USING POINT CORRESPONDENCES WITHOUT PROJECTIVE DEFORMATION FOR MULTI-VIEW STEREO RECONSTRUCTION

Adrien Auclair, Nicole Vincent

Université Paris-Descartes
CRIP 5-SIP, 75006, Paris
adrien.auclair@math-info.univ-paris5.fr
nicole.vincent@math-info.univ-paris5.fr

Laurent D. Cohen

Université Paris-Dauphine, Ceremade, Paris
CNRS, UMR7534, F-75016, Paris
cohen@ceremade.dauphine.fr

ABSTRACT

This paper proposes a novel algorithm to reconstruct a 3D surface from a calibrated set of images. In a first pass, it uses Scale Invariant Features Transform (SIFT) descriptor correspondences to drive the deformation of a mesh toward the true object surface. We introduce a method to handle the fact that these local descriptors are computed at positions that are not projections of mesh vertices in the images. In order to avoid projective deformations due to the large windows of interest of this descriptor, correspondences are only computed between images from the same viewpoint. This is used in a first pass to recover large concavities of the object. In a second pass, a one dimensional Lucas-Kanade tracker is used to recover small scale details. Using publicly available benchmarks, our algorithm obtains high accuracy while being among the fastest ones.

Index Terms— Multi-view Stereovision, Deformable Surface , SIFT, Lucas-Kanade Tracking.

1. INTRODUCTION

The reconstruction of 3D shapes from images has been a very active research field during last years and literature is vast. Most algorithms use the fact that if a 3D position in space belongs to the object surface, pieces of images around projections of this 3D point must be similar. This photo-consistency constraint can be evaluated locally by computing a normalized cross correlation (NCC) or as a simple sum of square differences (SSD) of pixel intensities. For example, in [1], the authors compute this correlation on all the vertices of a 3D grid and then use a deformable model to converge to the surface that minimizes these local correlations (adding a silhouette and a smoothness force). Some other methods work on a voxelisation of the space and use a graph cut approach to find the surface that minimizes an energy based on locally computed correlations ([2], [3]). Most of the existing methods use similar low-level photo-consistency measures but differ in the minimization algorithm. In recent algorithms, features correspondences are used to intialize 3D patches and then a process of expanding and filtering is used to approach the true object leading to excellent results [4].

In this article, we propose to evaluate the photo-consistency with more advanced local image descriptors. We chose to use the SIFT descriptors introduced in [5] as they obtain the best results in the comparative study of [6]. In a first pass, these descriptors are used to drive the deformation of a mesh toward the true object surface. In a second pass, SSD is added to the model to recover small scale details.

In our deformable framework, the current 3D surface of the object is noted S. The n input images are noted $\{I_i\}_{i\in 1..n}$. The projection matrix of the i^{th} camera is noted Π_i. Inversely the projection of a pixel from camera i to its closest point of the surface S is noted Π_i^{-1}. We note $R_{t\to c}$ the result of the projection of S, textured using image I_t, in camera c :

$$R_{t\to c} = \Pi_c \circ \Pi_t^{-1}(I_t) \qquad (1)$$

Local photo-consistency measures are classically computed between views taken from different cameras (i.e., between I_c and I_t with $c \neq t$). This is a problem as a square window in one image does not correspond to a square window in another image. Some algorithms use the hypothesis that the surface is locally planar and thus patches in two different images are related by a homography. This is a valid approximation if windows of interest are not too large. But this may be wrong especially when considering large patches as used by SIFT descriptors. Thus, we introduce a method to search correspondences between descriptors from images issued from the same viewpoints using equation 1. The first image is one of the input image (e.g. image I_c) and the second one is a synthesis image of the current approximation of the surface from the same viewpoint (e.g., one image $R_{t\to c}$).

A theoretical motivation for this method is that for c and t being close cameras, images I_c and $R_{t\to c}$ are identical (except for illuminations changes) if S is the true object surface and if the following condition on the shape of the object is verified :

$$\forall x \in I_c, \left[O_t, \Pi_c^{-1}(x)\right[\cap S = \emptyset \quad (2)$$

If this is verified, as S converges to the true surface object, $R_{t\to c}$ converges to I_c and it will be easier and easier to find correct correspondences. This is an advantage comparatively to algorithms that search correspondences between images from different viewpoints.

This deformation scheme is able to retrieve large concavities of the object. But because the positions of the SIFT descriptors are sparse on the images, small scale details cannot be found. To obtain detailed reconstruction, a one dimensional Lucas-Kanade tracker is then added in a second pass to minimize SSD. The novelty is that we introduce a way to use it in images from the same viewpoint. We demonstrate the accuracy of our approach on the publicly available benchmark of [7].

2. SURFACE EVOLUTION FRAMEWORK

2.1. Initialization

The visual hull of the object is used to initialize the surface as a triangular mesh. Images are first segmented as object and background. Then, the 3D space is represented as a grid of voxels. For each voxel, the number of times its projection is within the object is counted and an isosurface is then extracted using a marching cube algorithm.

2.2. Evolution

A force based approach is chosen to drive the triangular mesh. Each vertex v is attracted by three forces :

$$F(v) = w_p F_p(v) + w_c F_c(v) + w_s F_s(v) \quad (3)$$

F_c is a silhouette force, as defined in [8]. It attracts points detected as contour generators to a position coherent with the silhouette of the object. This requires that the silhouettes must have been precomputed. The distance maps to the silhouettes are computed off-line using a fast marching algorithm. A vertex v is detected as a contour generator in camera c if $\Pi_c(v)$ is close to the boundary of $\Pi_c(S)$. Using the distance map, the closest pixel p_v on silhouette of image c is found. We note X_c the closest point of v on the ray of the camera c passing through p_v. The corresponding force is $F_c(v) = X_c - v$. The force F_s is a smoothness force, computed as a first order Laplacian operator on the mesh ([1]). The force F_p attracts the vertex to a position for being more photo-consistent with the images. The weight w_p, w_c and w_s are used to balance the effects of the three forces.

In the next section, a force F_p, driven by SIFT correspondences is introduced. The goal of this force is to deform the mesh to recover large concavities of the object, starting with the visual hull. In section 4, this force is computed differently to retrieve small scale details.

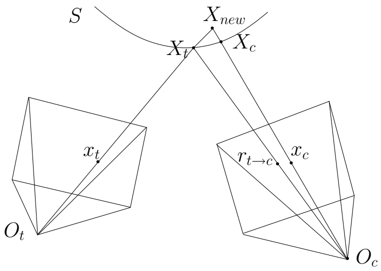

Fig. 1. Scene notations. O_c and O_t are cameras origins.

3. INTEGRATING SIFT CORRESPONDENCES

Considering one camera c, the classical SIFT algorithm described in [5] is used to find points of interest and their descriptors in I_c and in retroprojection images $R_{t\to c}$ where t is a camera close to c (i.e. cameras having similar viewing angle to the scene). The positions of these points are respectively noted $\{x_c^i\}$ and $\{r_{t\to c}^j\}$. When searching for correspondences, we only accept pairs that verify the epipolar constraint. This constraint states that a correspondence between x_c^i in I_c and $r_{t\to c}^j$ in image $R_{t\to c}$ can be accepted only if the distance between $\Pi_t \circ \Pi_c^{-1}(r_{t\to c}^j)$ and the epipolar line of x_c^i in camera t is below a certain threshold. Among the possible correspondences, we select the one that minimizes the distances between the descriptors.

3.1. Computing forces from point correspondences

We note $(x_c, r_{t\to c})$ an accepted and true correspondence for a pair of close cameras (c, t), x_c being a point in image I_c and $r_{t\to c}$ its corresponding position in $R_{t\to c}$. If the surface S was the exact surface object, the vector $(x_c - r_{t\to c})$ would be zero. Thus, we search which surface modification cancels this vector. We note X_c the antecedent of x_c : $X_c = \Pi_c^{-1}(x_c)$. X_t is the antecedent of $r_{t\to c}$: $X_t = \Pi_c^{-1}(r_{t\to c})$. And x_t is the projection of X_t in image I_t (see figure 1). If the surface was correct, we would have :

$$\Pi_c \circ \Pi_t^{-1}(x_t) = x_c \quad (4)$$

This is equivalent to say that the ray starting at x_t in camera t intersects the surface at a 3D point whose projection in camera c is x_c. We note X_{new} the position of this point. It is obtained as the intersection of the 3D rays (O_t, x_t) and (O_c, x_c)).

A major problem is that the point X_c is not a vertex of the mesh and a force that directly attracts this point to X_{new} cannot be defined. Thus, a force is computed for each vertex that is part of the triangle containing X_c. On figure 2, these vertices are p_1, p_2 and p_3, having normals $\vec{n_1}$, $\vec{n_2}$ and $\vec{n_3}$. We note p'_1,p'_2 and p'_3 these points translated along their normals such that the triangle formed by these new points is parallel to

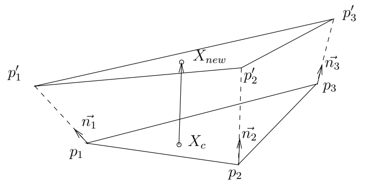

Fig. 2. Computing the force due to one SIFT correspondence.

the triangle (p_1, p_2, p_3) and contains X_{new}. The force applied to the vertex p_1 is defined as : $F(p_1) = p'_1 - p_1$. The same principles applies to p_2 and p_3.

For all couples of close cameras, for all SIFT correspondences, forces computed at three vertices as explained above are computed and stored in a list for each vertex. For a given vertex v, the average of its forces is computed and used as photo-consistency force $F_p(v)$. This first deformation scheme is applied until the maximum motion of a vertex is under a certain threshold. The most important surface evolution observed is that large concavities are recovered. Figures 4.(c) are examples of resulting surfaces after this first pass has been applied.

4. ADDING A MORE LOCAL TERM

Because of the sparsity of the SIFT points, the first pass described in previous section cannot find low scale details. Thus, in a second pass, a photoconsistency force computed per vertex is added to recover these details. This force locally deforms the surface to be more photo-consistent according to a SSD measure.

The algorithm we propose benefits from the fact that we search correspondences between images free of projective deformation. Considering a window centered at the projection x_c of a vertex v in camera c (see figure 3), we want to find the position of the window in the retroprojection $R_{t \to c}$ that minimizes the SSD. The most common tool to achieve this is the two dimensional Lucas-Kanade features tracker [9]. This tool is powerful in many cases but our problem is difficult as both patches may be largely different because of the current surface error. Thus, we propose to use the epipolar geometry to reduce the search to a one dimensional space that is much more robust. This would be simple if the patch similar to a window around x_c was searched along the epipolar line of x_c in camera t (noted $epi_t(x_c)$), using $\Pi_t \circ \Pi_c^{-1}(x_c)$ as starting position. But this would suffer from projective deformation. This is particularly true when using a multi-level implementation of the Lucas-Kanade that works on large window at high scale.

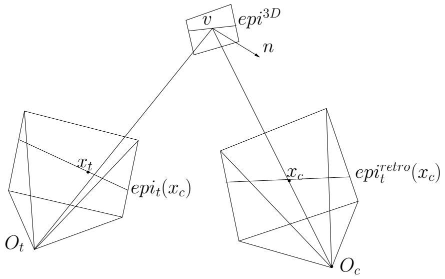

Fig. 3. Retroprojection of the epipolar line.

Instead of tracking along $epi_t(x_c)$, the point is searched in an equivalent direction in image $R_{t \to c}$. To achieve this, the surface S at v is locally approximated by a plane of normal n. The line $epi_t(x_c)$ is projected on this plane (noted epi^{3D} on figure 3) and then retroprojected by Π_c in $R_{t \to c}$. The obtained line is noted $epi_t^{retro}(x_c)$. Using [9], the displacement d that minimizes the SSD along this line of unit director vector dir is solution of :

$$\left(\sum_{W_c} (g.dir)^2 \right) .d = \sum_{W_c} ((I_c - R_{t \to c}).(g.dir)) \qquad (5)$$

where W_c is the window of interest around x_c and g the image intensity gradient of I_c. This system is applied recursively in a Newton-Raphson manner to minimize the SSD between windows of interest in I_c and $R_{t \to c}$. Points whose gradient along the direction dir is too small are not tracked.

Once the projection of a vertex v has been tracked using a couple of images I_c and $R_{t \to c}$, a method similar to the one of 3.1 is used to compute a force to apply to v. The major difference is that instead of computing forces for three vertices of a triangle, the force is only computed for vertex v. Then, it is similarly added to the list of forces and used to compute the average force F_p for each vertex.

5. IMPLEMENTATION

There are a few details that need to be explained for implementation. The first one is the computation of the functions Π_c^{-1}. In each camera, the mesh is rendered encoding triangles labels in RGB channels. For a given pixel x_c in camera c, the label of the triangle containing $\Pi_c^{-1}(x_c)$ is obtained immediately by reading this rendered image at position x_c. Then, the exact position $\Pi_c^{-1}(x_c)$ is computed as the intersection between this triangle and the ray from camera c passing at x_c.

Concerning mesh deformation, we require that the initial visual hull has the correct topology. Then modifying topology is avoided by using simple conditions on edges lengths, vertices displacements and angles between triangles. This does not theoretically prevent from any topological change but it was sufficient in all our experiments. The first pass of the algorithm is applied on large triangles to quickly recover large concavities. Then, the second pass is applied to a mesh having maximum edge length divided by two.

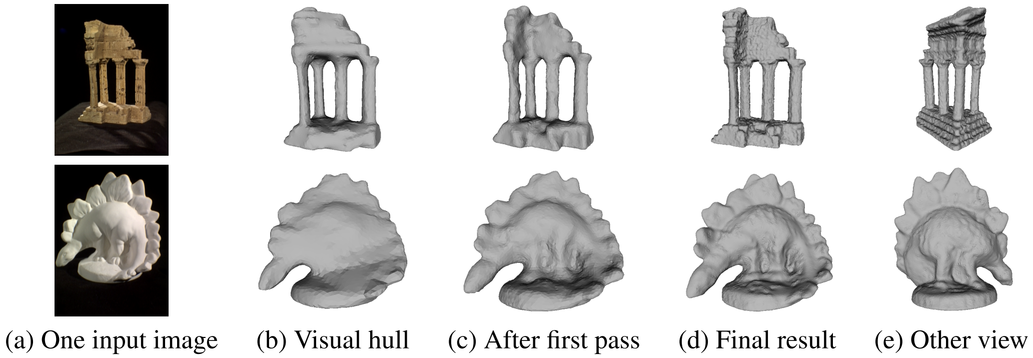

Fig. 4. Results on the evaluation datasets of [7]. (d) and (e) are final results of the algorithm.

6. RESULTS

We tested our algorithm on four datasets given in [7] (evaluations are available at `http://vision.middlebury.edu/mview/eval`). Figure 4 shows intermediate results and the final surfaces. On the sparse dino dataset (16 views), 96.8% of the surface lies within 1.25mm of the ground truth. Only a few methods obtain better completeness. Moreover, our algorithm does not require a long step of exhaustively processing correlations. Thus, it is several times faster than similar algorithms. Still on the sparse dino dataset (16 views), our algorithm runs in less than 30 minutes on a 2.3Ghz Intel Core Duo machine which is among the fastest ones for this level of accuracy.

7. CONCLUSION

In this article, we used a deformable scheme to reconstruct the 3D surface of an object from a set of calibrated images. There are three new aspects in our contribution. First, SIFT correspondences are searched between images virtually issued from the same viewpoints. With this method, one does not have to consider the projective deformations between patches used by the points of interest descriptors. The second point is that forces are computed per triangle and not per vertex. The third point is the use of a one-dimensional Lucas-Kanade tracker to refine the obtained surface. Again, this is achieved between images free of projective deformation. Using publicly available datasets, our algorithm obtains accuracy and completeness that are slightly below the state of the art algorithms but it is several times faster. Another advantage of our algorithm is its simplicity comparatively to other methods with similar level of quality. In future work, we plan to evaluate the precise impact of searching correspondences with images from the same viewpoints.

8. REFERENCES

[1] Carlos Hernandez Esteban and Francis Schmitt, “Silhouette and stereo fusion for 3D object modeling,” *Comput. Vis. Image Underst.*, vol. 96, no. 3, pp. 367–392, 2004.

[2] G. Vogiatzis, P. H. S. Torr, and R. Cipolla, “Multi-view stereo via volumetric graph-cuts,” in *CVPR*. 2005, vol. 2, pp. 391–398, IEEE Computer Society.

[3] Alexander Hornung and Leif Kobbelt, “Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding.,” in *CVPR (1)*, 2006, pp. 503–510.

[4] Yasutaka Furukawa and Jean Ponce, “Accurate, dense, and robust multi-view stereopsis,” in *CVPR*. 2007, IEEE Computer Society.

[5] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” in *International Journal of Computer Vision*, 2004, vol. 20, pp. 91–110.

[6] Krystian Mikolajczyk and Cordelia Schmid, “A performance evaluation of local descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” *CVPR*, vol. 1, pp. 519–528, 2006, evaluation available at http://vision.middlebury.edu/mview/eval.

[8] S. Nobuhara and T. Matsuyama, “Dynamic 3D shape from multi-viewpoint images using deformable mesh model,” in *Processings of 3rd International Symposium on Image and Signal Processing and Analysis, Rome, Italy*, Septembre 2003, pp. pp. 192–197.

[9] Carlo Tomasi and Takeo Kanade, “Detection and tracking of point features,” Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, April 1991.