SCENE SEGMENTATION FROM DEPTH AND COLOR DATA DRIVEN BY SURFACE FITTING

Giampaolo Pagnutti, Pietro Zanuttigh

Department of Information Engineering, University of Padova

ABSTRACT

Scene segmentation is a very challenging problem for which color information alone is often not sufficient. Recently the introduction of consumer depth cameras has opened the way to novel approaches exploiting depth data. This paper proposes a novel segmentation scheme that exploits the joint usage of color and depth data together with a 3D surface estimation scheme. Firstly a set of multi-dimensional vectors is built from color and geometry information and normalized cuts spectral clustering is applied to them in order to coarsely segment the scene. Then a NURBS model is fitted on each of the computed segments. The accuracy of the fitting is used as a measure of the plausibility that the segment represents a single surface or object. Segments that do not represent a single surface are split again into smaller regions and the process is iterated until the optimal segmentation is obtained. Experimental results show how the proposed method allows to obtain an accurate and reliable scene segmentation.

Index Terms— Segmentation, Depth, Color, Kinect, NURBS

1. INTRODUCTION

Scene segmentation by way of images has attracted a huge amount of research, but it is an ill-posed problem and remains a very challenging task. Many segmentation techniques based on different insights have been developed, e.g., approaches based on graph theory [1], on clustering algorithms [2, 3], on region splitting and merging, and on many other techniques. The recent introduction of matricial Time-of-Flight range cameras and structured-light consumer depth cameras (e.g., Microsoft Kinect) has opened the way to the opportunity of combining depth information together with the color information for scene segmentation purposes. Within this perspective the segmentation problem can be formulated as the search for effective ways of meaningfully partitioning a set of samples featuring color and geometry information. Note how this is close to what happens inside the human brain where the disparity between the images seen by the two eyes is one of the clues used to separate the different objects in a scene together with prior knowledge and other features extracted from the color data acquired by the human visual system.

It is a very recent research field but some works addressing scene segmentation by way of color and geometry information have recently been published. A first possible solution is to perform two independent segmentations from the color image and the depth data, and then join the two results [4]. In [5] two likelihood functions, based on color and depth data, are combined together in order to segment the background from the foreground. Two different approaches for the segmentation of binocular stereo video sequences are presented in [6]: one, based on Layered Dynamic Programming and the other based on Layered Graph Cuts. Some other recent works try to jointly solve the segmentation and stereo disparity estimation problems, e.g., [7] and [8]. Clustering techniques can be exploited for joint depth and color segmentation as in [9] and [10]. In [11] we proposed a segmentation scheme based on spectral clustering that is able to automatically balance the relevance of the two clues. This paper proposes a novel scene segmentation scheme that includes the approach of [11] inside an iterative scheme where the segmentation is progressively refined by recursively splitting the segments that do not represent a single surface in the 3D space. The check is performed by fitting a NURBS surface over each segment and analysing the accuracy of the fitting.

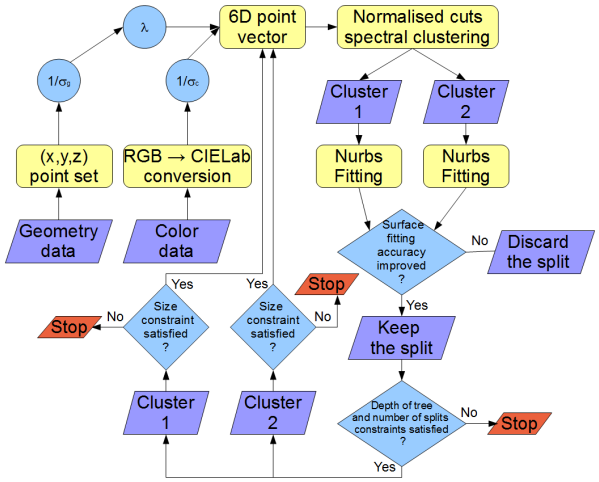

Fig. 1. Overview of the proposed approach.

The paper is organized in the following way: Section 2 presents the general workflow of the segmentation algorithm. Section 3 briefly recalls the employed joint color and depth segmentation scheme, while Section 4 presents the employed surface fitting algorithm. Finally Section 5 shows how the two steps are combined into the proposed approach. The results are presented in Section 6 and Section 7 draws the conclusions.

2. GENERAL OVERVIEW

Fig. 1 shows a general overview of the proposed approach. The color image and depth map are firstly converted to a unified representation and then fed into the proposed iterative segmentation algorithm. The acquired data is segmented into two clusters using both color and depth information [11]. Then a NURBS model is fitted over each of the two segments. The accuracy of the fitting is compared to the one obtained in the previous step for the same cluster (except for the first iteration). If the segmentation has allowed to obtain a

better fitting the process is iterated by recursively splitting the two segments, otherwise it is stopped. The process is iterated until it is not possible to obtain any improvement by further subdividing any of the produced segments.

3. JOINT COLOR AND DEPTH SEGMENTATION

Following an approach similar to [11] and [12], before entering the main loop of the proposed algorithm, a six-dimensional representation of the scene samples is built from the geometry and color data. After the joint calibration of the depth and color cameras it is possible to compute the 3D coordinates x,y and z of the 3D point of the scene corresponding to each sample in the depth map and to associate to it a vector containing the R, G, and B color components. Geometry and color then need to be unified in a meaningful way. Color values are converted to the CIELab perceptually uniform space, i.e., each sample p_i is represented by the vector

$$\mathbf{p_i^c} = [L(p_i), a(p_i), b(p_i)]^T \quad , i = 1, ..., N \tag{1}$$

Geometry can be simply represented by the 3D coordinates $x(p_i)$, $y(p_i)$, and $z(p_i)$, i.e. as:

$$\mathbf{p_i^g} = [x(p_i), y(p_i), z(p_i)] \quad , i = 1, ..., N \tag{2}$$

The scene segmentation algorithm should be insensitive to the relative scaling of the point-cloud geometry and should bring geometry and color distances into consistent representations, therefore the geometry components are normalized by the average σ_g of the standard deviations of the point coordinates obtaining the vectors $[\bar{x}(p_i), \bar{y}(p_i), \bar{z}(p_i)]$. Following the same rationale, the color information vectors $[\bar{L}(p_i), \bar{a}(p_i), \bar{b}(p_i)]$ are obtained by normalizing color data with the average σ_c of the standard deviations of the L, a and b components. From the above normalized geometry and color information vectors, each point is finally represented as:

$$\mathbf{p}_i^f = \begin{bmatrix} \bar{L}(p_i) \\ \bar{a}(p_i) \\ \bar{b}(p_i) \\ \lambda\bar{x}(p_i) \\ \lambda\bar{y}(p_i) \\ \lambda\bar{z}(p_i) \end{bmatrix}, i = 1, ..., N \tag{3}$$

where λ is a parameter balancing the contribution of color and geometry. High values of λ increase the relevance of geometry, while low values of the parameter increase the relevance of color information. For more details on the effect of this parameter and on how to automatically set it see [11].

The computed 6D vectors are then clustered in order to segment the acquired scene. Normalized cuts spectral clustering [3] is an effective approach for this task. This method is based on the partition of a graph representing the scene according to spectral graph theory criteria. The minimization is done using normalized cuts and accounts both for the similarity between the pixels inside the same segment and the dissimilarity between the pixels in different segments. The algorithm is very computationally expensive and several methods have been proposed for its efficient approximation. In the method based on the integral eigenvalue problem proposed in [13], the set of points is first randomly subsampled and then the subset is partitioned and the solution is propagated to the whole points set by a specific technique called Nyström method. In order to avoid small regions due to noise a final refinement stage removing regions smaller than a pre-defined threshold T_p is applied.

4. SURFACE FITTING ON THE SEGMENTED DATA

NURBS (Non-Uniform Rational B-Splines) are piecewise rational polynomial functions expressed in terms of proper bases, see [14] for a thorough introduction. They allow to represent freeform parametric curves and surfaces in a concise way, by means of control points. A parametric NURBS surface is defined as

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{4}$$

where the $\mathbf{P}_{i,j}$ are the control points, the $w_{i,j}$ are the corresponding weights, the $N_{i,p}$ are the univariate B-spline basis functions, and p, q are the degrees in the u, v parametric directions respectively.

In our experiments, we set the degrees in the u and v directions equal to 3. We set the weights all equal to one, thus our fitted surfaces are non-rational (i.e., spline). We also set the number of control points equal to ten both in the u and v parametric directions. These parameters turn out to be a reasonable choice, since they provide enough degrees of freedom to properly represent the shape of any common object, while ensuring at the same time the model simplicity needed for our method to be effective. A higher number of control points would indeed make the fitting error always small, independently on how the segmentation algorithm was successful in detecting the objects in the scene.

Then, we set the (u_k, v_l) parameter values corresponding to the points to fit as lying on the image plane of the camera, and we consequently determine the NURBS knots (needed for the definition of the $N_{i,p}$ basis functions) as in Chapter 9 of [14]. Notice that, since we fit each segment separately instead of the whole scene, the points to fit do not form a complete rectangular grid. Therefore, to avoid excessive surface oscillations, we first calculate a least-squares fitting plane through the segment points, and we use it to add extra points to be approximated by the final surface along a rectangular border outside the segment area, as shown in Fig. 2. Finally, by considering Eq. 4 evaluated at (u_k, v_l) and equated to the points to fit, we obtain an over-determined system of linear equations. We solve it in the least-squares sense by means of singular value decomposition (SVD), thus obtaining the surface control points.

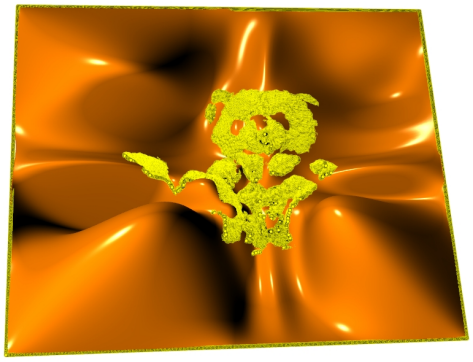

Fig. 2. Fitted surface (orange) approximating the points belonging to one of the segments (yellow). The extra points calculated onto the best fitting plane are the yellow ones along the surface borders.

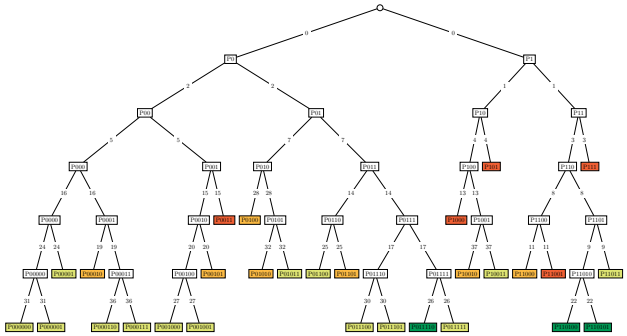

Fig. 3. Tree structure for the segmentation of a sample scene. The colored nodes correspond to the final segments. Red segments: further segmentation was attempted but rejected since not satisfying the MSE improvement constraint. Orange: the segmentation was rejected since one of the resulting sub-segments would be smaller than T_p. Light green: not split since smaller than $2T_p$. Green: stopped since the maximum tree depth T_d was reached.

5. ITERATIVE TREE STRUCTURED FITTING AND SEGMENTATION

After presenting the two main building blocks the complete segmentation procedure can now be described. The 6D representation of Eq. 3 is firstly built and used as input. We will denote with P the complete 6D point cloud. Then the image is segmented into two parts P_0 and P_1 using the segmentation algorithm of Section 3. A surface is then fitted on each of the two segments, obtaining the two surfaces S_0 and S_1. The MSE between the depth samples in P_0 and P_1 and the points obtained by sampling the two NURBS approximations S_0 and S_1 at the corresponding locations is computed thus obtaining the two values e_0 and e_1. In order to proceed to the next step a set of conditions must be satisfied, i.e.,:

- The size of P_0 and P_1 must be bigger than $2T_p$. If one of the two segments does not satisfy the constraint it is kept as part of the final solution and it is not split any more. This is consistent with the choice of not allowing segments smaller than T_p made in Section 3 since the split would produce at least one segment smaller than T_p.
- A maximum number T_d of recursive splits on each segment is set, when the segmentation tree of Fig. 3 reaches the maximum allowed depth the procedure is stopped on the corresponding branch.
- A maximum number of splits (i.e., segments) T_s is also set. Again when it is reached the procedure is stopped.

However at this first iteration the stop conditions are very unlikely to be reached and the procedure continues recursively by splitting the two point clouds P_0 and P_1 into two parts obtaining the sets P_{00}, P_{01} and P_{10}, P_{11} respectively. Note that the point clouds are sorted on the basis of the computed MSE and at each step the point cloud with the maximum MSE is processed, e.g., at the first step if $e_1 > e_0$ then $P1$ is processed firstly. In order to describe the general case let us assume that the segment P_i is considered for splitting (e.g., $i = 0$ or $i = 1$ at the first iteration): the segment is split into two subsegments P_{i0} and P_{i1} as before, the two NURBS approximations S_{i0} and S_{i1} and the MSE values e_{i0} and e_{i1} are also computed. At this point the weighted average of the MSE given by the new segmentation is compared with the one of P_i:

$$\frac{e_{i0}|P_{i0}| + e_{i1}|P_{i1}|}{e_i(|P_{i0}| + |P_{i1}|)} < T_e \tag{5}$$

where the weights are the cardinalities of the two sets and T_e is a suitable threshold (for the results we set $T_e = 0.8$). If the constraint of Eq. 5 is satisfied it means that the segmentation has improved the accuracy of the scene representation by recognizing the different surfaces (i.e., objects) in the scene. In this case it must be kept and the two sub-segments P_{i0} and P_{i1} are further subdivided with the same procedure. If the constraint is not satisfied the segmentation is discarded, the segment P_i is kept as a single object and no further processing is done on this branch of the tree. Before proceeding the previously introduced conditions are also checked on each segment before splitting, i.e.:

$$(|P_i| > 2T_p) \wedge (Depth(T_i) < T_d) \wedge (count(i) < T_s) \tag{6}$$

where $Depth(T_i)$ is the depth of the $i-th$ node and $count(i)$ is the number of splits made until the current iteration. The same procedure is applied recursively to all the sub-segments generated during the computation until the condition of (5) is satisfied and none of the stopping conditions is violated leading to a tree structure similar to the one of Fig. 3.

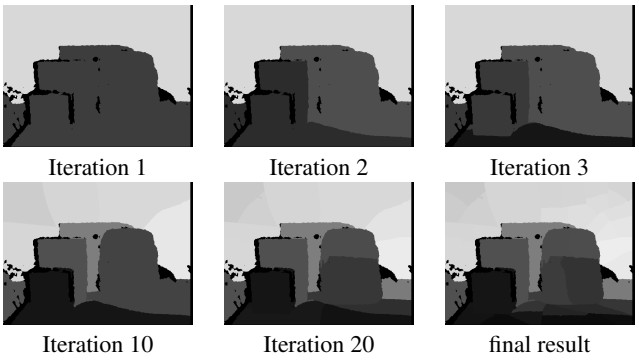

Fig. 4. Execution of the proposed algorithm on the scene of Fig. 5, first row. The images show the segmentation after 1,2,3,10 and 20 iterations and the final result.

6. EXPERIMENTAL RESULTS

In order to evaluate the performances of the proposed approach a set of images and depth maps of some sample scenes has been acquired with the Kinect sensor (they are available at `http://lttm.dei.unipd.it/downloads/segmentation`). After calibrating the sensor with the method of [15], it is possible to build the representation of Section 3. The results are shown in Fig. 5: the first two columns show the acquired images and depth maps while the third column shows the results of the application of the method of [11], that roughly corresponds to directly segment the image into the desired number of clusters in a single step with the method of Sec. 3. The fourth column shows the result of the application of the proposed method. It is possible to see that it allows to obtain better performances than the compared approach on all the considered scenes, since the objects are well recognized and there are almost no segments extending over multiple objects at different depths. This is due to the fact that if a segment spans over

Fig. 5. Segmentation of some sample scenes with the proposed method and with the approach of [11].

multiple objects with different 3D positions the surface fitting would not be accurate and the proposed method will further segment the region until it is split into the various objects. This can be noticed also from Fig. 4 that shows the partial results of the execution of the proposed method after $1, 2, 3, 10$ and 20 iterations, it is possible to see that the various objects in the scene get progressively separated while more and more splits are performed. Another strength point is that the proposed approach is able to recognize the different objects in the scene but at the same time avoids oversegmenting them. In particular note how some objects and people that are divided into several parts by the approach of [11] (e.g., the people in rows 4 and 5) are instead kept together by the proposed method since the surface estimation and evaluation scheme allows to recognize that they are part of a single surface. The edges of the object are also very well captured. Finally note that, even if the MSE evaluation is done on geometry data, in the clustering step the proposed approach makes use also of color information, as it is possible to notice from some details, e.g., the hexagons on the soccer ball.

7. CONCLUSIONS

In this paper we proposed a novel scheme for the joint segmentation of color and depth information. The proposed approach not only exploits color and depth information together to improve the segmentation performances but also exploits a surface fitting scheme to determine if the segmentation has correctly divided the 3D surfaces present in the scene. Experimental results demonstrate its effectiveness and its ability to avoid oversegmentation by the analysis of the 3D surfaces corresponding to the objects in the scene. Further research will be devoted to a more advanced evaluation of the surface fitting based not only on the MSE but also on parameters of the fitted surface, e.g., the local curvature. The replacement of the binary splits with the possibility of creating multiple clusters in a single step will also be considered. Finally parallel implementations will be considered to optimize the computation time.

8. REFERENCES

[1] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sept. 2004.

[2] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[4] F. Calderero and F. Marques, "Hierarchical fusion of color and depth information at partition level by cooperative region merging," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP 2009*, 2009, pp. 973–976.

[5] L. Wang, C. Zhang, R. Yang, and C. Zhang, "Tofcut: Towards robust real-time foreground extraction using a time-of-flight camera," in *3DPVT*, 2010.

[6] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer segmentation of binocular stereo video," in *IEEE conf. on Computer Vision and Pattern Recognition (CVPR)*, june 2005, vol. 2, p. 1186 vol. 2.

[7] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr, "Joint optimisation for object class segmentation and dense stereo reconstruction," in *Proceedings of the British Machine Vision Conference*, 2010.

[8] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo- joint stereo matching and object segmentation," in *IEEE conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2011.

[9] A. Bleiweiss and M. Werman, "Fusing time-of-flight depth and color for real-time segmentation and tracking," in *Proceedings of DAGM 2009 Workshop on Dynamic 3D Imaging*, 2009, pp. 58–69.

[10] Marcus Wallenberg, Michael Felsberg, Per-Erik Forssén, and Babette Dellen, "Channel coding for joint colour and depth segmentation," in *Proceedings of Pattern Recognition 33rd DAGM Symposium*, Frankfurt/Main, Germany, September 2011, vol. 6835 of *Lecture Notes in Computer Science*, pp. 306–315, SpringerLink.

[11] C. Dal Mutto, P. Zanuttigh, and G.M. Cortelazzo, "Fusion of geometry and color information for scene segmentation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 5, pp. 505–521, 2012.

[12] C. Dal Mutto, P. Zanuttigh, and G.M. Cortelazzo, "Scene segmentation assisted by stereo vision," in *Proceedings of 3DIMPVT 2011*, Hangzhou, China, May 2011.

[13] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[14] Les Piegl and Wayne Tiller, *The NURBS Book (2Nd Ed.)*, Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[15] Daniel Herrera C, Juho Kannala, and Janne Heikkila, "Joint depth and color camera calibration with distortion correction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2058–2064, Oct. 2012.