Spiral Particle Distribution for Template based Tracking

VLADIMIR PLEŠTINA[1], VLADAN PAPIĆ[2]
[1]Faculty of Science, University of Split, Split, CROATIA
[2]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Split, CROATIA
[1]vladimir.plestina@pmfst.hr, [2]vladan.papic@fesb.hr

Abstract: - This paper presents a new approach for tracking template-based objects based on spiral particle distribution algorithm. Proposed algorithm uses points on Archimedean spiral as possible location of object in next frame. Before applying algorithm, the system is provided with off-line learning of training data. After that, start point is initialized and algorithm for tracking is applied. Tracking starts from initialization location and searches for the best matching point on spiral as a new starting point in the next frame. In this work our algorithm is explained and compared with basic particle filter tracking algorithm. Experiment is demonstrated with real data on Croatian popular amateur game.

Key-Words: - Particle filter, template-based tracking, spiral particle distribution

1 Introduction

Computer vision object tracking has application in many commercial, educational and security systems [1]. Tracking an object is a difficult task and depends on many parameters. In literature is presented a lot of different ways to track an object [2], [3], [4], [23], [24] and to solve occlusion problems [5]. Some authors use object shape [6], color [7] texture [8] or their combination as features for tracking. Some authors use statistical methods such as particle filter [9], [10], [11]. The knowledge about features of tracking objects makes it possible to create faster and more accurate system. Similar like human brain processing: if we expect to see a person, we can exclude all other objects. Based on this presumption we have decided to include specific knowledge [8], [12] in our tracking system. Also, in order to improve its performance, our algorithm uses training data.

Our motivation arises in the challenge of compare existing and find a new, faster method that tracks people in water such as sea or pool. Method would be very useful for tracking players and analyses actions in water sports to improve the team strategies. This algorithm can be used for beach surveillance and it could be implemented in applications used as a support for lifeguards during summer season. To accomplish these tasks, this system should be able to automatically detect persons based on templates data and track it. Tracking system is modular [13] so it is possible to use different data as templates.

Because the particle filter tracker is an algorithm based on stochastic tracking [14], [15] it produces different results for the same video sequence on each execution. This inconsistence provoked us to propose a new approach. Proposed spiral particle distribution algorithm is based on two assumptions. First, object location in next frame is not far away from starting point. Second, based on spiral function; exact location of every particle is always known. As the function is mathematically described, the same result is obtained in every interaction. Also, it is possible to regulate angle of spiral or number of particles to speed up tracking.

Additional problem is water reflection on different weather conditions. This was solved by using adaptive HSI color model for water segmentation. This paper presents a new approach for tracking objects based on visual features such as color, shape and texture that templates consist. The idea was to create template independent tracking system so a different kind of data can be used as templates.

The paper is organized as follows: Used methods are explained in section 2. Experimental results and discussion are presented in section 3. Conclusion and future work are described in section 4.

2 Methods

2.1 Particle Filter

Particle filter [7] is also known as condensation algorithm and it is popular method in Computer vision tracking. It was developed to track object

where posterior density $p(X_t|Z_t)$ and the observation density $p(Z_t|X_t)$ are usually non-gaussian. Tracked object values are described with state vector X_t and observation data from *1* to *t* is described with vector Z_t. This vector contain all observation data $\{z_1,\ldots,z_t\}$. Particle filter approximate probability distribution based on set of samples. Each sample has one element that represents the hypothetical state *s* of an object and corresponding weight *w*. Sum of all weights is 1. We can present this as:

$$S = \left\{ (s^{(n)}, w^{(n)}) \,||\, n = 1...N \right\} \tag{1.1}$$

and the sum of weights

$$\sum_{n=1}^{N} w^{(n)} = 1 \tag{1.2}$$

Where N is number of particles. Particles are randomly distributed on initial location and each element of the set is weighted based on the observation model. New weights are set for each sample

$$w^{(n)} = p(z_t \mid X_t = s_t^{(n)}) \tag{1.3}$$

Mean state of the object is estimated at each step of the time

$$mean\left[S\right] \sum_{n=1}^{N} w^{(n)} s^{(n)} \tag{1.4}$$

After each interaction, weights are normalized, and particle distribution start again in *t+1* frame based on frame in time *t*.

2.2 SPAD Algorithm

The Archimedean spiral is the locus of points corresponding to the locations over time of a point moving away from a fixed point with a constant speed along a line which rotates with constant angular velocity. In polar coordinates it can be described by the equation

$$\rho = k\varphi, k = \frac{\upsilon}{\omega}, (k > 0, -\infty < \varphi) \tag{1.5}$$

where *k* is a real number. If we take discrete number of points from the spiral curve, we can assume them as points of proposed spiral algorithm. Each point has its own coordinates on the image and become possible location in the next frame. Points are distributed around last known good location and the assumption is that in the next frame tracked object is not too far from the present location. Starting from the initialization coordinate *o* and moving along the spiral, a discrete number of points *N* is sampled. Set of points *S* has *N* coordinates with weights from set *W*. Each point represents one coordinate with one weight. Every discrete point from the spiral is used as a base for template comparison. Based on the template data, weight *w* is allocated to each coordinate *a*. The coordinate with the biggest weight is the most significant and becomes the starting point *o* in the next frame *f+1*.

$$S = \left\{ a_1, a_2, ...a_N \right\} \tag{1.6}$$

$$W = \left\{ w_1, w_2, ...w_N \right\} \tag{1.7}$$

Spiral particle distribution algorithm:

1) Define threshold value *Thr*,
2) Define number of particles *N*
3) Define number of frames *F*
4) Go to the first frame
5) Start location initialization (*o*)
6) Weight normalization from w_1 to w_N
7) Spiral distribution starting from location *o*
8) Spiral sampling from *1* to *N*
9) For every a_N check likelihood with training data
10) Allocate weight w_N for each a_N based on likelihood
11) Determine the biggest weight
12) Setting a_N with the best weight as new *o* for the next frame
13) Go to 5 until end of frames

It is very important to choose right number of particles *N* so that tracking is accurate and fast enough. The number of particles is the number of possible object locations in the next frame. Location initialization is the starting point of the spiral. Each particle initially has the same weight and each particle gets new weight based on template likelihood. New weight is compared with initially

defined threshold. Particle with the biggest weight is the most likely new start position in the next frame. Process is repeated until the last frame.

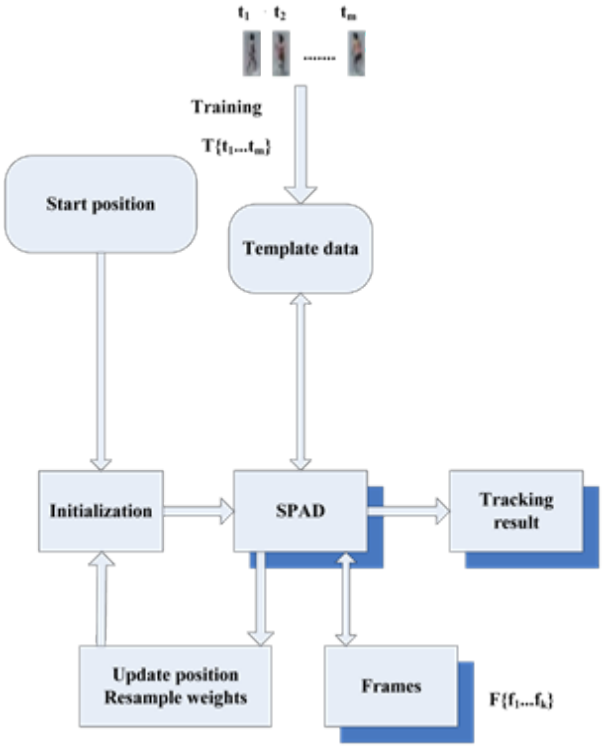


Fig.1: SPAD Algorithm

2.2.1 Training Process and Template Matching

SPAD algorithm uses template matching technique for tracking [16]. A template data is previously prepared set of data *T* based on player characteristics. Training is supervised machine learning process using player images. After training is done set of player characteristics are extracted and it is used as template data in SPAD algorithm and observation model in particle filter tracking. In the template matching process template data set is compared with data in state vector. This process is repeated for all discrete points of the spiral from set *S*. Mean state of high weight points is estimated position for the start point in next frame.

As a similarity measure for the template matching, the normalized correlation coefficient is used [17], [18], [19].

$$T'(x',y') = T(x',y') - \frac{1}{wh}\sum_{x'',y''} T(x,y) \tag{1.8}$$

$$I'(x+x',y+y') = \\ = I(x+x',y+y') - \frac{1}{wh}\sum_{x'',y''} T'(x+x'',y+y'') \tag{1.9}$$

$$R(x,y) = \frac{\sum_{x',y'} T'(x',y')\cdot I'(x+x',y+y')}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \tag{1.10}$$

In equations (1.8), (1.9), (1.10), *T(x,y)* is the template image data, *I(x,y)* is the searched image data with size *w* and *h* and *R(x,y)* is the result. Based on the result, every coordinate from set *S* gets weight from set *W*.

3 Experiment and Discussion

Proposed algorithm was tested on popular Croatian amateur game called *Picigin*. There are several reasons why this testing was challenging; only one low resolution camera was used; Players played in the sea were people walk by, so there was occlusions with other players and random walkers.

Tracking data obtained by manual tracking of *500 frames* were used as a *ground truth data*. Results of the SPAD tracking algorithm and tracking results of particle filtering [7] that use same template data as observation model were compared with this ground truth data.

First problem was how to segment dynamic texture as water and how to solve problem with reflections. As a solution input RGB color image was transformed into HSI color space [20] and value that represents sea was extracted.

3.1 Training Process

During the training process 30 images of players were taken. They were *15 pixels* wide and *40 pixels* high. Also, non-player images that contains sea and environment around were taken so that system has more positive and negative data about ambient. Some positive training data is presented in Fig.3.

As it can be seen in Fig. 2 and Fig. 3, resolution of images and templates is very low, and it is not possible to distinguish one player from another. This is standard problem on old surveillance cameras. It is also a problem with people that walk around and cause occlusions with players. Players were segmented in different positions and used as training data. Proposed system was trained with player images, but characteristics from each template image were extracted separately.

Sea was removed from the background so that shape and form of a player can be used. Set of data was prepared to compare it with state model of each particle. Result of the training process was matrix that consists all templates optimized based on shape, form and color of the player.

(a) RGB image

(b) HSI adaptive sea segmentation

Fig.2: Sea Segmentation

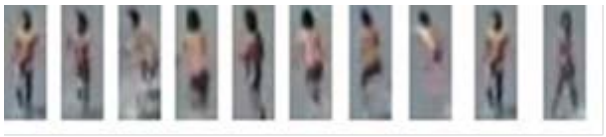


Fig.3: Subset of Positive Training Data

3.2 Tracking

Ground truth data obtained from manually tracked player was compared with SPAD algorithm tracker using template data and particle filter that use same observation data. One action scene was tracked where player start to run and disappear in water.

After training process was finished, SPAD algorithm and particle filter tracker were started on Dell optiplex computer with 3 GHz CPU and 4 GB of RAM. Tracking software was developed in Matlab. In Fig. 5 different path results can be seen. Figure 5 a) shows real path of player. It is referent point and true data so the SPAD algorithm, particle filter tracking and manually tracked data can be compared. Image b) is outlined a path of SPAD algorithm and on image c) is outlined a path of particle filter tracking. As particle filter is a stochastic way of tracking it is important to notice that in every new interaction, different results were obtained. Results were compared using distance between coordinates of tracking points in every frame for ground zero data - SPAD algorithm and ground zero data - particle filter tracking. Average distance between ground zero data and SPAD algorithm was 5,898 pixels. Average distance between ground zero data and particle filter was 10,479 pixels.

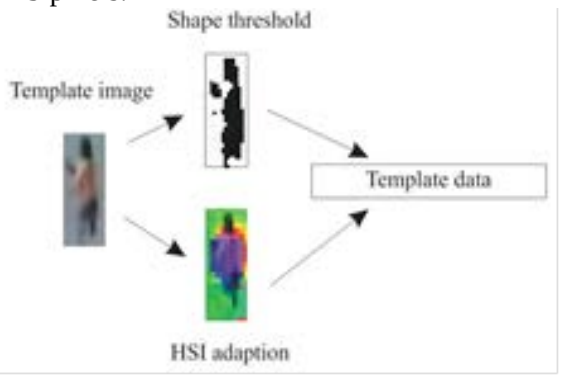


Fig.4: Template data

From this data, it is obvious that SPAD algorithm is more accurate. SPAD algorithm is a little bit slower than particle filter. For 500 frames, 167,56 seconds was needed for tracking using SPAD algorithm while particle filter algorithm was done in 136,72 seconds. Tracking results are given in Table 1. Fig. 6 represents graphic view of difference between SPAD tracking and to ground truth data during tracking sequence. On x-axis is frame number and on y-axis is difference in number of pixels. On Fig. 7 graphic view of difference between Particle filter tracking and ground truth data is presented. Better results are closer to 0. Result shows that SPAD algorithm provided better results during all frames.

Table 1: Tracking

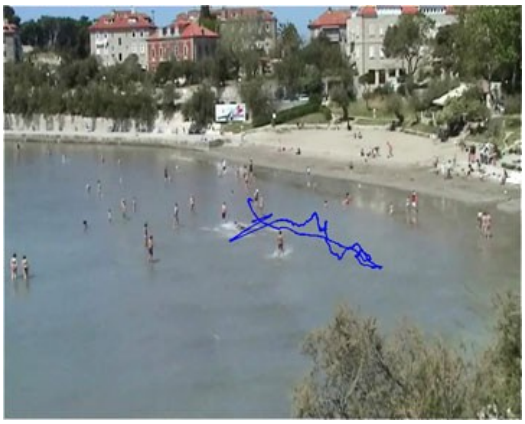
Method	Time	Avg. distance	Lost frames
SPAD algorithm	167,56 s	5,898 pixels	16
Particle filter	136,72 s	10,479 pixels	23

SPAD algorithm lost player tracking in 16 frames and particle filter lost it in 23 frames (Table 1). It is important to notice that SPAD algorithm has lost person and then find it again in the next frame. That means that person was incorrectly located 16 times during the whole period of tracking. It is considered that tracking has lost a person if distance between real data and tracking method data is more than 20 pixels. During comparison, even when a person was lost, SPAD algorithm has never been more than 40

pixels away from the real data. Particle filter has lost person 23 times, but in some cases, it was more than 40 pixels away. Particle filter in some interactions even had bigger error and SPAD algorithm always produced the same results. We did not have any significant problems with occlusions, but problems are possible because of similarity between people.

(a) Manually tracked path outline

(b) Outlined path of SPAD algorithm tracking



(c) Outlined path of particle filter tracking

Fig.5: Experimental results of tracking

4 Conclusions and Future Work

In this work we present a novel algorithm for tracking in special situations. Basic idea is to use spiral distribution with different angles or number of discrete particles. Water segmentation in different weather conditions is also interesting problem that can be solved by analyzing different Color models. Template matching and correlation between segmented data in presented case were used for comparison.

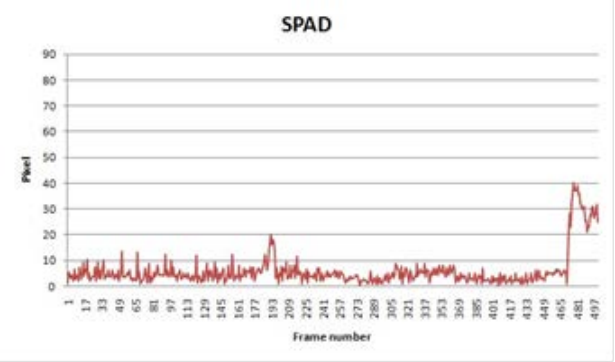


Fig.6: Object position difference SPAD from GTD

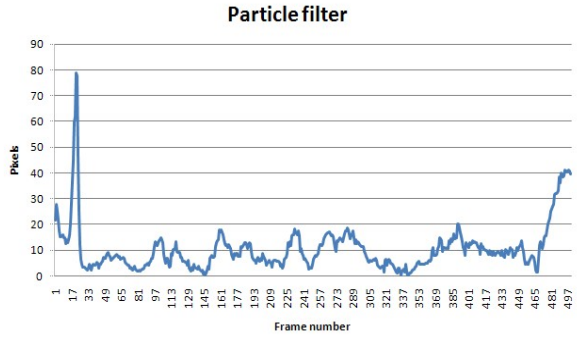


Fig.7: Object position difference PF from GTD

Experimental results showed that the proposed algorithm is significantly more accurate than the standard particle filtering approach using the same kind of template data on our specific problem. Computation was slightly slower but it is important to notice that all code of SPAD algorithm is in Matlab script and some parts of particle filter tracking was written in C++, so we expect that translation of SPAD algorithm into C++ code should significantly improve its performance.

Implementation of principal component analysis (PCA) for data optimization or some other template comparison technique instead of the used one should be quite simple. In the future work our idea is to assemble template database with people in water sports and apply SPAD algorithm for tracking. Also, we intend to speed up algorithm with code optimization [21] and use different kind of template matching [22].

References:

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," Acm Computing Surveys (CSUR), vol. 38, no. 4, 2006.

[2] J. Pers and S. Kovacic, "Computer vision system for tracking players in sports games," IWISPA 2000. Proceedings of the First International Workshop on Image and Signal Processing and Analysis. in conjunction with 22nd International Conference on Information Technology Interfaces. (IEEE Cat. No.00EX437), vol. 1, pp. 177–182, 2000.

[3] J Liu, X Tong, W Li, T Wang, Y Zhang, and H Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," Pattern Recognition Letters, vol. 30, no. 2, pp. 103–113, Jan. 2009.

[4] Junqiu Wang and Yasushi Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking.," IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 17, no. 2, pp. 235–40, Feb. 2008.

[5] C.J. Needham and R.D. Boyle, "Tracking multiple sports players through occlusion, congestion and scale," British Machine Vision Conference, 2001.

[6] Jacky C. Yuk, Kwan-yee Wong, Ronald Y. Chung, F. L. Chin, and K. Chow, "Real-time Multiple Head Shape Detection and Tracking System with Decentralized Trackers," Sixth International Conference on Intelligent Systems Design and Applications, pp. 384–389, Oct. 2006.

[7] K. Esther, E. Koller-meier, and L. V. Gool, "A Color-based Particle Filter," Technology, pp. 53–60, 2002.

[8] W. Lu, K. Okuma, and J. Little, "Tracking and recognizing actions of multiple hockey players using the boosted particle filter," Image and Vision Computing, vol. 27, no. 1-2, pp. 189–205, Jan. 2009.

[9] S. Saha, N. Bambha, and S. Bhattacharyya, "Design and implementation of embedded computer vision systems based on particle filters," Computer Vision and Image Understanding, vol. 114, no. 11, pp. 1203–1214, Nov. 2010.

[10] C. Orrite-Urunuela and JE. Herrero, "An efficient particle filter for color-based tracking in complex scenes," Advanced Video and, pp. 1–6, 2008.

[11] Chong Chen and Dan Schonfeld, "A particle filtering framework for joint video tracking and pose estimation.," IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 19, no. 6, pp. 1625–34, June 2010.

[12] E. Monier, P. Wilhelm, and U. Ruckert, "Template matching based tracking of players in indoor team sports," 2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), pp. 1–6, Aug. 2009.

[13] V. Pleština, H. Dujmic, and V. Papic, "Proposal of a modular system for tracking indoor and outdoor sports," Proceedings of the 9th WSEAS international conference on Simulation, modelling and optimization, pp. 132–137, 2009.

[14] K. Smith, D. Gatica-Perez, and J. Odobez, "Using Particles to Track Varying Numbers of Interacting People," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 00, no. c, pp. 962–969, 2005.

[15] A. Dearden, Y. Demiris, and O. Grau, "Tracking football player movement from a single moving camera using particle filters," 3rd European Conference on Visual Media Production (CVMP 2006). Part of the 2nd Multimedia Conference 2006, pp. 29–37, 2006.

[16] M. Perˇse, M. Kristan, J. Perš, and S. Kovačič, "A template-based multi-player action recognition of the basketball game," Proceedings of the ECCV Workshop on Computer Vision Based Analysis in Sport Environments, pp. 71–82, 2006.

[17] J.P. Lewis, "Fast normalized cross-correlation," Vision Interface, vol. 10, no. 1, pp. 120–123, 1995.

[18] K. Briechle and U.D. Hanebeck, "Template matching using fast normalized cross correlation," Proceedings of SPIE, vol. 4387, pp. 95– 102, 2001.

[19] D. Tsai, "Fast normalized cross correlation for defect detection," Pattern Recognition Letters, vol. 24, no. 15, pp. 2625–2631, Nov. 2003.

[20] Calin Rotaru, Thorsten Graf, and Jianwei Zhang, "Color image segmentation in HSI space for automotive applications," Journal of Real-Time Image Processing, vol. 3, no. 4, pp. 311–322, Mar. 2008.

[21] A. Sankaranarayanan, A. Srivastava, and R. Chellappa, "Algorithmic and architectural optimizations for computationally efficient particle filtering.," IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 17, no. 5, pp. 737–48, May 2008.

[22] S. Wei and S. Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update.," IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, vol. 17, no. 11, pp. 2227–35, Nov. 2008.

[23] W. Askar, O. Elmowafy, A. Youssif, G. Elnashar, "Real-time UAV Target Tracking System Based on Optical Flow and Particle Filter Integration", WSEAS Transactions on Signal Processing, ISSN / E-ISSN: 1790-5052 / 2224-3488, Volume 13, Art. #19, pp. 172-181, 2017.

[24] Z. Chen, Y. Bo, Y. Qu, X. Ling, X. Tao, Y. Liu "Dynamic Population Adaptive Particle Swarm Optimized Particle Filter for Integrated Navigation", WSEAS Transactions on Signal Processing, ISSN / E-ISSN: 1790-5052 / 2224-3488, Volume 11, Art. #28, pp. 235-244, 2015