Modification of the Multi-target Tracking Algorithm Based on Energy Minimization

A. Gringauz[1], E. Shalnov[2], A. Konushin[3]
[1,2,3]Lomonosov Moscow State University
Department of Computational Mathematics and Cybernetics
[1]grin3s@mail.ru, [2]shalnov.eugen@gmail.com, [3]ktosh@graphics.cs.msu.ru

Abstract

In the current work we consider a multi-target tracking problem. The proposed algorithm is based on energy minimization within a temporal sliding window and is a modification of the approach from [10]. We propose a method to impose physical constraints on the trajectory such as curvature or maximum velocity using the energy. Experimental evaluation of the algorithm shows that we were able to achieve a performance improvement compared to the base method [10].

Keywords: *tracking, MCMC DA, energy minimization.*

1. INTORDUCTION

Multi-target tracking is an important computer vision task. It implies constructing trajectories for all people in a given video fragment. The trajectory contains a unique identifier for every person and his position in all frames of the video. This task is important for many applications, for example: video surveillance, improving pedestrian safety. Despite a significant progress in recent years, humans are still far ahead of existing automatic algorithms in terms of solving this task.

In the current work we consider tracking by detection as one of the most promising approaches to this task.

All methods of multi-target tracking can be divided into three groups.

Methods from the first group determine the position of the object in the current frame based its position in the previous frame. Some examples of this approach are algorithms based on Kalman filtering [5] or particle filtering [6].

Methods from the second group use information from the following frames to estimate the person's position in the current frame [10, 3]. These algorithms use energy minimization within a temporal sliding window. The current frame is somewhere in the sliding window, and people's estimated positions might change when the window moves.

The third group is similar to the second. These methods use energy minimization on the whole video fragment, that is the size of the sliding window is equal to the duration of the video. The possibility of defining a continuous energy function that depends on every person's position in all frames was researched in [1, 7].

Besides this approach the same authors developed a descrete-continuous method [2]. In this approach the energy function is divided into two parts: continuous and descrete. Every trajectory is modeled by a cubic B-spline. The continuous part contains terms that impose constraints on how close the trajectory is to detections, velocity, inter-object occlusions. The descrete part is responsible for associating detections from adjacent frames.

The proposed algorithm is a modification of [10]. Due to the improvements the modified algorithm shows better perfomance than the base method.

2. PROPOSED METHOD

The base algorithm and its modification are described below. It consists of the following steps: applying object detector to every frame of the video, building tracklets, building trajectories, estimating people's positions in intermediate frames.

2.1. Searching For Objects

The objects we want to track in the video are people. So we use a HOG based detector [9] to find all people's heads. Using a head detector instead of a full body detector allows us sometimes to find a person even in case of an occlusion.

2.2. Building Tracklets

Then for every found detection a tracklet is built. A tracklet is an object containing information about a detection and a set of its motion estimates. A tracklet is built based on the information obtained using the "Flock of features" tracking algorithm. It uses only one detection and tracks it for several frames forwards and backwards. The position of the head of the person in the frame where it has been detected and its position found by the visual tracking algorithm are used to build the estimate. It is important that as the time of tracking with this algorithm increases the probability of finding a wrong position for a person also increases. That is the reason why it is used only to determine a position of a person in a small temporal neighborhood of the detection.

2.3. Building Trajectories

Let D denote the set of all detections in the temporal sliding window. H is the hypothesis that shows how D is divided into trajectories, that is: $H = \{T_1, ..., T_J\}$, $T_i = \{d_n^j\}$, d_n^j is the n^{th} detection in the j^{th} trajectory. Using Bayes' theorem:

$$(1) \qquad p(H \mid D) = \frac{p(D \mid H)\,p(H)}{p(D)} \propto p(D \mid H)\,p(H)$$

We must find:

$$(2) \qquad H^* = arg\,max_H\, p(H \mid D)$$

Following [10] let's assume that:

$$(3) \qquad p(H) = J! \prod_{T_j \in H} \left(\frac{|T_j|}{|D|} \right)^{|T_j|} p(c_j)$$

Here $|X|$ is the number of elements in the set X, c_j is the type of the j^{th} trajectory. Like in the base algorithm we propose two types of a trajectory: c_{fp} (a trajectory of false positives) and c_{ped} (a person's trajectory).

Let's consider the factor $p(D \mid H)$. In the base algorithm the trajectory was modeled by a Markov chain (see figure 1).

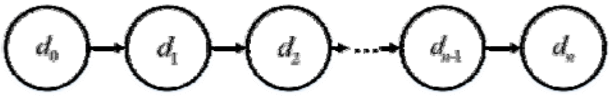

Fig. 1. Trajectory Model

The likelihood of this Markov chain is:

(4) $$p(D|H)=\prod_{T_j\in H}\left[p(d_1^j|c_j)\prod_{d_n^j\in T_j,\ d_1^j}p(d_n^j|d_{n-1}^j,c_j)\right]$$

This representation has its drawbacks:

- It imposes constraints on the behavior of the trajectory in the neighborhood of every point but it doesn't impose global constraints on the whole trajectory.
- The likelihood of the trajectory doesn't depend on other trajectories' behavior.

To overcome some limitations of the base algorithm we propose to use the idea from [2] to use B-splines to model the trajectory. The new trajectory model is shown in figure 2.

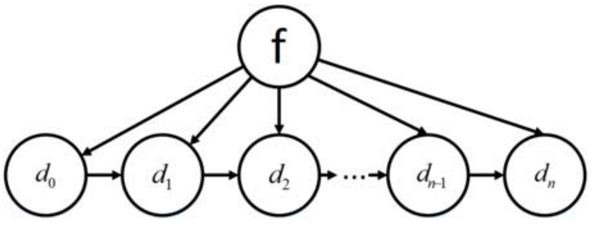

Fig. 2. Modified Trajectory Model

The likelihood of this bayesian network is:

(5) $$p(D|H)=\prod_{T_j\in H}\left[p(f_j|c_j)p(d_0^j|f_j,c_j)\prod_{d_n^j\in T_j,\ d_0}p(d_n^j|d_{n-1}^j,f_j,c_j)\right]$$

Here f_j is the spline, constructed for the trajectory T_j .

Following the work[10], every detection d_n^j is described by its size s_n, position x_n and motion estimate m_n. The likelihood of a single detection is:

(6) $$p(d_1^j|c_j,f_j)=p(s_1)p(x_1)p(m_1|c_j)$$

(7) $$p(d_n^j|d_{n-1}^j,c_j,f_j)=p(s_n|s_{n-1})p(x_n|x_{n-1},c_j)p(m_n|c_j)$$

2.3.1. Detection Size

The size of the first detection of a trajectory can't be estimated based on previous frames, so let's define an a priori distribution:

(8) $$ln(s_1)\sim N(\mu_p,\sigma_p^2)$$

The sizes of the following detections depend on the previous ones:

(9) $$ln\left(\frac{s_i}{s_{i-1}}|c_{ped}\right)\sim N(0,\delta_t\sigma_{s,p}^2)$$

(10) $$ln\left(\frac{s_i}{s_{i-1}}|c_{fp}\right)\sim N(0,\delta_t\sigma_{s,f}^2)$$

Here δ_t is the time difference between frames which the corresponding detections were found on.

2.3.2. Detection Position

We assume that the position of false positives may change only due to the noise.

(11) $$x_i|x_{i-1},c_{fp}\sim N(x_{i-1},2\Sigma_d)$$

Let v_p be a motion estimate, obtained from the tracking results from the previous frame and the next frame. This estimate is considered the most reliable. Consider the distribution $x_i|x_{i-1},c_{ped}$. First let's derive the estimate for the person's position at time i using the motion estimate v_p :

(12) $$x_p^{i-1}=x_{i-1}+\delta_t v_p$$

(13) $$\Sigma_p=\delta_t\Sigma_v$$

Here Σ_v is a parameter modeling an error in the speed v_p . In order to improve the estimate x_p^{i-1} we propose to use the remaining motion estimates. Let $Y_i=\{y_i\}$ be the set of motion estimates for a detection at time i . The more accurate estimate x_y^{i-1} is given by:

(14) $$x_y^{i-1}=x_p^{i-1}+\Delta x_y$$

(15) $$x_p^{y_{i-1}}=x_{i-1}+\delta_t y_{i-1}$$

(16) $$\Delta x_y=\Sigma_p(\Sigma_p+\delta_t\Sigma_{local})^{-1}(x_p^{y_{i-1}}-x_p^{i-1})$$

(17) $$\Sigma_y=(I-\Sigma_p(\Sigma_p+\delta_t\Sigma_{local})^{-1})\Sigma_p$$

The parameter Σ_{local} describes the error of the algorithm used for building the tracklets (see section 2.2), δ_t is the time difference between detections d_i and d_{i-1} .

Although this estimate is more accurate, it is less reliable, because the error of the tracking algorithm used for building tracklets increases. Therefore, we propose to define the distribution for a person's position at time i as a mixture of normal distributions:

$$x_i|x_{i-1},Y_{i-1},c_{ped}\sim$$

(18) $$\frac{\alpha^{\delta_t}}{|Y_{i-1}|}\sum_{y\in Y_{i-1}}N(x_y^{i-1},\Sigma_y+2\Sigma_d)+(1-\alpha^{\delta_t})N(x_p^{i-1},\Sigma_p+2\Sigma_d)$$

Here α is the probability of the visual tracking algorithm to lose a person in a given frame.

In the work [10] it was shown that using motion estimates Y_i of the tracklet d_i , besides the motion estimates Y_{i-1} of the tracklet d_{i-1} , improves tracking performance. Therefore it was proposed to use motion estimates from both associated tracklets.

(19) $$\begin{gathered}p(x_i|x_{i-1},Y_i,Y_{i-1},c_{ped})=\\ \beta p(x_i|x_{i-1},Y_i,c_{ped})+(1-\beta)p(x_i|x_{i-1},Y_{i-1},c_{ped})\end{gathered}$$

(20) $$\beta=\frac{|Y_{i-1}|}{|Y_{i-1}|+|Y_i|+2}$$

2.3.3. Motion Magnitude

The motion estimate m is needed to distinguish a person's trajectory from a trajectory of false positives. Following the work [10], it is modeled by a histogram of 4 bins. The boundaries of the bins represent the movement of $\frac{1}{8},\frac{1}{4},\frac{1}{2}$ of the detection's size. We assume that the histogram corresponds to one of the two multinomial distributions, depending on the type of the track:

(21) $$m_i|c_{ped}\sim Mult(m_{ped})$$

(22) $$m_i|c_{fp}\sim Mult(m_{fp})$$

2.3.4. Spline Distribution

Let's define an a priori distribution for a spline $p(f)$. Let K be a number of parts in the spline. Then it can be defined by a coefficient matrix $C\in\Box^{2K\times 4}$. Number 2 represents the fact that splines are constructed for two dimensions x and y. Supposing a spline consists of polynomials:

(23) $$f_i(t)=a_it^3+b_it^2+c_it+d_i$$

$$C=[a_i,b_i,c_i,d_i],i=\overline{1,2K}$$

Let's denote:

(24) $$A = \max_i |a_i|$$

And assume:

(25) $$f \sim N(A \mid 0, \sigma^2)$$

In the work [2] it was mentioned that the coefficient of the highest degree of the polynomial has the greatest influence on the person's speed. The parameters of the described distribution were estimated using the maximum likelihood method. The training set consisted of PETS-S2-L1[1], PETS-S2-L2, PETS-S2-L3 and TownCentre[2] datasets, containing ground truth. Modeling parameter a_i with a discrete distribution (see figure 3) showed that a normal distribution is a reasonable approximation of the distribution of this parameter obtained from the real data.

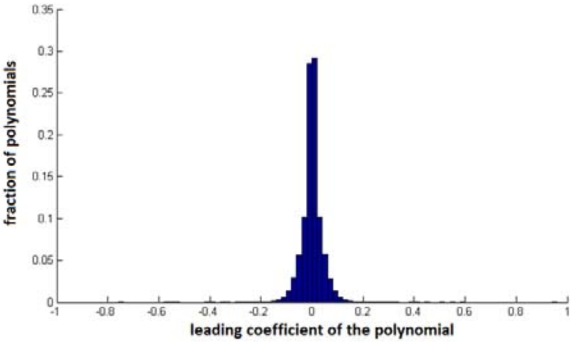

Fig. 3. Statistic for A

2.3.5. Associating Algorithm

As it was described in the beginning of section 2.3, at every moment the algorithm searches for an optimal splitting of tracklets into trajectories within a temporal sliding window. There is no efficient algorithm to find a trajectory hypothesis H^* (see eq. (2)), which is a global maximum point of the likelihood function. Therefore we use an approximate inference method MCMC DA [8]. The Metropolis–Hastings algorithm is used to obtain a sequence of samples from the distribution.

We use an initial hypothesis with high likelihood as an initialization for the algorithm in order to accelerate convergence. This hypothesis can be obtained from the tracking results from the sliding window corresponding to the previous time step. New tracklets are added using a greedy algorithm, trackets that went beyond the sliding window are removed from the hypothesis.

In order to generate new hypotheses, three types of trajectory transforms are used: "swap", "switch", "change type".

When a "swap" move occurs, two randomly chosen trajectories exchange tracklets having equal time stamps, or if no such tracklets exist, a random tracklet from the first trajectory is moved into the second. The "switch" move makes two random trajectories exchange tracklets from the beginning of the sliding window to a random moment of time. The "change type" move changes the type of a randomly chosen trajectory.

While generating new hypotheses, each of the three types of transforms can be chose with equal probability. The trajectories and moments of time are chosen from a uniform distribution.

The probability of accepting a new hypothesis in the Metropolis-Hastings algorithm is given by:

(26) $$p(H_{i+1} \leftarrow \overline{H}) = \min\left(\frac{p(\overline{H} \mid D)q(H_i \mid \overline{H})}{p(H_i \mid D)q(\overline{H} \mid H_i)}, 1\right)$$

Here $q(H_i \mid \overline{H})$ and $q(\overline{H} \mid H_i)$ is the probability of switching from the hypothesis $\overline{H}$ to H_i and from H_i to $\overline{H}$ respectively.

2.4. Restoring People's Positions

After finding an approximation of an optimal hypothesis, the people's positions at a certain time within a sliding window are estimated.

It is not necessary for the detections to be obtained in every frame for the algorithm to work. They may be obtained once in several frames. The linear interpolation is used to estimate a person's position in an intermediate frame.

3. EXPERIMENTS

The aim of the experimental evaluation was to compare the base method [10] with its modification. The modified part consists of modeling trajectories using splines, and thus taking into account some physical features of a person's movement (see sec. 2.3 and 2.3.4).

The base algorithm and its modification were evaluated on the TownCentre dataset. It contains a high resolution video sequence ($1920 \times 1080 / 25$ fps), filmed from a static camera. The calibration matrix and the ground truth are also provided.

To evaluate the methods standard precision and recall metrics were used along with the widely known CLEAR MOT [4] metrics. Here is a brief description of some of them. FP - number of false positives; FN - number of false negatives, ID - number of identity switches, MOTA - a total error that takes into account FP, FN and ID; MOTP shows how close the trajectory lies to the real person's position obtained from the ground truth. The results are shown in table.

The results of comparing the base method [10] with its modification

Algorithm	Baseline	Modification
Precision	74.73	**76.03**
Recall	47.92	**50.02**
FP	976	**950**
FN	3137	**3011**
ID	76	**71**
MOTA	30.46	**33.08**
MOTP	**44.24**	43.95

As it was mentioned in [10], the proposed approach can be used to process a video in real time. But in order to simplify the development process we chose MATLAB to implement the algorithm. Thus the time to process one frame significantly increased. Therefore we had to resort to a very rare usage of the head detector (only once in 20 frames). Our experiments showed that the proposed algorithm was able to perform tracking even in such poor conditions, although the quality of the result significantly dropped.

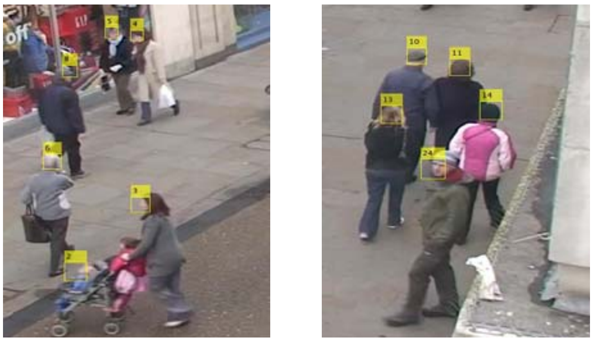

Fig. 4. An example of tracking results

[1] http://www.cvg.rdg.ac.uk/PETS2009/

[2] http://www.robots.ox.ac.uk/ActiveVision/Research/Projects/2009bbenfold_headpose/project.html#datasets

4. CONCLUSION

In this work we have proposed a modification of the base algorithm [10]. The experimental results showed that the modification improved the base method. It proves that imposing global constraints on the trajectory influences tracking performance. In future work we plan to add new factors such as occlusion reasoning. It is also possible to integrate an appearance model into the likelihood function.

5. ACKNOWLEDGMENTS

This paper was supported by RFBR, research project No. 14-01-00849 a.

6. REFERENCES

[1] Andriyenko A., Schindler K. Multi-target tracking by continuous energy minimization. CVPR, 2011.

[2] Andriyenko A., Schindler K., Roth S. Discrete-continuous optimization for multi-target tracking. CVPR, 2012.

[3] Benfold B. Reid I. Stable multi-target tracking in real-time surveillance video, CVPR, 2011.

[4] Bernardin K. Stiefelhagen R. Evaluating multiple object tracking performance: The CLEAR MOT metrics. Image and Video Processing, 2008(1):1 – 10, 2008.

[5] Black J., Ellis T., Rosin P. Multi view image surveillance and tracking. *In Motion & Video Computing Workshop*, Dec. 2002.

[6] Breitenstein M.D., Reichlin F., Leibe B., Koller-Meier E., Van Gool L. Robust tracking-by-detection using a detector confidence particle filter. In ICCV, 2009.

[7] Milan A., Schindler K., Roth S. Continious Energy Minimization for Multi-Target tracking, PAMI, 2013.

[8] Oh S., Russell S., Sastry S. Markov chain monte carlo data association for general multiple-target tracking problems // Decision and Control. 2004. 1. No 43. – P. 735 - 742.

[9] Prisacariu V.A., Reid I.D. FastHOG – a real-time GPU implementation of HOG. Technical Report 2310/09.

[10] Shalnov E., Konushin A. Improvement of MCMC-based video tracking algorithm. Pattern recognition and image analysis (PRIA-11-2013), 2013.