Tracking and Matching Connected Components from 3D Video

David da Silva Pires, Roberto M. Cesar-Jr.
Universidade de São Paulo - IME - USP
Rua do Matão, 1010, 05508-090, São Paulo-SP, Brazil
davidsp@vision.ime.usp.br, roberto.cesar@vision.ime.usp.br

Marcelo Bernardes Vieira, Luiz Velho
Instituto Nacional de Matemática Pura e Aplicada
Est. Dona Castorina, 110, Jardim Botânico, 22460-320, Rio de Janeiro-RJ, Brazil
mbvieira@impa.br, lvelho@visgraf.impa.br

Abstract

This work presents a method for the detection, tracking and spatial matching of connected components in a 3D video stream. The video image information is combined with 3D sites in order to align pieces of surfaces that are seen in subsequent frames. This is a key step in 3D video analysis for enabling several applications such as compression, geometric integration and scene reconstruction, to name a few. Our approach is to detect salient features in both image and world spaces for further alignment of texture and geometry. We use a projector-camera system to obtain high quality texture and geometry at 30 fps. Successful experimental results corroborating our method are shown.

1 Introduction

The acquisition and manipulation of 3D video information is a key topic of modern research in computer vision and graphics [1, 2]. There are some important open problems in the field that state-of-the-art research has recently started to address. The present paper describes a new solution for one of such problems, namely tracking geometrical connected components (**CCs** for short) by integrating both geometry and texture information provided by the 3D video stream. Data acquisition is performed by a recently proposed system known as 4D Video [3] that provides both dynamic geometry and texture information of the scene in real-time (30 fps).

The underlying idea behind the 4D Video concept is the incremental construction of structured geometry of the scene, i.e. the generation of 3D information accumulated as time flows (4D = 3D + Time). This idea represents one step beyond traditional range video where each frame provides depth information from the camera viewpoint (i.e. range images). The 4D Video concept opens new instigating possibilities that have been barely explored up to now. There is no method that has fully solved this problem, to the best of our knowledge. The main difficulty is how to integrate 3D information from each frame along time. A required intermediary step is tracking and matching 3D object structure along the video stream. This is the task addressed in the present paper. The fact that the objects may be deformable (i.e. non rigid) represents a further difficulty of the whole process.

In order to address this problem, our approach starts by identifying 3D CCs in each frame that represent whole objects or object parts. The next required steps are: (1) to track the connected components; (2) to identify corresponding salient points; (3) to estimate the geometrical transformation for 3D registration of the connected components. We report here our solution for these three steps, which consists in the original contributions of the present paper. The introduced techniques are not only fundamental intermediary steps to implement the 4D Video system, but also allow noise filtering to improve the data quality.

This paper is organized as follows: Section 2 reviews some relevant literature to put the present paper in context. Section 3 presents an overview of the proposed framework while Section 4 reviews the basic concepts behind the 4D Video acquisition system. The CC tracking procedure is explained in Section 5 and CC matching in Section 6. Section 7 shows some experimental results that corroborate our approach. This paper is concluded with some comments on our ongoing work in Section 8.

2 Related Work

In general, 3D acquisition methods in computer vision depend strongly on correspondence and calibration. These methods can be classified based on what type of input data is used and how correspondences are obtained.

The obvious choice for 3D acquisition would be a system based on a pair of cameras and the use of passive stereo methods. However, fully general stereo is an ill-posed inverse problem which is very hard to solve - and real-time requirements make matters even worse. The literature on passive stereo methods is very extensive and for this reason, we will restrict the discussion here to real-time systems. Most of the proposed algorithms cannot perform in real-time without some kind of hardware acceleration. In this context, a recent trend is to take advantage of programmable GPUs [1]. Another option is to use multiple fixed cameras and scene analysis to obtain a background-foreground decomposition of the scene. This is the basis for visual hull and photo hull methods [4, 5].

An alternative to passive stereo algorithms is a system based on camera/projector and active stereo. This option has the advantage of simpler and robust constrained stereo algorithms, but the price is that a pattern of light has to be projected on the scene. Recent work in this area investigates different configurations of cameras and projectors [1, 2]. Some approaches along this line of research use either visible color patterns from a projector [2] or a set of sparse laser sources [6], or even invisible infrared patterns [7].

Our 3D video system [3] is based on a camera/projector pair and active stereo. The hardware is built with off-the-shelf equipment which has many advantages, such as good cost-performance and compatibility. Furthermore, as we will see later, the active stereo color code is simple and effective.

3 Overview of the Proposed Approach

Tracking objects along video sequences is an important computer vision problem that has received much attention. Tracking methods have several applications such as feature detection and extraction, matching, image alignment and stitching for panorama generation.

In the present paper, we are interested in tracking 3D connected components obtained from each frame of a 3D video stream [8]. This is a different problem than tracking 2D objects in traditional gray-level or color video sequences. The output of the tracking procedure is an inter-frame mapping of each connected component.

Figure 1 illustrates our approach. Two subsequent frames are schematically shown (denoted as frame $t-1$ and frame t). Firstly, connected components are detected for each frame. The tracking procedure identifies which connected component in frame $t-1$ corresponds to which in frame t.

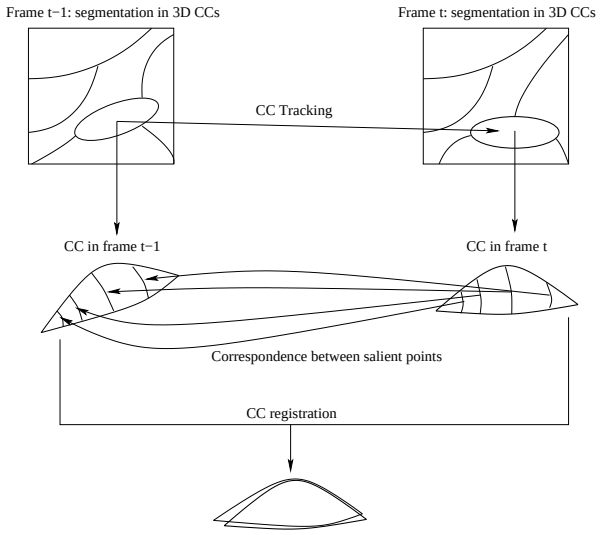


Figure 1. Overview of the proposed scheme for tracking and matching connected components in a 3D video stream.

Once the CCs are tracked along the video sequence, the next step is to match each pair of CC in subsequent frames. In order to match the CCs in an efficient way, three steps are followed: (1) salient points are detected in frame t; (2) the corresponding points are identified in frame $t-1$; (3) an alignment between the corresponding points in frames $t-1$ and t is carried out. These steps are indicated in Figure 1. It is important to note that our 3D video acquisition system provides both texture and geometry information of the scene at each frame, i.e. the data is provided as a Monge surface (range image) with texture. Besides the geometrical information of each 3D CC, our approach takes advantage of the texture image to improve the results in an efficient way. The acquisition rate (30 fps) implies a high temporal consistence that is explored by the tracking and alignment procedures.

Texture resolution is higher than that of geometry because active stereo (structured light) is adopted to obtain 3D geometry information. Based on the Nyquist rate concept, texture resolution must be at least twice that of geometry. In our experiments, it is more than twice. The texture image provides visual features to be explored whereas geometry provides geometrical (shape) features about the objects in the scene. Both are explored in a complementary way.

The 3D video capture system involves 3 different coordinate systems:

- **2D image coordinate system:** it is the parametric do-

main of the texture image, as well as for the range image (i.e. geometry). For each point (u, v) in the image, it is associated a color function $c(u, v) = (r(u, v), g(u, v), b(u, v))$, as well as a depth (range) information $d(u, v)$. As explained in Section 4, the depth information is not calculated by the system for all points (u, v), but it may be obtained from the sample points by interpolation. We use the notation $c(u, v, t)$ and $d(u, v, t)$ to denote the texture and range images of the frame at time t, respectively;

- **Camera coordinate system:** this coordinate system is given by the intrinsic (e.g. focal distance, aspect ratio and center of projection) and extrinsic (translation and rotation) parameters of the camera. Based on the image parameterization (u, v), it is possible to calculate the position of a scene point seen by the camera: lets consider a point P sampled on the range image, associated with pixel $p = (u, v)$ and with depth $w = d(u, v)$. We can calculate the direction vector of P as $\vec{v} = (P - O)/||P - O||$, where O is the origin of the center of projection. All the points belonged to the view ray determined by O and P are of the form $r(t) = O + t\vec{v}$, where the parameter t indicates the distance from origin O on the direction of $\vec{v}$. Then, the coordinate of surface point P is given, in the camera coordinates, by $P = r(w) = O + w\vec{v}$.

- **World coordinate system:** it is the global coordinate system of the scene, where both cameras and scene elements are embedded. Since both the camera and the scene elements may move, the position and orientation are time-dependent.

All these coordinate systems are explored in our approach, each one been chosen according to its suitability to allow the creation of a more efficient solution. CC tracking is carried out in the image coordinate system. Visual feature detection and texture alignment for the identification of salient points and their correspondents in the across different frames are carried out in the image coordinate system and allow mapping the information between the camera coordinates of different frames. Finally, shape feature detection and geometry alignment are carried out in the camera coordinate system and allow mapping position information on the global coordinate system.

4 Real-time 3D video acquisition

To detect geometrically connected components in a scene, the 3D capture system should provide high quality images and geometry in real-time. Quality is crucial for achieving precise analysis and synthesis. Real-time is required to exploit time coherence and capture subtle connected component motion and to reduce matching problems during spatiotemporal analysis.

The system used for obtaining 3D data is based on a camera/projector pair and active stereo [3]. It is built with off-the-shelf NTSC video equipment. The key of this system is the combination of the color code (b,s)-BCSL [9] with a synchronized video stream.

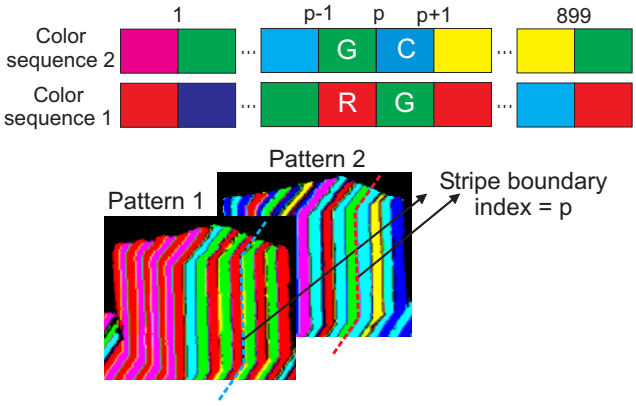


Figure 2. Decoding stripe transitions.

The (b,s)-BCSL code provides an efficient camera/projector correspondence scheme. Parameter *b* is the number of colors and *s* is the number of patterns to be projected. Two patterns is the minimum, giving the best time coherence compromise. The use of complementary patterns is required to robustly detect stripe transitions and colors. Our system applies six colors that can be unambiguously detected through zero-crossings: RGBCMY. In our experiments, we use a (6,2)-BCSL code that features two patterns of more than 900 stripes.

To build camera/projector correspondence, we project a subsequence of these two patterns onto the scene and detect the projected stripe colors and boundaries from the image obtained by a high-speed camera. The four projected colors, two for each pattern, detected close to any boundary, are uniquely decoded to the projected stripe index *p* (Figure 2). The correspondent column in the projector space is detected in O(1) by using (6,2)-BCSL decoding process. The depth is then computed by the camera/projector intrinsic parameters and the rigid transform between their reference systems.

We project every color stripe followed by its complementary color to facilitate the detection of stripe boundaries from the difference of the two resulting images robustly. The stripe boundaries become zero-crossings in the consecutive images and can be robustly detected with subpixel precision. One complete geometry reconstruction is obtained after the projection of the pattern 1 and its complement followed by pattern 2 and its complement.

The (6,2)-BCSL can be easily combined with video

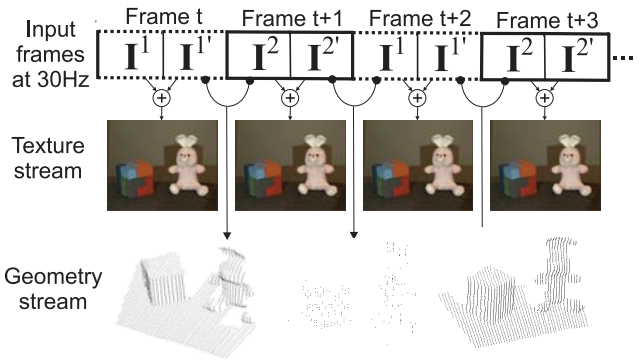


Figure 3. Input video frames, and the texture and geometry output streams with 30 fps rate.

streams. Each 640x480 video frame in the NTSC standard is composed of two interlaced 640x240 fields. Each field is exposed/captured in 1/60 of a second. The camera and projector are synchronized using genlock. For projection, we generate a frame stream interleaving the two patterns that are coded with their corresponding complements as fields in a single frame. This video signal is sent to the projector and connected to the camera's genlock input. The sum of the two fields gives a texture image and their difference provides projected stripe colors and boundaries. The complete geometry and texture acquisition is illustrated in Figure 3.

This system is suitable for tracking components because it maintains good balance between texture, geometry and motion detection. Our videos were obtained by projecting 70-90 stripes over scenes with different scales. We have used a Sony HyperHAD camera and an Infocus LP-70 projector.

Further information and video examples can be downloaded, together with a special purpose viewer, from `http://www.impa.br/~mbvieira/video4d`.

5 Identifying and Tracking the Connected Components

5.1 Connected components detection

The first step for connected components detection is to construct a 3D point connection graph. In a simple scheme, points closer than a threshold l are said to be *connected*. A *component* is formed by isolated trees of connected points. Naive approaches for constructing such graph can have a high time complexity.

The decoding methods used in active stereo systems automatically provide local information that can be used for detecting connected components. The *j* (line in camera space) and *p* (projected plane number) coordinates of a 3D point define an intrinsic topology.

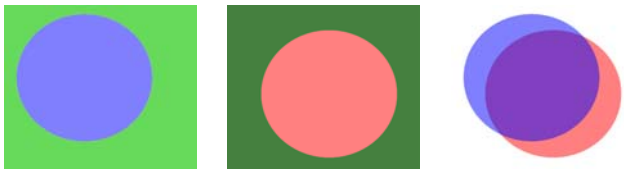
Indeed, points which are near in (j, p) discrete space are likely to be near in the 3D space. This is not always true because of depth discontinuities in the scene. On the other hand, distant points in (j, p) space are also distant in 3D space. Thus, using a (j, p) table reduces significantly the search space for computing the connection graph and the distinct components.

In this scheme, two points are connected if they are k-neighbors in the (j, p) table and their 3D space distance is smaller than a threshold l. One must be careful because j resolution is typically much greater than p resolution and, consequently, their unities are different.

5.2 Connected components tracking

Because the CCs are independently segmented frame by frame, it is necessary to implement a tracking procedure to identify which CC in frame $t-1$ corresponds to which in frame t. Different important events must be held by the tracking procedure, namely: moving CCs that change the shape; new CCs that appear in the scene; old CCs that disappear (e.g. by moving out of the imaged scene); CCs that merge (e.g. two different objects that touch each other in some instant); CCs that split (e.g. touching objects that move apart each other).

The proposed tracking procedure considers consecutive frames so that we may explore the fact that each CC undergoes small movements between subsequent frames because of the high acquisition rate (30 fps). In such cases, the intersection of a CC c on frame t with its corresponding CC on frame $t-1$ is expected to be large. The intersection between the CCs in subsequent frames is used to track the CCs (see Figure 4).



(a) (b) (c)

Figure 4. (a) and (b) show two corresponding connected components in subsequent frames; (c) Because of the high acquisition rate, the two CCs tend to have a large superposition area, as illustrated in (c).

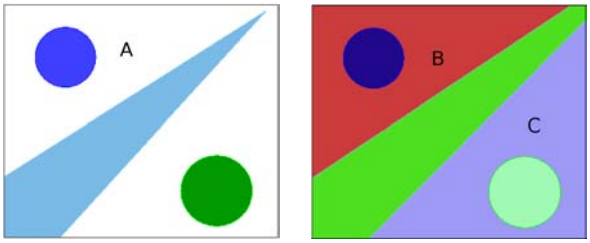
Let $L = \{l_1, l_2, ..., l_{|L|}\}$ be a set of labels of the CCs in each frame. Tracking a connected component can be seen as the determination of a mapping $M : L \rightarrow L$ such that, for each frame at time *t*, if a CC has label l_x on frame at time

$t-1$, and label l_y in the frame at time t, then $M(l_y) = l_x$. The range image $d(u,v,t)$ is used to build M. For each CC c at time t, the intersection between c and all CCs at time $t-1$ is calculated. Only those connected components at time $t-1$ with an intersection larger than a priori defined threshold are considered as candidates. Let $c_1, c_2, ..., c_n$ be these candidate CCs at frame $t-1$. The mean difference between the range coordinates of c and $c_1, c_2, ..., c_n$ is calculated and the CC in $t-1$ that minimizes this difference is chosen as the final candidate:

$$m(c, c_i) = \frac{1}{|c \cap c_i|} \sum_{(u,v) \in c \cap c_i} |d(u,v,t) - d(u,v,t-1)|$$

$i = 1, 2, ..., n$. A maximum distance threshold d_{max} is set so that, if $\min_i m(c, c_i) > d_{max}$, then no mapping between frames $t-1$ and t is created for c. In this case, c is considered to be a new CC that appeared in frame t, i.e. it was not present in frame $t-1$. A new label is then assigned to c.

It is important to note that this algorithm also correctly deals with the case where a single connected component is split from one frame to the next: the two newly created CCs in frame t are expected to be mapped onto the same CC in frame $t-1$. An example is shown in Figure 5.2.



(a) Frame $t-1$ (b) Frame t

Figure 5. (a) and (b) illustrate two subsequent frames containing three objects and the background, which are taken as 4 CCs. The background CC is indicated as A. As the middle CC moves and reaches the image edge, the background is divided into two CCs, as shown in (c). The CC tracking procedure is able to deal with such situation by identifying that both CCs B and C in frame t correspond to the CC A in frame $t-1$.

6 Matching the Connected Components using Texture and Geometry Information

Once the CCs are tracked, each pair of corresponding CCs in subsequent frames should be aligned. This is done by applying the iterative closest point (ICP) algorithm [10] to a set of selected salient points of the CCs. The ICP algorithm is widely used for aligning three-dimensional models based purely on the geometry, and sometimes color, of the meshes. Three steps are followed in order to select the points that feed the ICP algorithm:

- **Texture alignment:** the texture portions corresponding to the considered CCs are extracted and matched through correlation. The maxima point of the correlation of the two texture portions $c(u,v,t) \circ c(u,v,t-1)$ indicates the translation that one portion should undergo in order to match the other. A pointwise correspondence between the two portions of $c(u,v,t)$ and $c(u,v,t-1)$ is then established. The main advantage of using texture to create this correspondence between the two frames, instead of geometry, is the higher resolution presented by the former;

- **Identification of salient points in the geometry data (frame t):** The geometry data $d(u,v)$ represents range information measured along the light patterns projected onto the scene. The set of local maxima and minima points along each light stripe in frame t are taken as the salient points. These points are identified by numerical differentiation of $d(u,v,t)$ along each light stripe;

- **Identification of corresponding salient points in the geometry data (frame $t-1$):** The position in the texture $c(u,v,t)$ of each salient point of $d(u,v,t)$ is identified. Because of the above texture correlation alignment procedure, this is also the position of the salient point at the aligned texture image $c(u,v,t-1)$. It is important to note that, because of the texture image higher resolution, this salient point position in frame $t-1$ may not have a corresponding sample point at the geometry data of frame $t-1$. Therefore, the salient point at frame $t-1$ is obtained by interpolation of the geometry points.

The above procedure leads to two sets of corresponding salient points of the geometry data of frames t and $t-1$. These two sets of points are then registered using the well-known ICP algorithm [10], thus producing the desired result.

7 Experimental Results

In this section, we present experimental results using 3D video sequences. Figure 6 shows the texture information of 3 frames of a video sequence (left column) together with the corresponding segmented geometry data (right column). In the first frame (top row) there is a person in front of a flat

background, thus defining two connected components in the geometry space. The CCs are coded as different colors. As the person moves, the corresponding CCs are tracked as expected. The background structure that is behind the person in the first frame appears as a third CC, as shown in the second row. It is important to note that this third CC is not identified as the first background CC because it was not initially seen as a CC in the geometry data (first row) and because the person CC completely divides the background from bottom to top. This third CC that appears is also correctly tracked, as shown in the third row.

An example of the texture information and the corresponding segmentation using the geometry CC is shown in Figure 7. This segmented texture is used by the texture alignment procedure to create a mapping between the textures of subsequent frames, as shown in Figure 8. The salient points calculated for a given frame is shown in Figure 9(a), together with its corresponding interpolated salient points in the previous frame (b). The resulting matched salient points using ICP (the Scanalyze software has been used in our experiments - `http://graphics.stanford.edu/software/scanalyze/`) are shown as the image (e) of Figure 9, thus corroborating the introduced method.

The identification and the tracking of connected components procedures were developed with real-time constraints in mind, so they execute at video rate time. The texture alignment was written in a different language and not integrated with the remainder code yet. Thus, the entire process is almost fully automatic, needing just a few user interactions.

8 Conclusion

The presented method for tracking and matching connected components has shown to be effective in our experiments, being suitable to integrate the 4D Video system. The method correctly deals with different events that may occur during the video flow, such as when CCs appear, disappear or split. Our ongoing work focus on improving the implementation of the ICP to allow real-time CC registration. Kalman filters [11] will be applied in order to speed up the whole 4D Video process by combining the different available data sources to predict CCs position and pose as new frames are acquired. Although the adopted ICP software lead to satisfactory results, we are working on improving the ICP implementation to improve the match. Finally, a procedure for shape merging using the CC registration information is currently under development and will be reported in due time.

Acknowledgments

R. Cesar-Jr. is grateful to FAPESP (99/12765-2) and to CNPq (300722/98-2 and 474596/2004-4). D. S. Pires benefits of a CAPES scholarship and is grateful to Yossi Zana for fruitful discussions. L. Velho and M. B. Vieira are members of the VISGRAF Laboratory at IMPA, which is sponsored by CNPq, FAPERJ, FINEP, and IBM Brasil. M. B. Vieira is supported by a Pro-Doc grant from CAPES. The research reported in this paper was developed in the context of the CT-Info project financed by FINEP and the GIGA project from RNP.

References

[1] P. S. Huang, C. Zhang, and F.-P. Chiang. High-speed 3-d shape measurement based on digital fringe projection. *Optical Engineering*, 42(1):163–168, 2003.

[2] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics*, 23(3):548–558, 2004.

[3] M. B. Vieira, A. Sa, L. Velho, and P. C. Carvalho. A camera-projector system for real-time 3d video. In *IEEE International Workshop on Projector-Camera Systems (PROCAMS)*, 2005.

[4] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. Vande Moere, and O. Staadt. Blue-c: A spatially immersive display and 3d video portal for telepresence. *ACM Transactions on Graphics*, 22(3):819–827, 2003.

[5] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In *Proceedings of ACM SIGGRAPH 2000*, pages 369–374, 2000.

[6] V. Popescu, E. Sacks, and G. Bahmutov. Interactive point-based modeling from dense color and sparse depth. In *SPBG'04 Symposium on Point-Based Graphics*, 2004.

[7] C. Frueh and A. Zakhor. Capturing $2\frac{1}{2}$d depth and texture of time-varying scenes using structured infrared light. In *3DIM*, 2005.

[8] E. B. Meier and F. Ade. Object detection and tracking in range image sequences by separation of image features. In *IEEE International Conference on Intelligent Vehicles*, pages 176–181, 1998.

[9] A. Sa, P. C. Carvalho, and L. Velho. (b, s)-bcsl: Structured light color boundary coding for 3d photography. In *Proceedings of 7th International Fall Workshop on Vision, Modeling, and Visualization*, 2002.

[10] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[11] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *Proc. British Machine Vision Conference*, volume I, pages 63–72, Manchester, UK, September 2001.

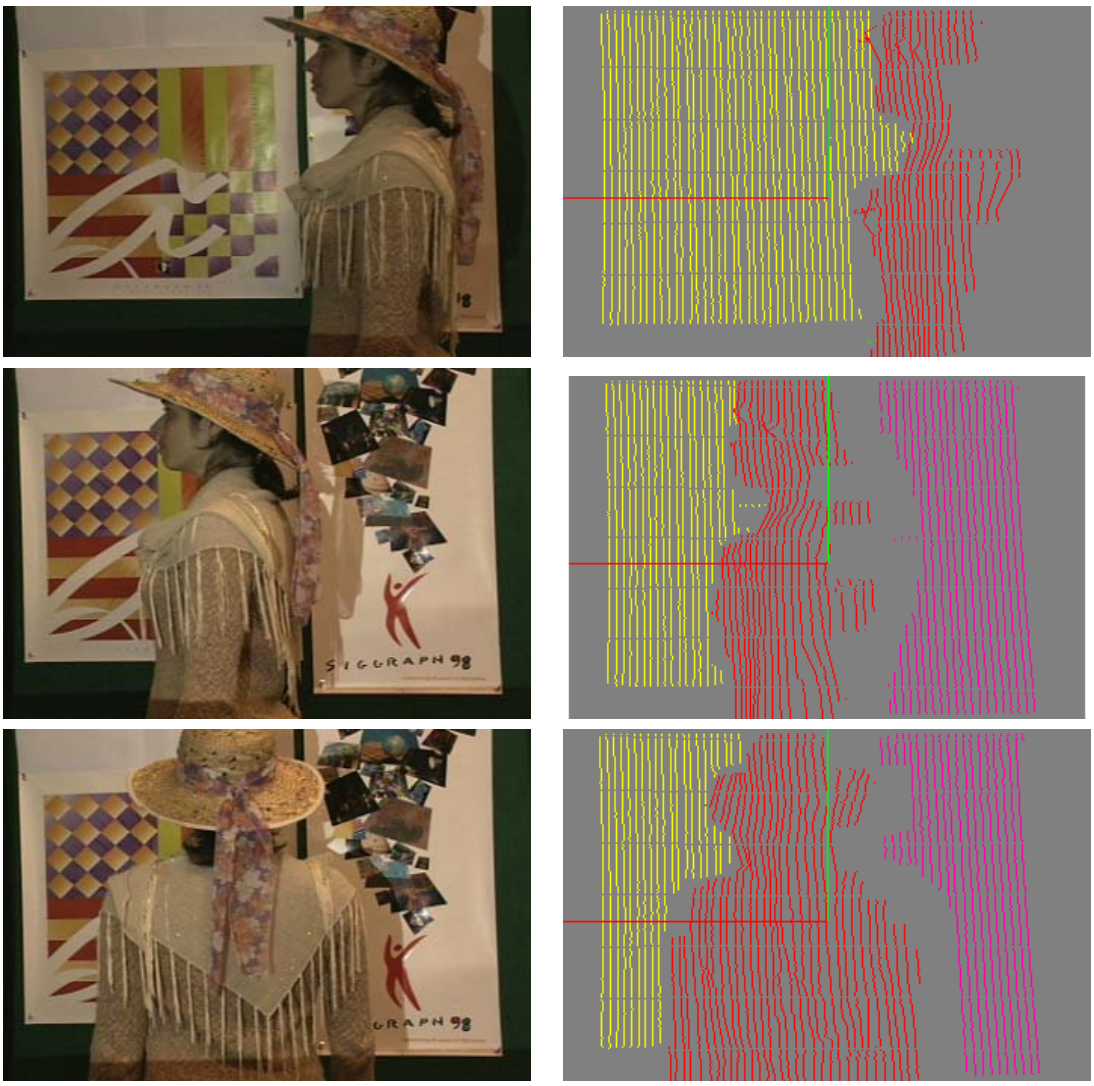
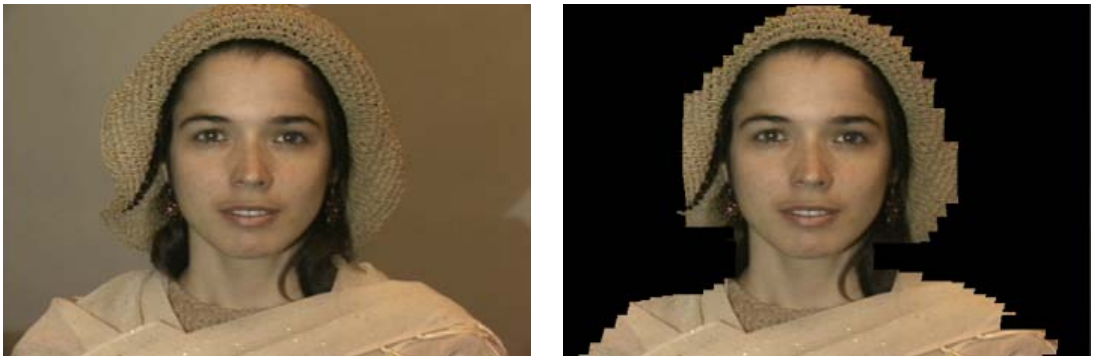


Figure 6. Example of some frames of video with its corresponding connected components being tracked. Texture information is shown in the left column while geometry (tracked CCs) are shown in the right column. These are not subsequent frames in the sequence.



(a) (b)

Figure 7. (a) Complete texture of a frame. (b) Texture of just one connected component, used to calculate the texture matching.

(a) (b) (c)

Figure 8. (a) Texture of a connected component in a frame $t-10$. (b) Texture of the same connected component in the frame t. (c) Frame t translated in $(u, v) = (1, 71)$ relative to frame $t-10$ and superimposed to this last. Frame t is shown with 50% of transparency. We used a difference of 10 frames just for visualization purposes. On consecutive frames, the translation is usually small.

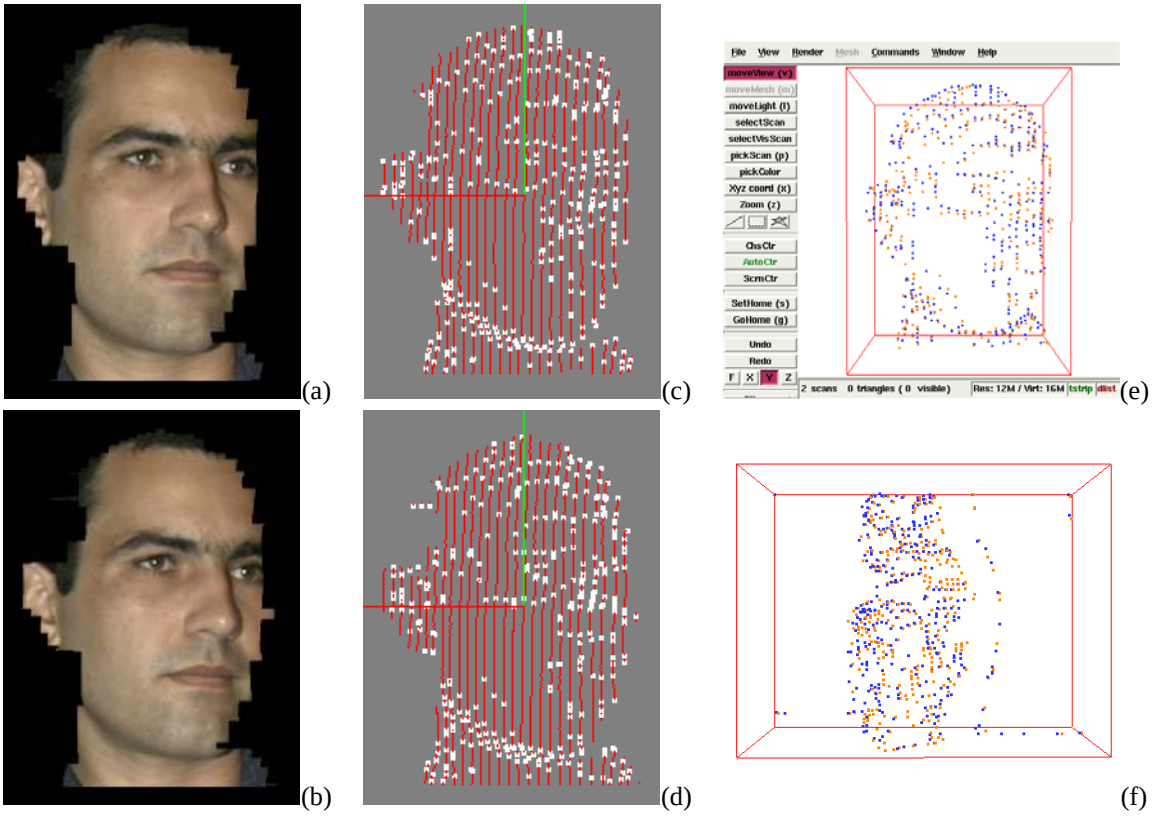


Figure 9. Two consecutive frames containing just one connected component are shown in the left column. Each frame is accompanied with its corresponding range data in the right column. The salient points (local maxima and local minima) are the white ones. The image (e) shows the resulting matched points using ICP. The image (f) shows another set of aligned points regarding Figure 8.