ACCURATE DEPTH-MAP ESTIMATION FOR 3D FACE MODELING

Giovanni Dainese, Marco Marcon, Augusto Sarti, Stefano Tubaro

Dipartimento di Elettronica e Informazione - Politecnico di Milano
Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
phone: +39-0223999639, fax: +39-0223999611
email: dainese/marcon/sarti/tubaro@elet.polimi.it

ABSTRACT

In the past few years several systems for tridimensional face reconstruction based on the analysis of 2D images have been proposed. The main goal of these systems is to provide fast and reliable 3D information for face recognition systems. Nonetheless, affordable image-based systems that are able to guarantee a high level of details and fast processing using commercial devices are far from being available. In this paper we propose a novel approach for a fast generation of detailed depth-maps of human faces based on a set of three calibrated cameras. The proposed algorithm is based on an fast implementation of the graph-cuts [1, 3] approach, which guarantees high-quality results in just a few seconds of processing time.

1. INTRODUCTION

Reconstructing a detailed depth-map from a set of cameras is a classical and long-debated problem of computer vision. In the past few years this problem has raised a great deal of interest due to the increasing number of applications, both in vision and in graphics, where this problem has become of crucial importance. Classical depth-map estimation algorithms based on multi-camera acquisition are, in fact, time consuming, particularly when the accuracy is an issue. In this paper we show how a global energy approach derived from graph-cuts algorithms can be used to speed up the image-based modeling process while guaranteeing an accurate reconstruction. The final result is a detailed 3D texture-mapped mesh of the acquired face, which can be used for recognition purposes or for the alignment and the normalization of facial features.

Although the algorithm described in this paper was developed specifically for 3D face recognition purposes, its range of application is much wider than that, as it can be used whenever a fast and detailed depth-map from multiple calibrated images is needed.

2. DEPTH MAP RECONSTRUCTION

In this section we show how to accurately reconstruct the depth-map of a face from a set of images. In order to do so, we started from the well-known *graph cuts* approach [1, 3, 4, 9], and we adapted it and optimized it to the problem of depth-map reconstruction.

In what follows we provide a brief description of the energy minimization approach that the graph cuts method is based on. After then we will show how to formulate the problem of depth map reconstruction in term of energy minimization.

2.1 Energy minimization approach

It is well known that the problems of depth map reconstruction and image restoration can be elegantly approached in terms of energy minimization [3, 4], with extremely appealing results. In the past few years powerful energy minimization algorithms have been developed based on graph cuts [3, 5, 24]. These methods are fast enough to be of practical interest, but unlike other methods such as simulated annealing, the solutions based on graph cuts cannot be applied to arbitrary functions. In this paper we will use some recent results [4] on graph construction, in order to extend the method to quite a general class of energy functions.

The energy minimization formalism exhibits several advantages. It allows a detailed description of the problem to be solved. Moreover, energy minimization naturally enables the use of soft constraints, such as spatial coherence and a global smoothness term. This allows us to avoid ambiguities with spatially smooth answers that preserves discontinuities.

2.2 Problem formulation

Let us assume that n calibrated images of the same scene are taken from different viewpoints (or at different times). Let us choose a reference camera and let $\mathscr{P}$ be the set of pixels of the corresponding image. A pixel $p \in \mathscr{P}$ corresponds to a ray in 3D-space. Consider the first intersection of this ray with an object in the scene. Our goal is to find the depth of this point for all the pixel of the preferred image. We thus want to find a labeling $f : \mathscr{P} \rightarrow \mathscr{L}$ where $\mathscr{L}$ is a discrete set of labels corresponding to increasing depths from the preferred camera. Equivalently, we want to obtain the *depth map* of the pixels in the preferred image.

A pair $\langle p, l \rangle$ where $p \in \mathscr{P}$, $l \in \mathscr{L}$ corresponds to some point in 3D-space. We will refer to such points as *3D-points*.

We define our energy function as consisting of two terms:

$$E(f) = E_{data}(f) + E_{smooth}(f)$$

In their work, Kolmogorov and Zabih [1] formulate the problem of scene reconstruction in a slightly different fashion, which allowed them to obtain a depth map for every image in the input set by an energy minimization approach. This leads to a computational expensive algorithm whose result is a unorganized clouds of point representing the surface of the visible part of the scene to reconstruct. Moveover, in order to achieve an effective reconstruction from the input set, a further energy term (called *visibility term*) must be accounted for, in order to avoid mutual intersections of re-projected rays coming from different cameras (see [1] for more details). Whereas with our definition we can treat a

very large number of camera configurations without these further limitations.

Notice also that in our approach it is no long necessary to define the visibility term like in [1]. In fact, assuming that the set of label corresponds to the increasing depths from the preferred camera, there cannot exists occluding pixels in the same image. As a consequence, a visibility term is no longer necessary. The other terms are also quite different. Our data term, for example, is defined as follow:

$$E_{data}(f) = \sum_{p \in \mathscr{P}} D(p)$$

where $D(p)$ is a non-positive value which results from the differences in intensity between corresponding pixels. $D(p)$ is computed for every pixel of the preferred image (we indicate this image with the index j) by this steps:

1. from p, we get the corresponding 3D-point by back-projecting it from the reference camera center of projection with the selected depth and then we project this 3D-point on each other calibrated image obtaining a set of $n-1$ corresponding pixels $\{q_1, q_2, \ldots, q_i, \ldots, q_n | i \neq j\}$;
2. on every non-reference image we compute the SSD (Sum of Square Difference) using a square window centered on q_i and the one centered on p, obtaining the set of values $\{d_1, d_2, \ldots, d_i, \ldots, d_n | i \neq j\}$;
3. finally, we have

$$D(p) = min(0, \sum_{\substack{i=1 \\ i \neq j}}^{n} d_i - K) \quad (1)$$

where K is a positive constant that is large enough to capture significant variations of the SSD function (a typical value is $K = 30$).

The smoothness term is quite similar to the one used in [1] and its goal is to encourage neighboring pixels in the preferred image to have similar depths. The smoothness term is defined as follow:

$$E_{smooth}(f) = \sum_{\{p,q\} \in \mathscr{N}} V_{\{p,q\}}(f(p), f(q)) \quad (2)$$

This term involves the notion of neighborhood: we assume that there is a neighborhood system on pixel

$$\mathscr{N} \subset \{\{p,q\} \mid p,q \in \mathscr{P}\}$$

This can be the usual 4-neighborhood system: pixels $p = (p_x, p_y)$ and $q = (q_x, q_y)$ are neighbors if they are in the same image and $|p_x - q_x| + |p_y - q_y| = 1$.

In [1], the function $V_{\{p,q\}}$ takes on the following form:

$$V_{\{p,q\}}(l_p, l_q) = \begin{cases} U_{\{p,q\}} & \text{if } l_p \neq l_q \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where the $U_{\{p,q\}}$ is the following non-decreasing function:

$$U_{\{p,q\}} = \begin{cases} 3\lambda & \text{if } \Delta I(p,q) < 5 \\ \lambda & \text{otherwise} \end{cases} \quad (4)$$

In order to obtain a smooth reconstruction that preserves discontinuities, we chose to follow a particular strategy in the use of the smoothness term. In fact, it is well-known that graph cuts techniques often yields flat and blocky results. This may not be important for disparity maps, but it is crucial for shape reconstruction. In order to avoid this problem, we make a first cycle of the reconstruction algorithm with a limited set of labels, in order to rapidly reach a value of the energy that is close to the local minimum that could be reached at convergence with the original algorithm. This corresponds to a good approximation of the position of the 3D-points, which can be improved with a second cycle at twice the resolution, where we change the function $V_{\{p,q\}}$ defined in (3) with this new function:

$$\hat{V}_{\{p,q\}}(l_p, l_q) = \begin{cases} U_{\{p,q\}} & \text{if } |l_p - l_q| > z\_threshold \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In fact, this function relaxes the penalty mechanism of the smoothness term, giving a 0 penalty not only to the neighboring pixels that lie at the same depth but also to the ones that stay sufficiently close to each other. The idea is supported by the fact that after the first cycle of the algorithm, only some of the pixels are approximatively well positioned in 3D-space by the consistency measure given by the data term, while the other pixels' locations are only decided by the smoothness term. This term, in fact, forces them to lie at the same level of the neighboring pixel, resulting in flat blocks. Thus, relaxing the constraint imposed by the first smoothness term, neighboring pixels have greater chance to occupy adjacent depths correctly.

2.3 Graph cuts Algorithm

Thanks to our energy redefinition the results obtained from the standard graph cuts algorithm (as defined in [1]) are much more accurate. As shown in the next paragraph, further depth map optimization guarantees high fidelity in the reconstructed data.

2.4 Depth map optimization

Even though the graph cuts algorithm is able to reconstruct an accurate depth map, it works only with a limited set of depths and, therefore, it introduces a considerable quantization error in the positioning of each one of the 3D points. in order to overcome this problem, it is necessary to adopt an optimization step which produces more regular depth maps. The output of this process is a new depth map, where the discontinuities are preserved while the other parts turn out to be smoother.

In order to do so, we consider the depth map as a functions of two variables defined on the preferred image and we apply a series of 2D filters to it. In particular, we start with a median filter to eliminate possible outliers and then we apply a dithering technique: some white noise is added to the depth function and, then, a low pass filter is used to reduce depth quantization error and obtain a smoother map. In order to preserve discontinuities, the 2D low-pass filter keeps the information needed from the neighbors of a pixel only if the depth distance is below a certain threshold. The size of the filter windows and this threshold are empirically chosen on the basis of the current reconstruction.

2.5 Mesh triangulation

From the previous section we learnt how to compute a depth map from a set of images of the interested object. We also said that every map can be seen as a 2D function defined on the preferred image. Starting from this point, we can easily implement a triangulation algorithm that produce a mesh from a depth map on the basis of the neighboring pixels. Consider four neighboring points and the six possible connection shown in figure 1:

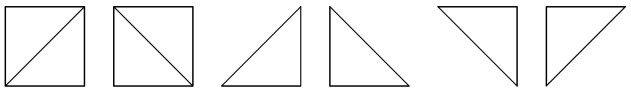

Figure 1: Six possible configurations for the creation of triangles from four neighboring points.

when two neighboring pixels have depths differing by more than some threshold, there is a step discontinuity. The threshold is determined directly by the human operator, as the maximum depth difference which has to be considered a surface discontinuity. If a discontinuity is present, a triangle should not be created. Therefore, for four neighboring pixels, we only consider 3D-points that are not along discontinuities. If three of them satisfy this condition, a triangle will be created in one of the last four style in figure 1. If none of the four are along a discontinuity, two triangles will be created, and the common edge will be the one with the shortest 3D distance, as shown in the first two styles in figure 1.

Figure 2: The acquisition system. Three cameras placed around a gate.

Repeating these steps for every mesh will lead to a volumetric function whose zero leveset locates the object surface. The resulting object can be seen as a sort of convex hull obtained by linking together the meshes and taking only the part of the 3D space contained in their intersection.

3. EXPERIMENTAL RESULTS AND SYSTEM DESCRIPTION

The proposed algorithm has been applied to a variety of test images (of faces) acquired with a calibrated trinocular camera system. The acquisition system is based on three synchronized cameras Powershot G3 from Canon as shown in figure 2. The acquired images are in 2272×1704 JPEG format and, after face segmentation, the area that is actually useful for 3D reconstruction uses about 1MPixel of the 4 that are available.

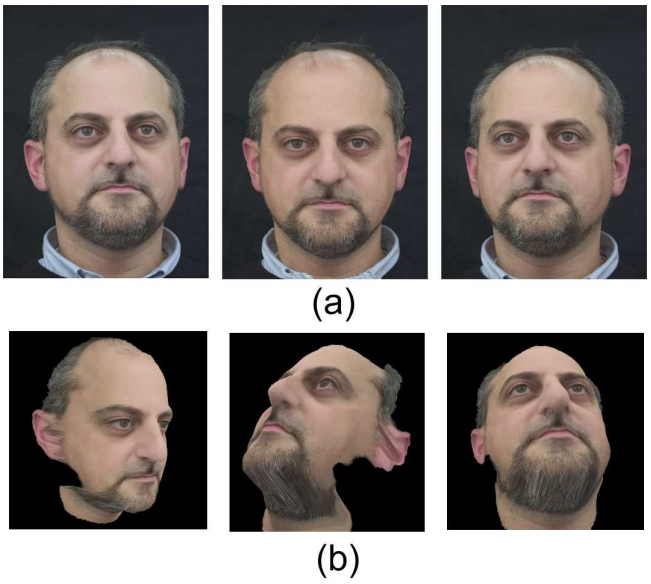

Figure 3: A depth-map from a face: a) the acquired images. b) three different views from 'virtual observers' of the depth-map

In figure 3 we show the three segmented facial images and the final 3D model. The reconstruction time is about 3 seconds on a Pentium IV 3GHz. In particular, using the described approach we can obtain a very good depth estimation of difficult facial zones were uniform skin color and highly non-lambertian reflectance generate ambiguity in depth estimation. The parameters K of equation (1) and λ of equation (4) are determined heuristically from a set of facial images. Anyway, we observed that the estimated values gives good values for every facial image analyzed. The parameters can be varied to gain some insight about the algorithm: for big values of λ the smoothness dominates the correlation, resulting in a map with many flat blocks of pixels, whereas little values of λ yields to an irregular depth map with many wrong discontinuities. In our experiment, we chose the values $K = 30$ and $\lambda = 5$.

4. CONCLUSIONS

Fast 3D depth-map extimation from multiple calibrated images is a critical process. In order to perform this task we presented a reconstruction algorithm based on graph cuts theory. We defined an energy function whose minimum represents the solution to our problem and we implemented a technique to improve the obtained depth maps. The parameters were optimized for 3D face reconstruction. Anyway we also obtained good results with completely different categories of 3D objects. In figure 4 we give an example of a scene containing a dinosaur above a ship: in the second raw is possible to view the good results for a virtual viewer placed in different positions around the depth-map.

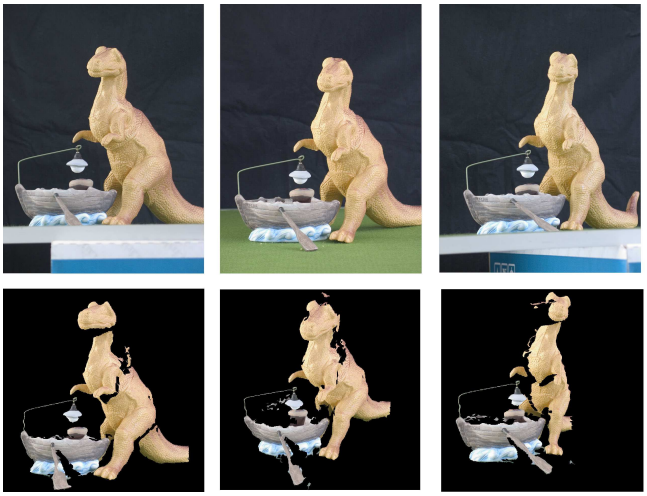

Figure 4: A depth-map from a scene: In the first raw the three acquired images; in the second raw there are three different views from 'virtual observers' of the depth-map

One advantage of this approach is indeed its comptational efficiency. In fact, we obtain a depth-map of the analyzed scene using images of about one megapixel in less than 3 seconds using normal hardware (3GHz Pentium4 processor with 1GB of RAM).

REFERENCES

[1] V. Kolmogorov and R. Zabih. "Multi-camera Scene Reconstruction via Graph Cuts". *In European Conference on Computer Vision*, 2002.

[2] V. Kolmogorov, R. Zabih and Steven Gortler. "Generalized Multi-camera Scene Reconstruction Using Graph Cuts". *In European Conference on Computer Vision*, 2003

[3] V. Kolmogorov and R. Zabih. "Computing Visual Corrispondence with Occlusion via Graph Cuts". *In International Conference on Computer Vision*, 2001.

[4] V. Kolmogorov and R. Zabih. "What energy functions can be minimized via graph cuts?" *In European Conference on Computer Vision*, 2002.

[5] Y. Boykov, O. Veksler and R. Zabih. "Fast Approximate Energy Minimization via Graph Cuts". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2001.

[6] S. M. Seitz and C. R. Dyer. "Photorealistic scene reconstruction by voxel coloring". *International Journal of Computer Vision*, 1999.

[7] S. Birchfield and C. Tomasi. "A pixel dissimilarity measure that is insensitive to image sampling". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[8] S. Paris, F. X. Sillion and L. Quan. "A Surface Reconstruction Method Using Global Graph Cut Optimization". *Asian Conference of Computer Vision*, January 2004.

[9] D. Snow, P. Viola and R. Zabih. "Exact Voxel Occupancy with Graph Cuts". *In Proc. Computer Vision and Pattern Recognition Conf.*, 2000.

[10] W. E. Lorensen and H. E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm". *In Proc. SIGGRAPH*, 1987.

[11] A. Laurentini. "The visual hull concept for silhouette-based image understanding". *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1994.

[12] K. N. Kutulakos and S. M. Seitz. "A theory of shape by space carving". *Int. J. of Computer Vision*, 2000.

[13] A. Hilton, "On Reliable Surface Reconstruction from Multiple Range Images", *Department of Eletronic & Electrical Engeneering, University of Surrey, Tecnical Report VSSP-TR-5/95*, October 1995.

[14] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Surface reconstruction from unorganized points", *Computer Graphics, vol.26(2)*, 1998.

[15] M. Soucy and D. Laurendeau, "Multi-resolution surface modeling from multiple range images",*Proceedings of IEEE Conference on Computer Vision and Pattern Recongnition"*, 1992.

[16] M. Soucy and D. Laurendeau, "A dynamic integration algorithm to model surfaces from multiple range images",*Machine Vision and Applications, vol.8"*, 1995.

[17] G. Turk and M. levoy, "Zippered polygon meshes from range images", *Computer Graphics Proceedings, SIGGRAPH*, 1994.

[18] M. Rutishauer, M.Stricker and M. Trobina, "Merging range images of arbitrarily shaped objects", *Proceedings of IEEE Conference on Computer Vision and Pattern Recongnition*, 1992.

[19] J. D. Boissonnat, "Geometric structures for three-dimensional shape representation", *ACM Transactions on Graphics, vol.3*, 1984.

[20] R. Szelisky and R.Zabih, "An experimental comparison of stereo algorithms", *In IEEE Workshop on Vision Algorithms*, September 1999.

[21] T. Poggio, V Torre and C. Koch, "Computational vision and regularization theory", *Nature*, 1985.

[22] S. Barnard, "Stochastic stereo matching over scale", *International Journal of Computer Vision, 3(1):17-32*, 1989.

[23] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and machine Intelligence*, 1984.

[24] Y. Boykov, O. Veksler and R. Zabih. "Markov Random Fields with efficient approximations". *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[25] S. Roy and I. Cox. "A maximum-flow formulation of the n-camera stereo corrispondence problem". *In International Conference on Computer Vision*, 1998.

[26] O. Veksler. "Efficient graph-based Energy Minimization Methods in Computer Vision". *PhD Thesis, Cornell Univeristy*, July 1999.

[27] L. Ford and D. Fulkerson. "Flows in Networks". *Princeton University Press*, 1962.