Generating pose hypotheses for 3D tracking: a bottom-up approach

Martim Brandão and Alexandre Bernardino
Instituto de Sistemas e Robótica, Instituto Superior Técnico
martimbrandao@gmail.com, alex@isr.ist.utl.pt

Abstract

Tracking an object's 3D pose from a color image can been accomplished with particle filters if its color and shape properties are known a priori. Unfortunately, initialization in particle filters is often manual or random, thus rendering the tracking recovery process slow or no longer autonomous. A method that uses existing object information to better decide on where to automatically start or recover the tracking process is proposed. Each 3D pose of an object is observed as a 2D shape and so training is made to infer pose from image information. The object is first segmented through color, then shape description is made using geometric moments and finally a learning stage maps 2D shapes to 3D poses with an associated likelihood measure.

1. Introduction

A class of methods used for 3D model-based object tracking is based on particle filters [1]. These represent the distribution of an object's 3D pose as a set of weighted hypotheses (particles). Particle filters, by maintaining several hypotheses over time, have increased robustness [2]. Hypotheses are tested by projecting the object model in the image and comparing actual image pixel information, meaning a top-down approach is used. In every frame a better fit for the object position is looked for, according to appropriate motion and noise distributions. Initialization (or re-initialization) is, though, a problem in particle (and basically all) filters. When no a-priori information of an object's location is known, particles are scattered randomly in space – making the method difficult or slow to converge whether you use few or many particles respectively. In an attempt to address this problem, the method proposed in this paper focuses on quickly and intelligently choosing where to place particles and start or restart looking for the target, based on image information, in a bottom-up manner. With the integration of both approaches, top-down's precision is kept, while both initialization speed and robustness to object reappearance is obtained from the bottom-up layer.

2. Segmentation

The first step in our method consists in segmenting the objects by color. We do so through color segmentation on the HSV color-space of the image, which was chosen in order to better achieve luminosity invariance. A color histogram of the object is known and so a Histogram Backprojection algorithm is applied, building a map representing the likelihood of each pixel belonging to the object. A scale-space of this map is then created to better deal with the simultaneous presence of both small and large objects.

The segmentation algorithm used in each scale was obtained through a flood-fill method using local maxima of the map as seeds. Instead of the standard stop criteria, Sauvola's binarization formula [3] was used to adapt the boundary detection threshold to the region's standard deviation.

3. Bottom-up 3D pose estimation

In order to estimate 3D pose from a segmented region, we compare its shape with trained ones. Since we use the perspective camera model, this training stage can be made independent of object position in the image. In run-time, if objects are not centered, we simulate a camera rotation to the centroid of the object. Training will therefore be made with the object centered in the image and a database is built that matches 2D shape to 3D orientation. The measured orientation can then be rectified using the equations for projecting rotated points in a pan-tilt camera [4]:

$$p = \arctan(x), \quad t = \arctan(y\cos(p)) \qquad (1)$$

where x,y are the region's normalized centroid coordinates and p,t are the pan and tilt rectification angles which, when applied after the measured rotation, give us the true orientation of the object.

Because perspective projection deforms the object as it moves away from the image center, a change of

coordinates is made to center the region before computing its shape features. A homogeneous transformation with *p* and *t* as the rotation angles will produce such result, thus rendering orientation estimation independent of position.

Depth (Z coordinate) is then computed from the relation of the region's area and trained area and depth:

$$Z = Z' \, sqrt(Area' / Area) \qquad (2)$$

Finally, X and Y are computed from the geometric center of the region, by assuming that the 3D geometric center of the object projects on the 2D center of the region given by (x,y)=(X/Z,Y/Z). This introduces some errors but allows us to generate good hypotheses that will be refined in the subsequent particle filtering stage.

To describe shape we use geometric moments, which hold point distribution information. Invariance to position and scale can easily be accomplished by using relative positions to the region's centroid and normalization to the area. A normalized distance function was defined assuming a normal distribution:

$$d = \sum_{p,q} \frac{(\tilde{n}_{pq} - {}^{i}n_{pq})^2}{\mathrm{var}(n_{pq})} \qquad (3)$$

$\tilde{n}_{pq}$ being the observed moment, ${}^{i}n_{pq}$ the moment of the i^{th} hypothesis and var(n_{pq}) the variance of the trained moment of order pq. The most likely pose estimate of a segmented object will then be the one with minimum distance to the measured moments.

3.1. Particle generation

A likelihood function was defined from *d* as *L*=exp(-*d*/2). From this likelihood function, a cumulative distribution was computed, from where N particles can be generated according to their likelihood by sampling the function in a uniform way.

4. Results and Conclusion

The method was tested on perfect segmentations to evaluate its localization error alone. These were generated by projecting the object in random poses, covering untrained orientations. Given the top-down integration context, the error that tracking will be subject to is related to the least error particle. A quaternion representation was used to compute a single error value between the real and estimated orientations.

The average of the absolute angle error was then measured for different numbers of particles generated and it was concluded that with any of the resolutions 20º, 15º and 10º, the error for N=100 has already reached a value equal or close to the resolution.

Table 1. Average absolute error of best particle

Res. (º)	Num. poses	X (cm)	Y (cm)	Z (cm)	Angle (º)
20	3240	0.27	0.18	0.97	21.6
15	7488	0.27	0.18	0.94	15.85
10	24624	0.29	0.18	0.9	11.95

The whole method was tested on real images as well, for complex objects, demonstrating the credibility of pose estimates with highest likelihood (see Fig. 1).

Figure 1. A learned object with the shape of a number "5" (top-left). The remaining images show the highest likelihood poses identified.

Note that in the first example no likely hypothesis exists on the number "6" since its shape is too different. We can from both table and images conclude the proposed method generates credible pose hypotheses with a tolerable error (less than 1cm on position and equal to resolution on orientation).

Acknowledgements

Work partially funded by FCT (ISR/IST plurianual funding) through the PIDDAC Program funds.

References

[1] M. Taiana, J. Nascimento, J. Gaspar and A. Bernardino. "Sample-Based 3D Tracking of Colored Objects: A Flexible Architecture", *BMVC2008*, Leeds, UK, 2008.

[2] V. Lepetit and P. Fua. "Monocular model-based 3d tracking of rigid objects: A survey", *Foundations and Trends in Computer Graphics and Vision*, 1(1), 2005, pp1-89.

[3] J. Sauvola and M. Pietaksinen. "Adaptive document image binarization", *Pattern Recognition*, 33, 2000.

[4] B. Tworek, A. Bernardino, and J. Santos-Victor, "Visual self-calibration of pan-tilt kinematic structures", *Proc. ROBOTICA 2008*, April, 2008.