21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Single Camera Multi-person Tracking Based on Crowd Simulation

Zhixing Jin, Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside
jinz@cs.ucr.edu, bhanu@cris.ucr.edu

Abstract

Tracking individuals in video sequences, especially in crowded scenes, is still a challenging research topic in the area of pattern recognition and computer vision. However, current single camera tracking approaches are mostly based on visual features only. The novelty of the approach proposed in this paper is the integration of evidences from a crowd simulation algorithm into a pure vision based method. Based on a state-of-the-art *tracking-by-detection method, the integration is achieved by evaluating particle weights with additional prediction of individual positions, which is obtained from the crowd simulation algorithm. Our experimental results indicate that, by integrating simulation, the multi-person tracking performance such as MOTP and MOTA can be increased by an average about* 2% *and* 5%*, which provides significant evidence for the effectiveness of our approach.*

1. Introduction

In the area of computer vision, tracking individuals is always a popular research topic. It is one of the most important aspects of computer vision, and has various applications in the areas such as surveillance, monitoring, and security. But it is also a challenging problem that is still far away from a perfect solution. In this paper, we are focused on the situation to track individuals in a crowded scene. This problem is challenging because of the large number of occlusions from other individuals in a crowded scene, which makes the traditional visual feature not effective. Therefore, it may be useful to take the advantages of some approaches from a different perspective of computer vision. This is our motivation to integrate the crowd simulation into tracking algorithm.

One of the *state-of-the-art* categories in tracking area is the combination of tracking and detection, for which it is named as "tracking-by-detection" [1, 3]. The general idea of this type of approaches is to initialize and modify the tracking results according to the output from detectors such as HOG (Histogram of Oriented Gradient). Different from background based trackers, they are more robust since they rely less on the background information. However, it is not easy to directly applying the outcomes from detectors because their outputs are sparse and unreliable. So several algorithms are proposed to solve the data association between trackers and detectors [1, 6]. To better utilize the information obtained from the detector, the tracking-by-detection approach used in our work not only addresses the data association between the trackers and the final detection results, but also tries to utilize the intermediate output from the detector by integrating a detection confidence map into the evaluation process of particle weights [3].

In computer graphics, the purpose of crowd simulation is mainly to mimic the walking routes of a group of people in a natural manner, with collision avoidance. A good crowd simulation model is able to provide valuable information in predicting the positions of each agent in this crowd, given the information about their current status (*e.g.* positions, velocities). Those crowd simulation models include but not limited to: social force [5], RVO2 (Reciprocal Velocity Obstacles) library [8] and continuum dynamics [7]. Of course, current tracking program cannot offer accurate information about the exact position and velocity of an individual for crowd simulation methods, but with reliable estimation, we can still get the approximation of possible future locations, which is quite helpful in the tracking progress, especially in a crowded scene. The dataset that our experiment uses is published by UCSD, and the crowd simulation algorithm adopted is the RVO2 library with our novel estimating method for the preferred velocity.

In the rest part of this paper, we provide a detailed description on our technical approach in Section 2 and show the experimental results in Section 3. Finally, Section 4 provides the conclusions.

978-4-9906441-1-6

2. Technical Approach

The idea of our approach is to integrate the prediction of the positions of a crowd into the tracking-by-detection algorithm, by modifying the evaluation of particle weights. Therefore, the technical description of our approach in this section will be divided into three stages: the pure tracking-by-detection method, the crowd simulation algorithm, and the observation model deriving from them. The diagram of the system is illustrated in Figure 1.

2.1 Tracking-by-detection

As mentioned above, the fundamental tracking method applied in our work is the *state-of-the-art* tracking-by-detection approach, which is also a combination of particle filter, boosted classifier, and detector. But in our work, we did not implement exactly the same algorithm as the original one since we are more focused on the potential positive influence brought by the integration of crowd simulation. In the following, we will describe the general idea of this method as well as all the differences in our version. For readers who are interested in exploring all the details about the original version, please refer to the paper [3].

The first component is a *bootstrap filter*. The state $\mathbf{x}$ consists of the position (x, y) and velocity (u, v). At each time step t, the resampling of particles is carried with an initial weight $1/N$ (N is the number of particles), so the weight of each particle w_t^i is proportional to $p(o_t|\mathbf{x}_t^i)$ computed by the observation model (described in Section 2.3). The motion model used is the constant velocity motion model. In the original work [3], an additional procedure is used to deal with fast camera motion, but in our algorithm this is removed because the camera is fixed in our dataset.

A second part is the detector and the data association between the trackers and detections. For each iteration, the algorithm chooses the largest matching score and then determines an association between the pair if none of the tracker or detection is associated before. This processing is repeated until none of the matching scores is larger than a predefined threshold τ. The equation to calculate matching score is

$$S(tr, d) = g(tr, d)\left(c_{tr}(d) + \alpha \sum_{p \in tr}^{N} p_{\mathcal{N}}(d - p)\right) \tag{1}$$

where tr and d stand for tracker and detection respectively, and $p_{\mathcal{N}}$ is zero mean normal distribution.

Here $g(tr, d)$ is the gating function to limit the possible locations of detections according to the current velocity (*cf.* [3]), $c_{tr}(d)$ is the evaluation result from the classifier. In the original version, both ISM and HOG detectors are tested. However, for each dataset, only one of them is applied. Therefore, in our work, only the HOG detector is chosen because the current ISM model is trained on the side view of persons.

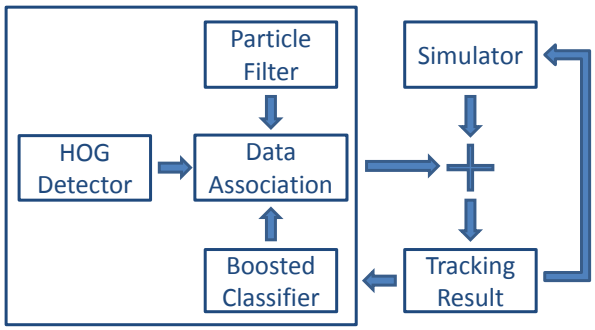

Figure 1. The system diagram. The tracking result is obtained from the integration of tracking-by-detection and simulator, and used for the next frame.

The last component is a boosted classifier [4]. For each individual, a series of weak learners are trained against all nearby individuals and background, and are selected later using AdaBoost. The classifier is updated only when the tracker is associated with one detection which is not overlapped with other detections.

In the original version, the tracker is automatically initialized by non-associated detection results, but our approach initializes trackers by annotations since simulation needs more accurate position information.

2.2 Crowd simulation

The crowd simulation algorithm integrated in the current approach is the RVO2 library [8]. This is a simple but rather efficient simulation model. Besides those settings such as the radius, maximum speed of each agent and the information about the obstacles like walls, it only requires the current position and preferred velocity (including the direction) for each agent to simulate one step. This crowd simulation approach adopts the strategy called Optimal Reciprocal Collision Avoidance (ORCA) based on the computation of the velocity obstacles defined as follows

$$VO^{\tau}_{A|B} = \{v|\exists t \in [0, \tau] : v \cdot t \in D(p_B - p_A, r_A + r_B)\} \tag{2}$$

Here, $D(p, r)$ defines a circle centered at position p with radius r. So $VO^{\tau}_{A|B}$ is the set of all relative velocities of A with respect to B that will cause a collision in the next period of time with a length of τ. At each time step, for each agent in the environment, all the ve-

locity obstacles from other agents are computed. After that, the global optimal solution is then solved by linear programming, which is quite efficient (several times faster than real-time frame rates depending on the total number of agents).

However, there remains a problem. The simulation needs the information about preferred velocity of each agent, which is typically a vector with the direction towards the goal position and the magnitude equal to a predefined speed. But in our approach, the goal position of each agent is impossible to learn before it is finally reached, and even the speed is not able to be accurately defined. Therefore, we estimate the preferred velocity according to the historical information. This estimation procedure is based on the importance sampling of the derivative of recent velocities (accelerations). If the most recent m accelerations are recorded at time t for agent k: $A_k = \{a_{k,t-m}, a_{k,t-m+1}, \ldots, a_{k,t-1}\}$, then the weight of each acceleration is defined as

$$w_{a_{k,t-i}} = \frac{m-i+1}{\sum_{j=1}^{m} j}(i = 1, 2, \ldots, m) \quad (3)$$

This gives a larger weight to the more recent acceleration. Then, a set of l accelerations is generated using an importance sampling method $A'_k = \{a'_{k,1}, a'_{k,2}, \ldots, a'_{k,l}\}$, and the corresponding set E of l velocity estimations can be obtained with the following equation

$$e_{k,i} = v_k + a'_{k,i} + \varepsilon_{a,k} \quad (4)$$

Where e is a vector denoting the velocity estimation and v_k is the current velocity vector. $\varepsilon_{a,k}$ is a zero mean normal distribution, with its variance $\sigma_{a,k} \propto \max_{i,j}(||a_{k,t-i} - a_{k,t-j}||)$.

Finally, to reduce the computational complexity, not all the possible combinations are calculated. Instead, only l of them are taken into account: $C_i = \{e_{1,i}, e_{2,i}, \ldots, e_{K,i}\}, i = 1 \ldots l$, and l possible locations are simulated for each agent in this case.

2.3 Observation model

The weight of each particle $w_{tr,p}$ at each time step is also estimated in a similar manner to the original version [3]. However, besides the three terms that already exist, the particle weights are also influenced by the prediction of possible locations for each tracker. So the modified observation model is expressed as

$$w_{tr,p} = \beta\mathcal{I}(tr)p_{\mathcal{N}}(p-d^*)+\gamma d_c(p)p_0(tr)+\eta c_{tr}(p)+\delta s(p) \quad (5)$$

In the above equation, the first three terms are respectively related to the detection result, the detection confidence map, and the classifier evaluation result, which is the same terms from the original observation model. The fourth term is then the additional evaluation from the simulation result, named as "Simulation term". Its definition is as below

$$s(p) = \sum_{q \in Q_k} p_{\mathcal{N}}(p - Q_k) \quad (6)$$

Here, Q_k is the set of possible locations of agent k which is related to the current tracker. This term evaluates the possibility of the position of a particle.

3. Experimental Results

In this section, the experimental settings and results are reported. The crowd dataset tested by our approach is the UCSD crowd dataset. There are 189 individuals that have already been annotated for their positions and velocities. However, in our experiment, we use the overlapping area of two patches as the evaluation criteria, so we annotated the patches containing each individual from the dataset by ourselves. The annotation is done every 10 frames, with interpolation in between.

The parameter setting for the tracking-by-detection algorithm follows the original paper (*e.g.* $\beta : \gamma : \eta = 20 : 2 : 1$) [3]. For the parameter values that are not explicitly given in the paper, we determine them experimentally. The influence brought by different δ values is also investigated. The performances reported below are based on the optimal δ selection.

For the RVO2 library, there are 10 parameters for each agent. Among them, three parameters (position, velocity, and preferred velocity) are calculated during the runtime. The rest seven parameters are optimally assigned by training also on the UCSD crowd dataset using a genetic algorithm, in our prior work.

The results from the pure tracking-by-detection method and our integrated tracking-by-simulation approach are compared. Figure 2 illustrates the results on several frames in the video sequences from these two approaches, as well as the annotated ground truth. As a qualitative analysis, we can see that our tracking-by-simulation approach significantly outperforms the original tracking-by-detection method.

To analyze the performance quantitatively, the evaluation metrics CLEAR MOT [2] is used. The MOTP (multiple object tracking precision) and MOTA (multiple object tracking accuracy) values for the tracking-by-simulation are 74.47% and 56.9% respectively, while for the tracking-by-detection, the performances are evaluated as 72.68% and 51.24% respectively. From the MOT metrics, we can observe that as the tracking algorithm integrated by simulation, its performance increases by about 2% and 5%. For crowded scene with

Figure 2. Some sample tracking results. The first row is the results generated by the proposed tracking-by-simulation and the second row is from tracking-by-detection. The third row is the ground truth.

Table 1. The influence of parameter δ.

δ/η	0.5	1	2	5
MOTP	71.89%	72.29%	74.47%	71.11%
MOTA	50.79%	53.46%	56.9%	49.73%

higher density, the difference of the MOTA value can raise up to about 8%.

The importance of the simulation component in our approach is investigated by setting the coefficient δ in the observation model to various values. Table 3 shows the performance change at different δ values. It is revealed that when $\delta = 2\eta$, the algorithm is most effective, at least in our experiment.

4. Conclusions

As the scenes of the video sequences become more and more crowded, the traditional tracking approaches purely based on computer vision become not so effective. Therefore, in this paper, we propose a novel tracking method that is integrated by the output information from crowd simulation. This tracking-by-simulation approach is derived from the recent tracking-by-detection approach and one of the most efficient crowd simulation algorithm RVO2. The experimental results show that our tracking-by-simulation method outperforms the tracking-by-detection approach significantly, with an increase of about 2% and 5% in the MOTP and MOTA metrics, on the UCSD crowd dataset. In the future, we will expand the method to a more flexible integration so the simulation is not necessary to be done based on every single individuals.

Acknowledgement: We would like to thank the Statistical Visual Computing Lab at University of California, San Diego to provide the crowd video dataset. This work was supported in part by NSF grant 0905671.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.

[2] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *J. Image Video Process.*, 2008:1:1–1:10, Jan. 2008.

[3] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1820 –1833, sept. 2011.

[4] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260 – 267, june 2006.

[5] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487, 2000.

[6] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2953 – 2960, june 2009.

[7] A. Treuille, S. Cooper, and Z. Popović. Continuum crowds. In *ACM SIGGRAPH*, pages 1160–1168, 2006.

[8] J. van den Berg, S. Guy, M. Lin, and D. Manocha. Reciprocal n-body collision avoidance. In *14th International Symposium on Robotics Research*, Sept. 2009.