PREDICT POSE AND POSITION OF RIGID OBJECTS IN VIDEO SEQUENCES

*Fuensanta Torres**, *Walter G. Kropatsch, and Nicole M. Artner*

PRIP, Vienna University of Technology, Austria

ABSTRACT

A novel system is presented for predicting the pose, the position and the appearance of 3D rigid objects in video sequences. We consider rigid objects which can be approximately modeled by a convex polyhedral shape. Our approach works in monocular videos and where the position of the camera is fixed. To address this, we propose a two-step approach, first a process to obtain the information and the second part is the process to apply the information. The system integrates a 3D model-based tracking, pose determination and aspect graph indexing. The aspect graph provides advantages to deal with self-occlusions and changes in appearance. The tracking method decreases the needed time to detect and delineate objects in the frames and compensates for pose module error. The pose module compensates for tracking error [1].

***Index Terms*—** tracking, pose estimation, aspect graph, key characteristics

1. INTRODUCTION

Shape appearance and motion of real-world objects obey the laws of physics and also their changes happen in a smooth manner. This implies that there exists a strong correlation in the temporal evolution of image content. Therefore, in our approach we plan to use the aspect graph of a 3D object to deal with changes in appearance due to the movement of this object in front of the camera. The aspect graph is a graph with a node for every aspect (view) and edges connecting adjacent aspects. An aspect is the appearance of an object, when seen from a specific view point. Fig. 1 illustrates the aspect graph of a die. The use of aspect graphs is not new (e.g. [2, 3, 1]). The aspect graph is used to determine when features will disappear or become difficult to track and to predict where and when the new features will appear. Our system integrates a 3D model-based tracking [4], pose determination and aspect graph indexing. Furthermore we are studying the possibility of automatically finding a group of key characteristics and a set of rules which form all the possible movements of a 3D tracked object. Our approach could be divided into two parts, first a process to obtain the information which finds out the key characteristics of the tracked object. The second part is the process to apply the information [5]. The latter is currently being worked on and will be described in a later paper, it predicts the pose, the position and the appearance of the 3D object in the future frames. We have been studying the motion of rigid objects which can be approximately modeled by a convex polyhedral shape. The direct extension to non polyhedral object can be considered if a 3D description of the object is available. This paper deals with the starting point of this method, a process to automatically obtain the information. The input of this task is a video sequence of a convex polyhedron and the output is the a 3D model and information about its shape, its appearance, its position and its orientation.

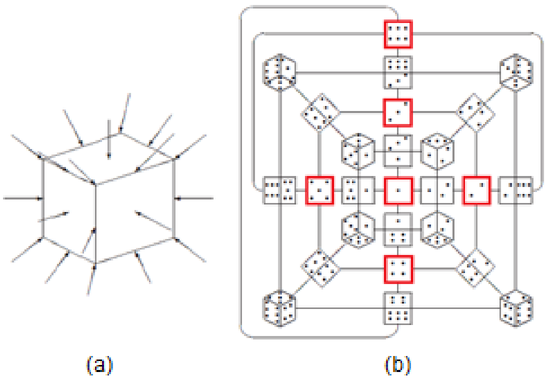


Fig. 1. a) Different viewing angles. b) Aspect graph encoding the relationships between the faces of a die.

The rest of the paper is organized as follows: Section 2 describes the process to extract the shape, the appearance, the position and the orientation. Section 3 presents the 3D model. The experimental results revealing the efficacy of the method are described in Section 4. Finally, the paper concludes along with discussions and future work in Section 5.

2. PROPOSED PROCESS

The relevant steps to obtain the shape, the appearance, the position and the orientation of 3D tracked object in a video sequence are summarized in this section.

*Thanks to Doctoral College on Computational Perception (Vienna University of Technology, Austria) for funding.

2.1. 3D Shape

The proposed process to extract the 3D shape of a convex polyhedron runs four consecutive steps which are described below.

2.1.1. 2D silhouette (Canny edge detector)

To obtain the 2D shape or the 2D silhouette of the polyhedron we use the Canny edge detector [6, 7]. An edge detector is the most common approach for detecting meaningful discontinuities in intensity values.

2.1.2. Parallel edges (Hough Transform)

Ideally, the Canny Edge Detector discussed in the previous section should yield pixels laying only on edges. In practice, the resulting pixels seldom characterize an edge completely because of noise, breaks in the edge from non uniform illumination, and other effects that introduce spurious intensity discontinuities. Thus edge-detection algorithms typically are followed by linking procedures to assemble edge pixels into meaningful edges. Perhaps the most often used approach to find and link line segments in an image is the Hough transform [6].

The Hough transform of a line segment in 2D can be described with two real-valued parameters using the Hessian normal form for representing lines (eq. 1).

$$x \cdot cos\theta + y \cdot sin\theta = \rho; \tag{1}$$

The ρ value is the distance between the line and the origin, while θ is the angle of the vector from the origin to this closest point (x, y).

Finding parallel lines in a Hough transform can be challenging [8]. Ideally, the parallel lines have the same θ, which means that they must lie in the same column in the Hough domain. In practice, this is not always the case because of the quantization in the image space, the quantization in parameter space of the Hough transform, as well as the fact that edges in typical images are not parallel due to non ideal parallel projection and the discretization. One strategy to overcome this problem is our three-stage strategy:

1. We introduce a discretization step of the θ -axis equal to 5 degrees, which is enough to suppress the effect of the a non ideal parallel projection.

2. The total sum of each column of the Hough transform, is always the same value and equal to the total numbers of points in the image domain. However there are peaks with high values which correspond to the edges. Therefore once each element of the Hough domain is raised to the power of a high value, the total sum of each column does not have anymore the same number, the columns with parallel edges have the highest numbers.

3. The parallel edges in the 2D shape are the cells containing the highest value in each column highlighted in the last step.

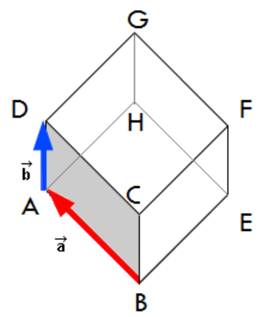


Fig. 2. (Best viewed in color) Corners of a die: a) different images of a die; b) Result of Canny edge detector; c) Edges in the 2D silhouette; and d) visible corners (see text for details)

2.1.3. Edges between visible faces (Histogram)

Between two visible faces always there is a visible edge. In the case of three visible faces these edges are the lines segments joining two cut-off points between the edges in its silhouette. We use the gray-scale value histogram when the polyhedron has exactly two visible faces. An edge in the image space represents the directionality of the brightness changes in the image. Therefore, we run over the whole length of the silhouette and repeatedly dividing its silhouette in two parts. We calculate the histogram of these two parts. Finally we compare how similar are these two histograms. The line which splits the polyhedron in two parts with the most dissimilar histograms is the edge between the two visible faces.

2.1.4. Corners

Knowing all the visible edges, it is possible to extract the visible corners of the polyhedron. The visible corners are the cut-off points between the edges. In order to calculate the coordinates of the non visible corners we calculate the third coordinate (z) of the visible corners. First of all, it is necessary to establish the origin of the z-axis, the origin will be the closest point to the camera. In the case of only one visible face all the visible corners have z equal to 0. In the case of two or three faces we use the aspect graph to identify the current viewing angle. To calculate the z coordinate of the rest of corners we use the knowledge of the geometry of the polyhedron. For instance in the case of a cube (fig. 2.) the sizes of its edges must be always the same (eq. 2) and its vectors must be orthogonal (eq. 3).

$$|\vec{a}| = \left|\vec{b}\right|; \tag{2}$$

$$\vec{a} \cdot \vec{b} = 0; \tag{3}$$

With the edges $\vec{a} = (a_x\ a_y\ a_z)$ and $\vec{b} = (b_x\ b_y\ b_z)$.

Hence, we solve a system of two equations with two unknown values, a_z and b_z.

To calculate the coordinates of the invisible corners we apply again the knowledge of the geometry. In the cube of the fig. 2 the invisible corner is H (eq. 4).

$$H = E + \vec{a}; \tag{4}$$

2.2. The Position

The position of the polyhedron can be described by the center of gravity, this is the center of mass (C). Center of Mass for a convex polyhedron (eq. 5).

$$C = \frac{\Sigma_{i=1}^{k} x_i}{k}; \tag{5}$$

x_i is a corner of the polyhedron and k is the total number of corners.

2.3. The Orientation

In geometry the orientation or the pose of an object, is part of the description of how it is placed in the space. Euler angles parameterize orientation using only three numbers (α, β, σ). These are the rotation angles around the Z, Y, and X axis respectively [9].

2.4. The Appearance

In this section we identify which are the number of considered aspects of the polyhedron and which are the representative points of each aspect. In the case of a convex polyhedron we consider one aspect per each face. Moreover, the hallmarks are specific for each type of convex polyhedron.

3. CONSTRUCTION OF THE 3D MODEL

We build a multi-view appearance model [4]. This 3D model has the same geometry and the same topology (the aspect graph) as the tracked object. We extract the geometry of the tracked object from the video sequence. We look for one frame of the video sequence in which appears only one visible face (aspect). We extract the corners of the visible face and thereby we obtain the dimensions of this face. We build a 2D template per each aspect with the set of representative points which identify each aspect(view). Finally, we place the templates in their corresponding faces of the 3D model.

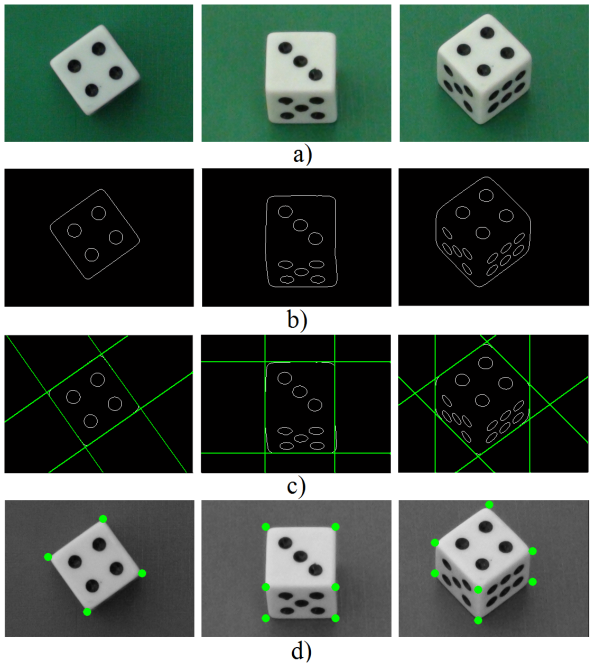


Fig. 3. (Best viewed in color) a) different images of a die; b) result of Canny edge detector; c) edges in the 2D silhouette; and d) visible corners (see text for details)

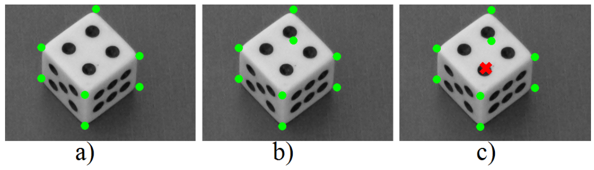


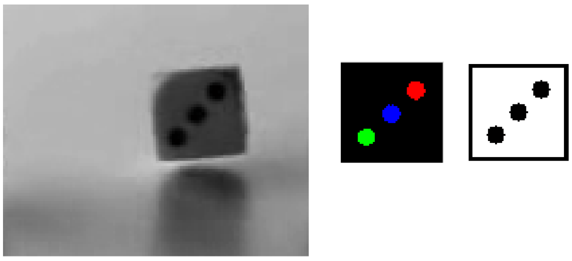
Fig. 4. (Best viewed in color) a) the visible corners; b) all the corners; and c) all the corners and the center of gravity (with a x), (see text for details).

4. EXPERIMENTAL RESULTS

Fig. 3. a) shows three images of a die (a convex polyhedron) from different views, with one, two and three visible faces respectively; b) the same images after the Canny edge detector; c) The parallel edges of the 2D silhouette are highlighted (green lines); and d) The visible corners are highlighted (green dots) Fig. 4. a) shows the visible corners; b) the visible corners as well as the non visible corner; and c) all the corners and the center of gravity, marked with a x. Fig. 5. is an example to find the representative points of a die. Fig. 6. is the 3D model of a die. Fig. 7. is another example of a cube, a poker die where our approach extracts the corners.

5. CONCLUSION

This paper has proposed the starting point of a novel approach. The systems integrates pose estimation, tracking and



a) b) c)

Fig. 5. a) image of a die; b) connected components; and c) geometry and representative points of the face 3

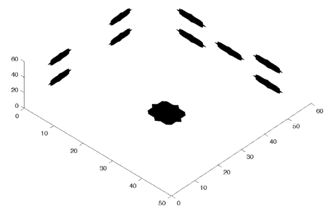


Fig. 6. 3D multi-view appearance model of a die

a) b) c)

Fig. 7. a) one view of a poker die; b) edges in the 2D silhouette are highlighted; and c) the visible corners are highlighted

aspect graph indexing. It works in monocular videos, where the position of the camera is fixed and the 3D tracked object is known. The starting point has been an approach to obtain the shape, the position, the orientation, the appearance and to build a 3D multi-view appearance model in a convex polyhedron. The second part is the tracking a pose algorithm and a new tracking algorithm. This one is currently being worked on and will be described in a later paper. The tracking algorithm method looks for to find the minimum number of points needed for predict future poses and positions and also to correct possible errors or variations. Moreover, our final goal is to use this technique for tracking real world objects (e.g. cars).

6. REFERENCES

[1] S. Ravela, B. Draper, J. Lim, and R. Weiss, "Adaptive tracking and model registration across distinct aspects," *International Conference on Intelligent Robots and Systems, pp. 174-180*, 1995.

[2] J.-S. Hu, T.-M. Su, C.-W. Juan, and G. Wang, "3d object tracking using mean-shift and similarity-based aspect-graph modeling," *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 2383-2388*, 2007.

[3] K. W. Bowyer and C. R. Dyer, "Aspect graphs: An introduction and survey of recent results," *International Journal of Imaging Systems and Technology, pp. 315-328*, 1990.

[4] O. Javed A. Yilmaz and M. Shah, "Object tracking: A survey," *ACM Comput. Surv., vol. 38, no. 4, pp. 1-45*, vol. 13, no. 1, pp. 234–778, 2006.

[5] W. G. Kropatsch, "When pyramids learned walking," *In Jan Olof Eklundh Eduardo Bayro-Corrochano, editor, The 14th International Congress on Pattern Recognition, CIARP 2009, volume 5856 of Lecture Notes in Computer Science, pp. 397- 414*, 2009.

[6] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing*, Prentice-Hall, 2004.

[7] M. Lv, H. Su, and Y. Li, "An adaptive canny detector with new differential operator," *Wireless Communications Networking and Mobile Computing (WiCOM), pp. 1- 4*, 2010.

[8] T. Tuytelaars, M. Proesmans, and L. Van Gool, "The cascaded hough transform," *Proc. IEEE Int'l Conf. on Image Processing (ICIP-97), pp.736-739*, 1997.

[9] F. Dunn and I. Parberry, *3D math primer for graphics and game development*, Wordware Publishing, 2002.