point clouds is acquired for the second experiment. The sensor acquires the images at 25 FPS (frames-per-

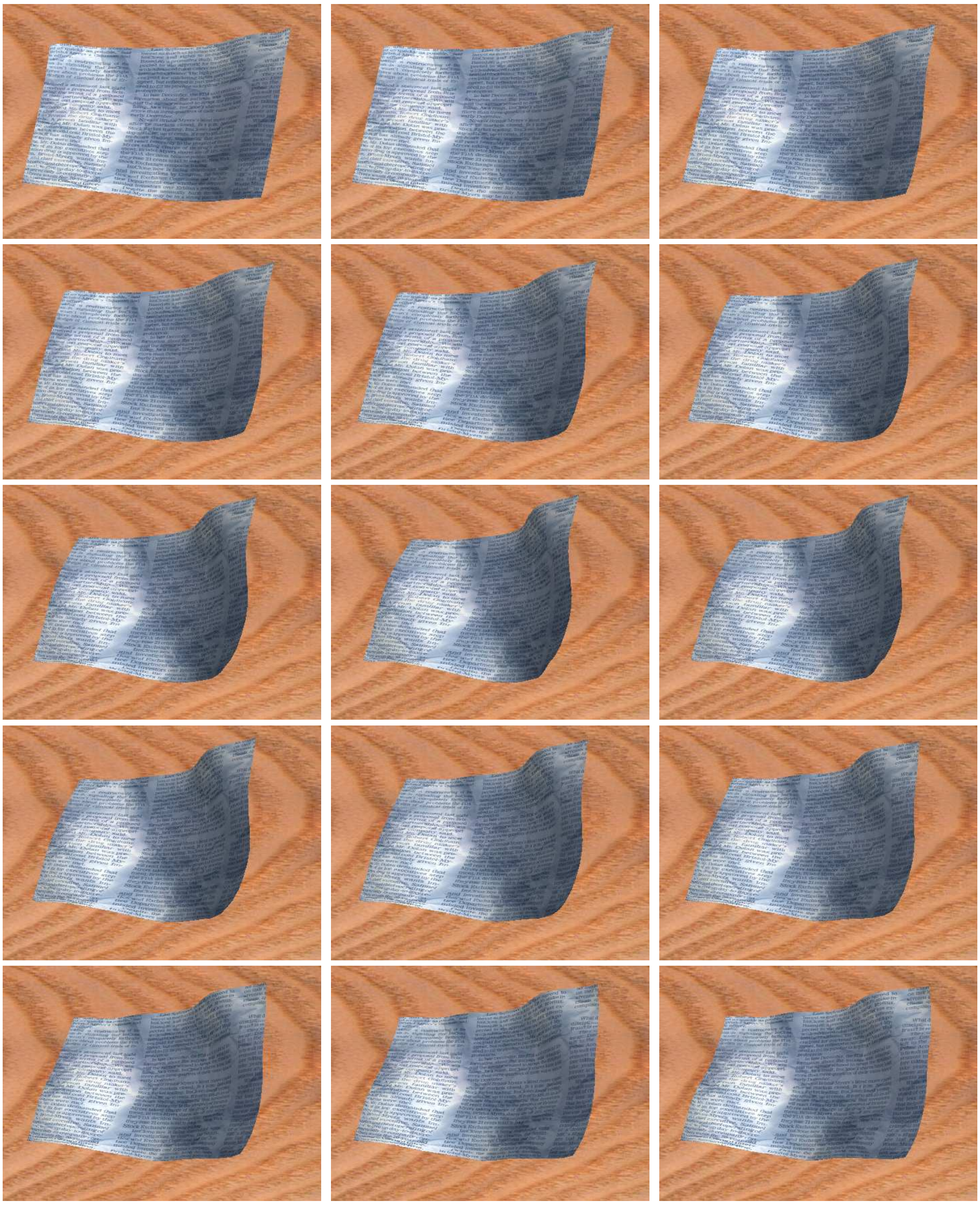
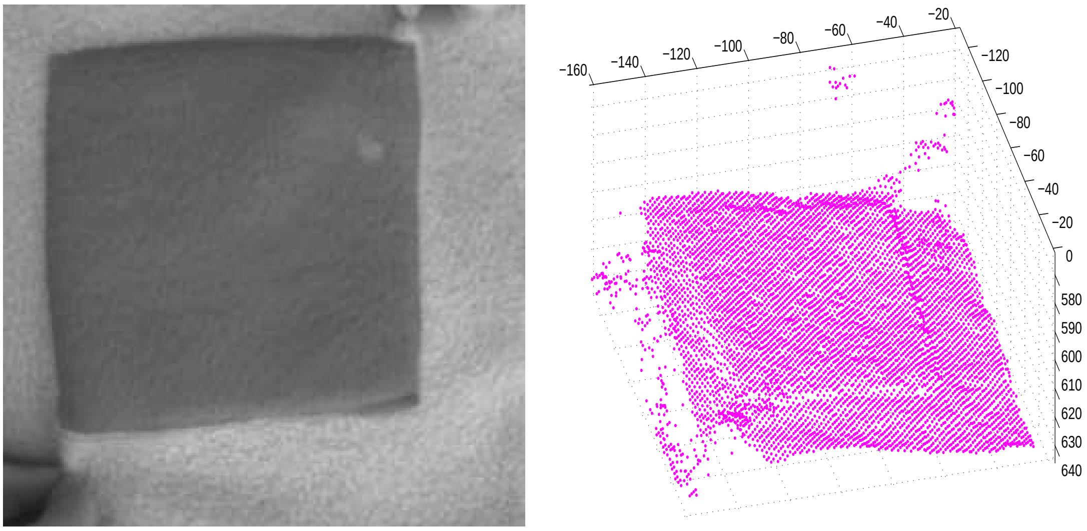


Figure 4: Synthetically interpolated views. Images are generated by linear interpolation between Figures 2.a, 2.b and 2.c respectively.

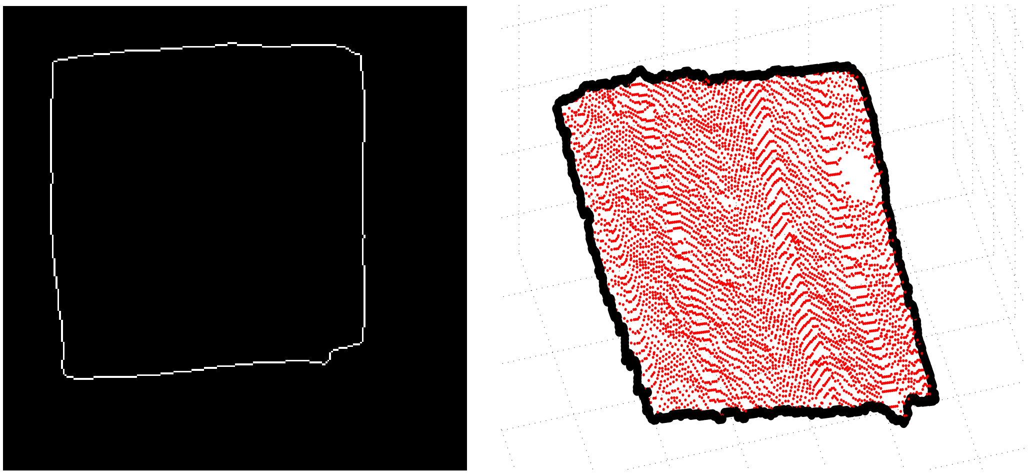
second), and provides both intensity (i.e., 2D) and 3D information. The quality of the 2D images is very low, and the range data is noisy. Moreover, the sensor can operate only on a very limited field of view (i.e., $30\ cm^3$). We use a blanket as the object of interest. Figure 5.a shows a picture of the blanket. Note that although 2D correspondences are recovered by the stereo system between the left and right images, the 2D correspondences between the frames contiguous in time are not reliable due to the distortions introduced by the surface deformation. In this experiment, we aim at observing the blanket deformation only on the portion delimited by the dark square. Figure 5.b shows the 3D point cloud. There are many spurious points especially on the boundaries, and the scene is not easily recognizable. We use the intensity image[6] for selecting automatically our region of interest (i.e., the dark



(a) (b)

Figure 5: The blanket sequence: intensity image of the blanket (a) and the 3D point cloud (b).

square), from which we recovered both the 3D data and the boundary. Figure 6.a shows the image-boundary extracted by standard image processing techniques, while Figure 6.b depicts the 3D data (i.e., the selected point cloud and 3D boundary). The sequence is made of 100 point clouds. Model initialization is carried out for the first cloud only. Each iteration uses the output of the previous one as an initial condition. Figure 7 shows a selection of the output sequence. For each frame, is visualized: 1) the intensity image, with the extracted 2D boundary and the 2D projection of the estimated model, and 2) the point cloud - after the region of interest selection, evidencing both the 3D boundary and the grid. The blanket is handled from the bottom-left and upper-right corners, respectively. On the early frames,



(a) (b)

Figure 6: Data extraction: 2D boundary (a) and selected 3D data (b). 3D boundary is highlighted with dark color.

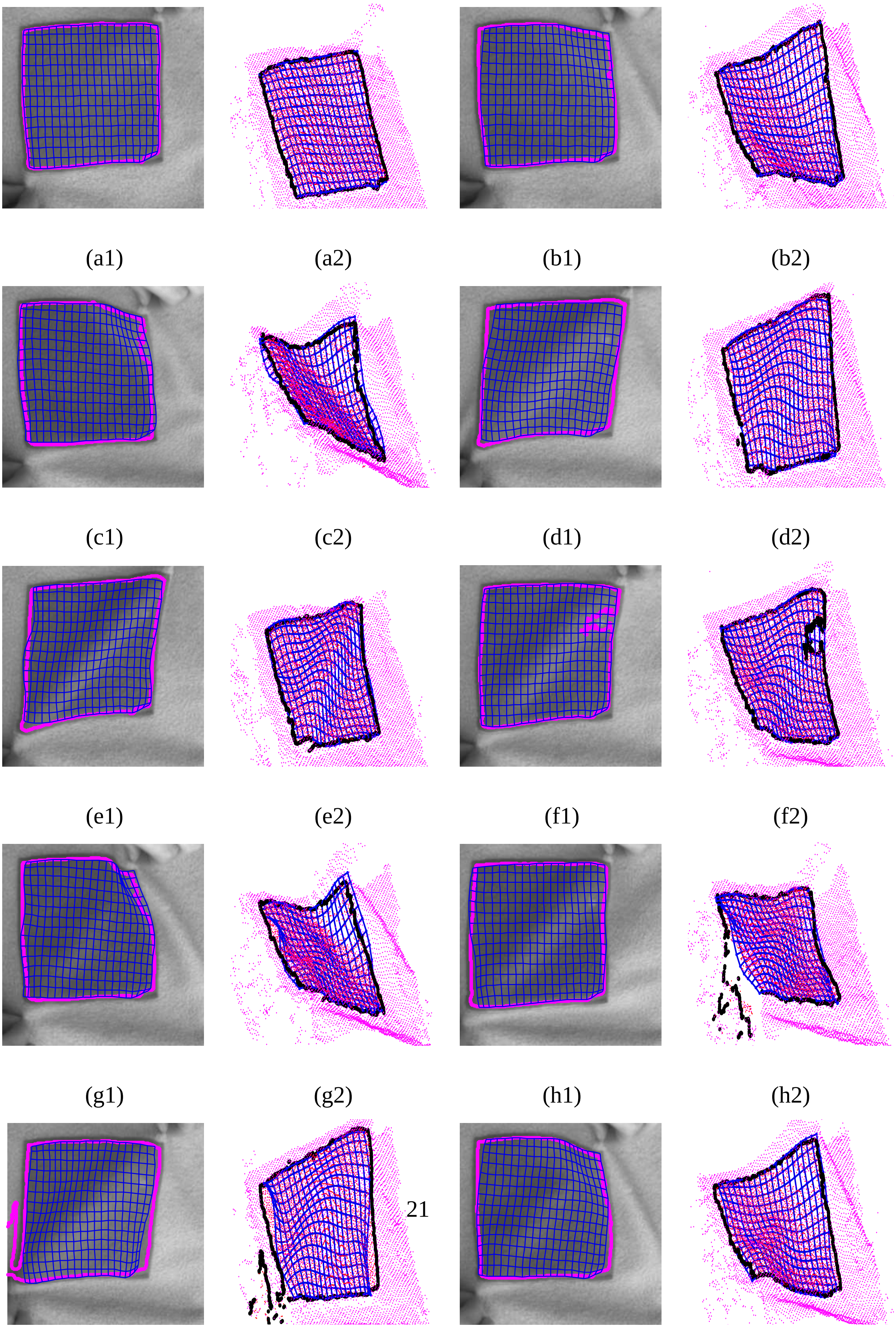
[6] The intensity is the left image of the stereo-pair, which is associated to the disparity map. Indeed, there is a mapping between the 2D and 3D information. Note that we do not use intensity information for fitting.

the blanket is gradually bent toward the square center, then it is strongly stretched, moving the corners far from each other. Finally, in the late frames, random deformations are generated especially around the corners. Some frames are particularly challenging. In frame (c) a strong shrinking is evidenced on the top-right corner. In frame (f) a wide hole appears on the top-right side. In frames (h) and (i) data boundaries are clearly wrong on the bottom-left side. Results are satisfying since the fitting is correct for the whole sequence. The mesh grids are well superimposed on data points maintaining a smooth shape. Nevertheless, the projections of the grids to the 2D images confirm the accuracy of the registration. Finally, after the mesh fitting, a dense set of accurate deformations is available. We used them to synthesize a movie rendered as if it was projected to a deforming screen. We project every frame of the movie to a model of our sequence[7]. Some frames are shown in Figure 8.

Performance evaluation. For both experiments, a model of size 15×20 is used. We have seen that a higher value of λ_b is necessary (i.e., $\lambda_b = 1.5$) for a correct convergence of the algorithm to the optimal solution. The other terms are set almost equally to 1. The distance transform parameters are important: the size of the voxels trades off speed and result accuracy. Here, we have divided the volume into $36 \times 36 \times 18$ voxels.

Table 1 shows the running times of both the proposed experiments. For each experiment we highlight the time spent for pre-processing (i.e., grid pre-alignment and the computation of constants $L_{m,k}$), the total energy computation, the mean number of LM iterations, the total

[7] We loop over the extracted 100 models.



(a1) (a2) (b1) (b2)

(c1) (c2) (d1) (d2)

(e1) (e2) (f1) (f2)

(g1) (g2) (h1) (h2)

(i1) (i2) (j1) (j2)

time for each frame and the total time of the whole experiment. Note that the computational cost of one LM iteration is very similar for both experiments. This is due to the fact that it depends only on the chosen grid size, being independent on the amount of input data. The computational cost of pre-processing instead depends of the total number of points and thus the first experiment takes a longer time.

The method has been implemented in Matlab on a Pentium M 1.86GHz. Note that since the pre-processing is carried out only for the first frame of the sequence it strongly affects the total time of the first experiment (for which only single frames are processed) while being less relevant for the second experiment.

Experiment	Pre-process (s)	Energy (s)	# LM iter	Time per frame (s)	# frames	Total time (s)
Exp. 1	19.36	2.22	11	24.42	1	43.78
Exp. 2	15.24	2.34	7	16.34	100	1649.34

Table 1: Performances of the main steps of the proposed framework. Running times (sec.) and iterations are mean values.

6 Discussion

We proposed a new approach for capturing the deformation of 3D surfaces from range image or 3D scans. As a surface model, we deform a generic geometry image, which is aligned with the observed data points. A cost function is devised that combines a priori information, such as spatio-temporal smoothness, and observations. Both non-extensibility and boundary

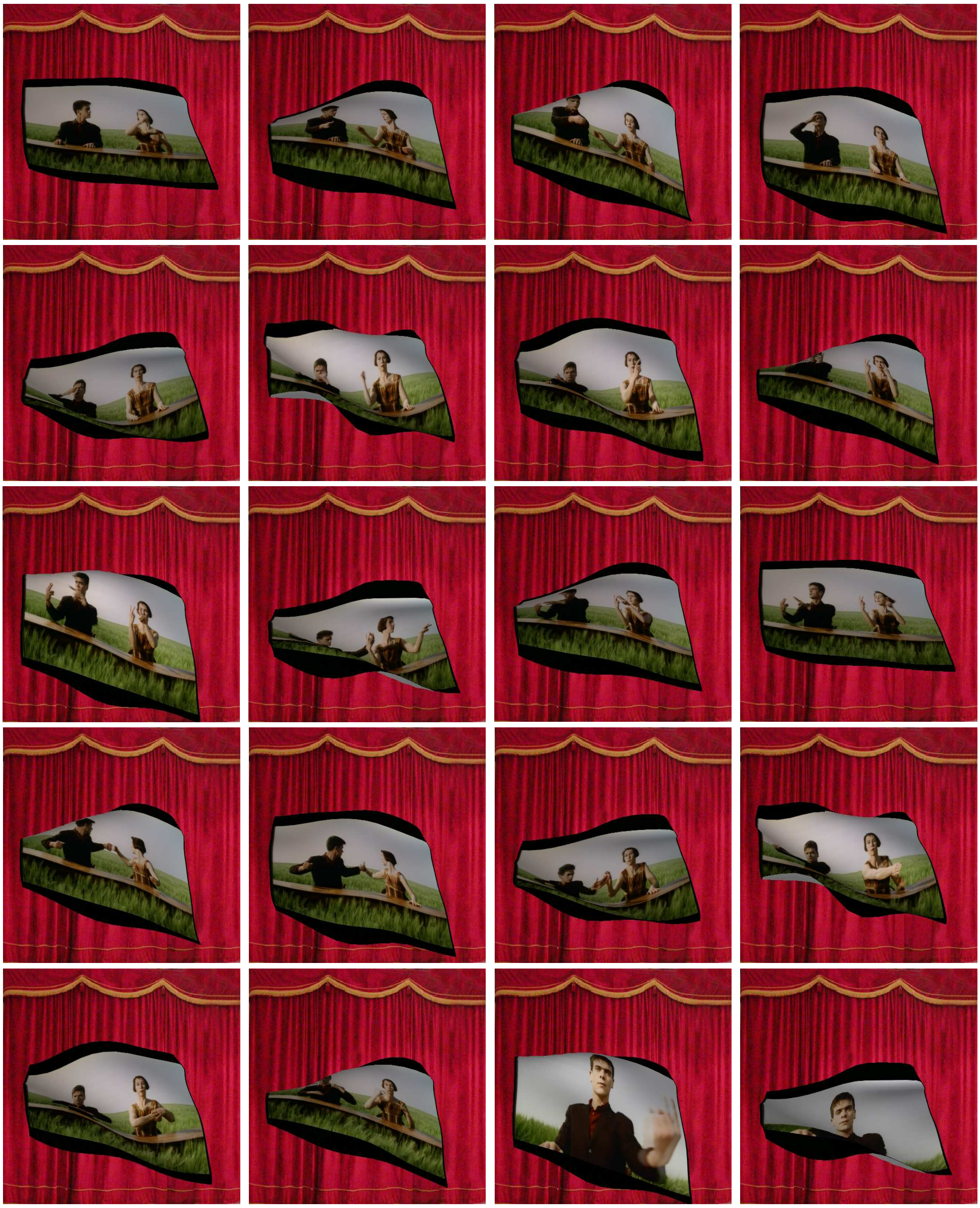


Figure 8: Synthesized movie: some selected frames. Each frame of the movie is projected to the reconstructed model by simulating a deforming video screen.

attraction terms are crucial for disambiguating this intrinsically ill-posed problem. The optimization phase is solved with the Levenberg-Marquardt algorithm, while taking advantage of the sparsity of the Jacobian and Hessian matrices.

Results are promising since the performances are satisfying for the analyzed cases. The method has been tested onto two kinds of datasets, thereby evidencing the versatility in dealing with different sensors. In the first experiment, the source data was accurate and the estimated models was according to what we expected. In the second experiment, a whole sequence of 3D point clouds has been processed. This has allowed us to capture cloth deformations without using any markers nor special projected patterns. Although data was very noisy, especially around the boundary, the method performed robustly. We discussed the behavior of our algorithm in the presence of holes and broken boundary. Moreover, since the captured deformation depends only on the size of the chosen grid, the proposed method allows one to use an arbitrary level of deformable details in the captured surfaces. We also highlight that the method is easy to implement and that it is reasonably fast. Finally, some graphical results were shown for generating new synthetic deformations, and for changing the appearance of the original surface, as well as for synthesizing a deforming video. This shows the usefulness of the method for computer animation. Several applications are made possible by the proposed method, including $i)$ cloth motion capture, with no marker or pattern, $ii)$ dense augmented reality for special effects such as a liquid spreading on a deformable surface and $iii)$ surface completion and interpolation in the presence of missing or erroneous frames. The main limitation regards the hypothesis to work on planar-like surfaces. The extension to shapes with more complex topology will be addressed in future work.

References

[1] T. P. Koninckx, A. Griesser, and L. Van Gool. Real-time range scanning of deformable surfaces by adaptively coded structured light. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, pages 293–300, 2003.

[2] S. May, B. Werner, H. Surmann, and K. Pervolz. 3d time-of-flight cameras for mobile robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 790–795, 2006.

[3] S. Rusinkiewicz, O. H. Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3):438–446, 2002.

[4] M. Alexa. Recent advances in mesh morhping. *Computer Graphics Forum*, 21(2):173–196, 2002.

[5] M. Fornefett, K. Rohr, and H. S. Stiehl. Radial basis functions with compact support for elastic registration of medical images. *Image and Vision Computing*, 19:87–96, 2001.

[6] B. C. Vemuri, S. Huang, S. Sahni, C. M. Leonard, C. Mohr, R. Gilmore, and J. Fitzsimmons. An efficient motion estimator with application to medical image registration. *Medical Image Analysis*, 2(1):79–98, 1998.

[7] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[9] I. Guskov, S. Klibanov, and B. Bryant. Trackable surfaces. In *Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 251–257, 2003.

[10] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum*, 22(3):263–272, 2003.

[11] V. Schols and M. A. Magnor. Cloth motion from optical flow. In *Vision, Modelling and Visualization*, pages 117–124, 2004.

[12] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. *Computer Graphics Forum*, 24(3):439–448, 2005.

[13] M. Yamamoto, P. Boulanger, J. A. Beraldin, M. Rioux, and J. Domey. Direct estimation of deformable motion parameters from range image sequence. In *Proceedings of*

the IEEE Internatinal Conference on Computer Vision and Pattern Recognition, pages 460–464, 1990.

[14] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[15] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proceedings of the IEEE Internatinal Conference on Computer Vision and Pattern Recognition*, pages 1246–1251, 2005.

[16] A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. In *Proceedings of the Neural Information Processing Systems*, pages 1009–1016. 2007.

[17] F. Wang, B. C. Vemuri, and A. Rangarajan. Groupwise point pattern registration using a novel CDF-based Jensen-Shannon Divergence. In *Proceedings of the IEEE Internatinal Conference on Computer Vision and Pattern Recognition*, pages 1283–1288, 2006.

[18] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proceedings of the IEEE Internatinal Conference on Computer Vision and Pattern Recognition*, pages 822–828, 2005.

[19] M. Prasad, A. Zisserman, and A. W. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proceedings of the IEEE Internatinal Conference on Computer Vision and Pattern Recognition*, pages 1345–1354, 2006.

[20] U. Castellani, V. Gay-Bellile, and A. Bartoli. Joint reconstruction and registration of a deformable planar surface observed by a 3d sensor. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, pages 1138–1139, 2007.

[21] A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image Vision Computing*, 21(13-14):1145–1153, 2003.

[22] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

[23] B. J. Brown and S. Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, pages 822–828, 2004.

[24] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *Proceedings of the European Conference on Computer Vision*, pages 558–569, 2004.

[25] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Transaction on Graphics*, 21(3):355–361, 2002.

[26] U. Castellani, A. Fusiello, and V. Murino. Registration of multiple acoustic range views for underwater scene reconstruction. *Computer Vision and Image Understanding*, 87(3):78–89, 2002.

[27] M. Salzmann, S. Ilic, and P. Fua. Physically valid shape parameterization for monocular 3–D deformable surface tracking. In *British Machine Vision Conference*, pages 1138–1139, 2005.

[28] S. Pissanetzky. *Sparse Matrix Technology*. Academic Press, 1984.

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