Top-Down Tracking and Estimating 3D Pose of a Die

Fuensanta Torres* and Walter G. Kropatsch

PRIP, Vienna University of Technology, Austria
http://www.prip.tuwien.ac.at

Abstract. Real-time 3D pose estimation from monocular image sequences is a challenging research topic. Although current methods are able to recover 3D pose, they are severely challenged by the computational cost. To address this problem, we propose a tracking and 3D pose estimation method supported by three main pillars: a pyramidal structure, an aspect graph and the checkpoints. Once initialized the systems performs a top-down tracking. At a high level it detects the position of the object and segments its time-space trajectory. This stage increases the stability and the robustness for the tracking process. Our main objective is the 3D pose estimation, the pose is estimated only in relevant events of the segmented trajectory, which reduces the computational effort required. In order to obtain the 3D pose estimation in the complete trajectory, an interpolation method, based on the aspect graph describing the structure of the object's surface, can be used to roughly estimate the poses between two relevant events. This early version of the method has been developed to work with a specific type of polyhedron with strong edges, texture and differentiated faces, a die.

Keywords: tracking, 3D pose estimation, pyramid, checkpoints, aspect graph.

1 Introduction

The proliferation of high speed videos, high-end computers and the need for automated video analysis have generated an increasing interest in visual tracking and pose estimation algorithms. The higher resolution of the images and the higher frame rate increase the data rate by a higher factor than the increase in computing power. This paper addresses the challenging problem of real-time tracking and 3D pose estimation exploring the efficient use of knowing the past for predicting and for verifying the future. Selecting the right features for tracking plays a critical role [3]. Nowadays, the illumination changes, the partial occlusion and the matching errors are simple to achieve with localized features [7]. However, computation of descriptors that are invariant across large view changes is usually

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

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 492–500, 2012.

expensive [15]. To overcome this weakness, the state of the art feature descriptors, detect and match points in successive images, in a non-recursive way, [10], [1], [8]. SIFT [8] is known to be a strong, but computationally expensive feature descriptor and on the contrary Ferns [10] classification is fast, but requires large amounts of memory. Therefore, our work investigates the applicability of a new markerless tracking method based on the checkpoints. Checkpoints are a small group of 3D points on the known object surface, which allow reliable tracking and preserve the structure. These are robust to illumination changes, computationally cheap and do not require large amount of memory. Moreover, they are 3D points. Therefore, once initialized their positions in the next frame can be predicted assuming smooth movement, without the need to back-project the 2D locations to obtain the 3D pose.

Top-down tracking encodes the current frame into a hierarchical structure, a pyramid, which reduces the search cost and allow large view changes. The use of a hierarchical approach for tracking have been widely used in the literature [16], [5], [9]. The main drawback of theses approaches have been the high computational cost to build a pyramid per frame. To overcome this weakness, we can use the computational power and increasing programmability of the graphics processing unit (GPU) present in modern graphics hardware that provides great scope for acceleration of computer vision algorithms which can be parallelized [13].

In order to reduce the computational effort our method distinguishes relevant (frame with only one visible face of the die) and normal events (with two or three visible faces) of the time-space trajectory of the object. The changes between two relevant events are handle by the aspect graph [12], [11].

The rest of the paper is organized as follows: Sec.2 describes the different structures and processes of this approach. Sec. 3 presents the top-down tracking and 3D pose estimation method. The experimental results revealing the efficacy of the method are shown in Section 4. Finally, the paper concludes along with discussions and future work in Section 5.

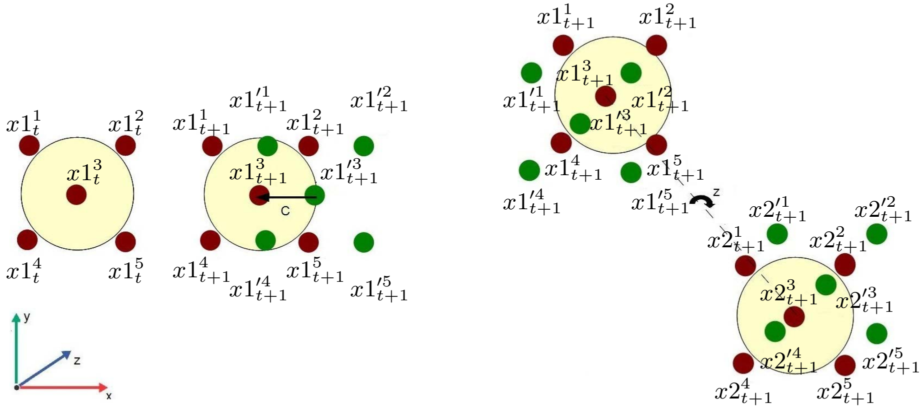
2 Definitions

We begin by providing some necessary definitions.

2.1 Checkpoints

Checkpoints are a small group of points characterizing local and salient features embedded in the object's surface and allowing to detect and correct displacements in the image frame. They require to distinguish between the background and the foreground of the object. This early version of the method is based on a strong foreground-background contrast, the background and the foreground points are differentiated by their gray values.

Let S= (I_t, I_{t+1}..., I_{t+k}) be an image sequence. The initial estimation of a group of checkpoints location in time t ($x1_t$= ($x1_t^1$, $x1_t^2$, $x1_t^3$, $x1_t^4$ and $x1_t^5$)) can be found by giving some correspondences between 3D points in the object model and their projections in I_t [7]. Checkpoints are projected into the current image $x1'_{t+1}$ frame and the corresponding pixel values checked whether they belong to the object or the background. Based on the result the correction is estimated that brings the object back to a location where the checkpoints are appropriately placed in the image $x1_{t+1}$. The correction (C= (s or/and T or/and R)) is the uniform scale (s), the translation (T) and the rotation (R) to get $x1_{t+1}$ from $x1'_{t+1}$ in the current frame I_{t+1}. For this purpose, considering a circular target, five checkpoints ($x1'_{t+1}$), which preserve the order, placed as shown Fig. 1 a), $x1^1$, $x1^2$, $x1^4$, $x1^5$ in the background and $x1^3$ in center of the object, in the foreground, are enough to detect the translation and the scale error. However, to estimate the rotation error, at least two groups of checkpoints are needed ($x1_t$ and $x2_t$) (Fig. 1 b)).



a) Translation b) Rotation

Fig. 1. Predicting and correcting translations and rotations of checkpoints

2.2 Prediction-Estimation-Correction

This section defines the Prediction-Estimation-Correction method (PEC) of the checkpoints' positions. Let I_t be the current frame and x_t be the checkpoints' locations in time t. Using a motion model[1], the checkpoints are predicted forward for one frame, x'_{t+1}. First, it checks if x'^1_{t+1}, x'^2_{t+1}, x'^4_{t+1}, x'^5_{t+1} are placed in the background(0) and x'^3_{t+1} in the foreground(1) of the image, otherwise the prediction is incorrect. When an error has been detected, it estimates the location where the checkpoints are appropriately placed in the image x'_{t+1}. The estimation method is based on a table with the possible cases of prediction errors and their respective estimation (Tab. 1). The table has been built considering all the possible movements of the prediction with respect to the real projection and their optimal improvement. Moreover, its efficacy has been demonstrated in the experimental results. In the table there are five columns (x'^1, x'^2, x'^3,

[1] here is used the 3D affine motion model.

x'^4, x'^5), which represent one group of checkpoints illustrated in Fig. 1 a) and the last column (x”) is the translation or the scale needed to get the estimated appropiate position from x’.

Table 1.

values at prediction					
x'^1	x'^2	x'^3	x'^4	x'^5	x''
0	0	$*$	0	1	$\searrow$
0	0	$*$	1	0	$\swarrow$
0	0	$*$	1	1	$\downarrow$
0	1	$*$	0	0	$\nearrow$
0	1	$*$	0	1	$\rightarrow$
0	1	$*$	1	1	$\searrow$
1	0	$*$	0	0	$\nwarrow$
1	0	$*$	1	0	$\leftarrow$
1	0	$*$	1	1	$\swarrow$
1	1	$*$	0	0	$\uparrow$
1	1	$*$	0	1	$\nearrow$
1	1	$*$	1	0	$\nwarrow$
0	0	1	0	0	s
1	1	1	1	1	$1/s$

The zeros in the table mean that the value of x'^i is close to the background, the one appears when it is more similar to the foreground and the * means that this checkpoint does not have any effect in the estimation. For instance, the case of Fig. 1 a) corresponds to the box in Tab. 1, x'^1, x'^3, x'^4 are equal to 1 while x'^2 and x'^5 are equal to 0. Therefore, the estimation(x”) is a translation of the prediction to the left. The direction and the sense of the arrows describe the translations for correction. The correction step finds the relationship between the estimated position of all groups ($x''1$, $x''2$ ...$x''i$) of the current frame and their prediction ($x'1$, $x'2$ ...$x'i$).

This least-squares problem in the 3D space is solved by using Horn [4], which returns the uniform scale factor (s), the rotation matrix (R_{3x3}) and the translation vector (T_{3x1}) needed to get the correction (x) from the prediction (x’)(eq. 1)

$$x = (s \cdot R_{3x3} + T_{3x1}) \cdot x'; \tag{1}$$

2.3 Recall of the Maximum Pyramid

The structure of a regular pyramid can be described as an array hierarchy in which each level l^t is at least defined by a set of nodes N_l. A node of a regular pyramid can be determined by its position (i, j, l) in the hierarchy, being l the level of the pyramid and (i, j) its (x, y) coordinates within the level. On the base level of the pyramid, the nodes are the pixels of the input image. Each pyramid level is recursively obtained by processing the level below. The children-parent relationships are fixed and for each node in level l+1, there is a reduction window of children at level l. We have selected a 2x2/4 pyramid [2] other types are also under investigation. To detect bright spots in images we use the Maximum pyramid, which uses the maximum as reduction function. The top of the pyramid receives the maximum gray value of the base image. There is a closed chain of links between the maximum in the base and the top. This can effectively be used to find its location top-down. Small non-maxima holes disappear quickly [6].

2.4 Aspect Graph

An aspect describes the appearance of an object from a specific view point. Views of one aspect may differ by continuous deformations but they all have the same topology. The appearance from one aspect to another aspect changes, i.e.

a new surface patch becomes visible, another one disappears. The aspect graph is a graph with a node for every aspect and edges connecting adjacent aspects. Therefore, it allows us to know the relationship between each aspect (Fig. 2).

3 Top-Down Tracking and Pose Estimation

The novel approach for target localization and 3D pose estimation is described in this section. The first step of tracking is to obtain a hierarchical representation of the current frame. In order to decrease the computational cost, we assume that the object does not move very much from one frame to the next one as well as their backgrounds are quite similar. Having the pyramid for the previous frame, it subtracts two consecutive frames to update in this pyramid only the information corresponding to their differences. Once the pyramid is available, the system performs the top-down tracking method (Fig. 3). This method segments the time-space trajectory of the object. It recovers the object position in all the frames, which increases the stability and the robustness for the tracking process. Although, the pose estimation is obtained only in the relevant events. We can use an interpolation method, based on the aspect graph, between two pose estimations to roughly estimate the pose in the intermediate frames. The top-down process is the following:

1. Target localization: works at the top level of the pyramid l_T^t. In this level the target region has approximately homogeneous color. It selects the nodes with this color, where the object is placed $N_{l_T}^t$.
2. Trajectory Segmentation: Each node below the target object in the top level $N_{l_T}^t$ is linked to its children. This top-down process continues until the method estimates if the current frame is a relevant or a normal event. In the case of a normal event, the object position is estimated. Otherwise, it determines its 3D pose estimation.

3.1 Object Position

The position of an object is determined in normal events and at the first level where the number of nodes of the target is bigger than a given threshold. It

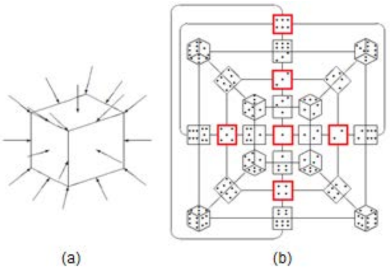


Fig. 2. a) Different viewing angles. b) Aspect graph of a die.

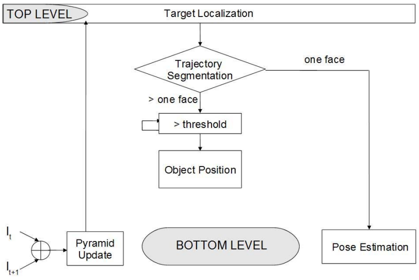


Fig. 3. Illustration of the Top-down tracking and pose estimation algorithm

is chosen in such a way that the completed target region has approximately homogeneous color, which is a compromise between homogeneous color in $ROI^t_{l_i}$ and precision in the PEC method. If the threshold raises, the precision increases but the homogeneity in the color decreases. At the highest level where $ROI^t_{l_i} <$ threshold, a group of checkpoints $(x1_{t,l})$ and the PEC method are used to to estimate the position of the object in the current frame.

3.2 Pose Estimation

This method works with relevant events. Two groups of checkpoints ($x1_{t,0}$ and $x2_{t,0}$) and the PEC method are used at the base level to estimate the 3D pose[14].

4 Experiments

In this section we demonstrate the effectiveness of our approach using a video sequence S= (I_t, I_{t+1}..., I_{t+k}) of a die. Figs. 5 a) and 5 b) have the same nine frames of the video sequence (I_1, I_2..., I_9). They show the prediction of the checkpoints's positions (green points) and the result of the PEC method (red points).

The object position method (Sec. 3.1) allows abrupt displacements and large view changes (Fig. 5 a)). Although, it does not detect rotation changes and its prediction is not very accurate, as can be seen in Tab. 2. This shows the biggest error in pixels between the estimated position of the center point of the die and its real position in the base level. We calculated the biggest error in the prediction and also in the correction in the frames I_1, I_{10}, I_{20}, I_{30}, I_{40}, I_{50}.

Otherwise, the isolated pose estimation method (Sec. 3.2) is not robust to large view changes and translations (Fig. 5 b)). But this refinement step increases the accuracy of the method. As shown in Tab. 2 the errors are smaller than 5 after of the PEC method, except in the case of I_{50}, where the die is lost.

We have observed that the size of the target in ROI at the different levels of the pyramid strongly depends of the number of visible faces of the die in the current frame. Fig. 4 shows a graph with the size of the die in ROI for the different frames of a video sequence at a given level. As can be seen in the minimum values of the graph there are frames with only one visible face and in the maximum values there are views with three visible faces. Our current method

Table 2. Errors in pixels at the base level with two methods

object position method (Sec. 3.1)		
Frame	Prediction error	Correction error
I_1	8.6	4.4
I_{10}	12.6	12.6
I_{20}	13	10
I_{30}	10	13.1
I_{40}	14.5	19
I_{50}	10.5	8

pose estimation method (Sec. 3.2)		
Frame	Prediction error	Correction error
I_1	6.9	1.5
I_{10}	9	1
I_{20}	13.29	0
I_{30}	5.5	2.7
I_{40}	11	5
I_{50}	28.9	42.3(lost)

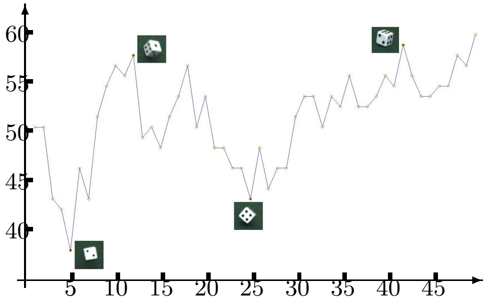
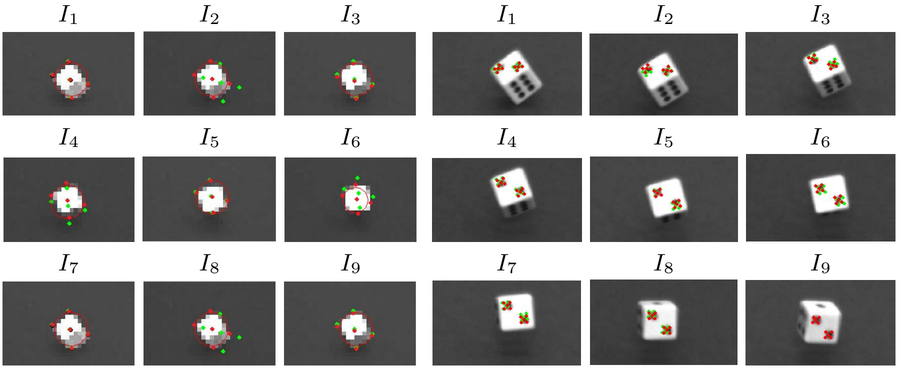


Fig. 4. Size of the target in ROI for each frame of a video sequence

to segment the time-space trajectory fails in some cases, which strongly depend on the position of the die in the frame (related to the shift variance problem of non overlapping pyramids). We are working to overcome this weakness.

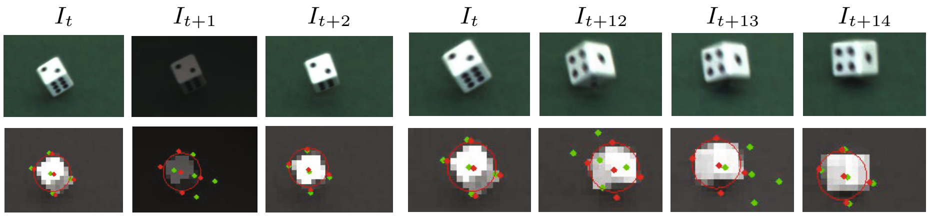
Finally, the strengths of our method have been proven with different experiments:

- Robustness to illumination changes: We changed the illumination in the training sequence Fig. 6 a). As can be seen in the bottom row the checkpoints handle a very abrupt lighting changes.
- Insensitivity to large view changes: Thanks to the object position method (Sec. 3.1), the algorithm can handle large view changes and it also updates the motion model. Fig. 6 b) shows in the top row the frames I_t, I_{t+12}, I_{t+13} and I_{t+14} of a video sequence. As can be seen in the bottom row, it localizes the die in the frame I_{t+12} and updates the motion model. Therefore, the prediction in the frame I_{t+14} is quite accurate.
- Computationally Cheap: Once initialized, the pyramid of the current frame I_{t+1} is the same pyramid as the previous frame I_t, where only the differ-



a) Object position b) Pose estimation

Fig 5. Prediction and correction of checkpoints



a) Robustness to illumination changes. b) Insensitivity to large view changes.

Fig 6. Experiments to prove the strengths of our method

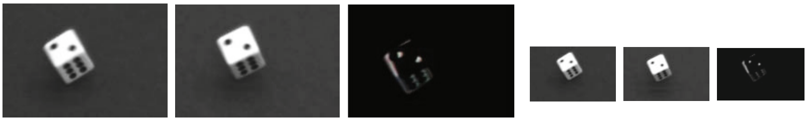


Fig 7. Two consecutive frames and their differences at the base level (l^0) and at l^1 respectively

ences between I_{t+1} and I_t have been updated. Fig. 7 shows two consecutive frames and their differences on the top row, while the row below shows the differences at the higher level(l^1). In this particular example, the dimensions of I_t and I_{t+1} are equal to 640x480= 307200 pixels, there are 5020 nodes different at the base level (l^0), 17 at l^1, 10 at l^2 , 2 at l^3 and 0 in the rest of levels.

5 Conclusions and Future Work

This paper has proposed a novel approach to track and to estimate the 3D pose of a (partially) known object. To demonstrate the new concept we have chosen a die because of it's simple structure: six well distinguished faces. We have developed a marker-less 3D tracking, which extracts the checkpoints with a top-down method and matches them across images, in a recursive way. This is robust to changes to illumination, computationally cheap and do not require large amount of memory. In order to reduce the search cost and allow large view changes, the method is based on the Maximum Pyramid. Moreover, the time-space trajectory of the object was divided into relevant and normal events, that reduces the computational effort and allows us to focus only on those relevant frames of the video stream. The 3D pose was estimated only in the relevant events. Although, the target was localized in all the events to increase the stability and the robustness for the tracking process. Finally, in order to obtain the 3D pose estimation in the complete trajectory, the future work will be an interpolation method, based on the aspect graph describing the structure of the object's surface, can be used to roughly estimate the poses between two relevant events [11].

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.J.: Surf: Speeded up robust features. Computer Vision and Image Understanding (CVIU) 110(3) (2008)
2. Brun, L., Kropatsch, W.G.: Construction of Combinatorial Pyramids. In: Hancock, E.R., Vento, M. (eds.) GbRPR 2003. LNCS, vol. 2726, pp. 1–12. Springer, Heidelberg (2003)
3. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. International Journal of Computer Vision, 1–26 (2011)
4. Horn, B.K.P.: Closed-form solution of absolute orientation using unit. J. Optical Society of America 4(4), 629–642 (1987)
5. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Proc. of ISMAR (2007)
6. Kropatsch, W.G., Bischof, H., Englert, R.: Hierarchies. In: Digital Image Analysis: Selected Techniques and Applications, ch. III Robust and Adaptive Image Understanding (2001)
7. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects: A survey. Foundations and Trends in Computer Graphics and Vision 1(1) (2005)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. Computer Vision and Image Understanding 20 (2004)
9. Marfil, R., Molina-Tanco, L., Rodríguez, S.F.: Real-time object tracking using bounded irregular pyramids. Pattern Recognition Letters (2007)
10. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast keypoint recognition using random ferns. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010)
11. Ramachandran, G., Kropatsch, W.: Using aspect graphs for view synthesis. In: Proceedings of Computer Vision Winter Workshop, CVWW (2012)
12. Ravela, S., Draper, B., Lim, J., Weiss, R.: Adaptive tracking and model registration across distinct aspects. In: International Conference on Intelligent Robots and Systems (1995)
13. Sinha, S.N., Frahm, J.M., Pollefeys, M., Genc, Y.: Gpu-based video feature tracking and matching. In: EDGE, Workshop on Edge Computing Using New Commodity Architectures (2006)
14. Torres, F., Kropatsch, W.G., Artner, N.M.: Predict pose and position of rigid objects in video sequences. In: Proceedings of International Conference on Systems, Signals and Image Processing, IWSSIP (2012)
15. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., Schmalstieg, D.: Pose tracking from natural features on mobile phones. In: International Symposium on Mixed and Augmented Reality, Cambridge, UK (2008)
16. Wagner, D., Schmalstieg, D., Bischof, H.: Multiple target detection and tracking with guaranteed framerates on mobile phones. In: Proceedings of Int. Symposium on Mixed and Augmented Reality (2009)