INTERACTIVE OFFLINE TRACKING SYSTEM FOR GENERIC COLOR OBJECTS

Zdeněk Sojma
Master Degree Programme (2), FIT BUT
E-mail: xsojma00@stud.fit.vutbr.cz

Supervised by: Adam Herout
E-mail: herout@fit.vutbr.cz

Abstract: This paper describes an interactive offline system for tracking of generic color objects. Main advantage of the algorithm is creation of object trajectory with high tracking accuracy in complex video scenes. The tracker is based on a fast object detector which uses color histogram bins as features and on the temporal coherence of object motion to compute trajectory. The system computes object trajectory from all available input data which may be interactively modified and added by user.

Keywords: object tracking, offline, interactivity, color bin feature, adaboost, mean shift

1. INTRODUCTION

Object tracking systems can be divided into two categories: *online* (real-time) and *offline*. Offline systems have one basic advantage; they have access to whole video in advance. This means that object trajectory can be calculated forward and backward through video in the same time. The user can specify object positions in several video frames (these frames are called *key frames*, also *k-frames*) which are then used to get better features for detection and to compute trajectory with higher accuracy. The user can also interactively refine output trajectory by modifying the k-frames. Offline algorithms have the best usage in very complex video scenes with many occlusions or in situations that require high tracking accuracy (for example when a film maker adds special effects to a video). My work in this paper follows an article called Interactive Offline Tracking for Color Objects [2]. My goals are to implement the application for offline tracking and evaluate the output results.

2. TRACKING ALGORITHM OVERVIEW

When the application starts, the user specifies an object state (position and size) in a two or more key frames where the first and the last k-frame determinate the tracking range in video. The system then calculates and outputs the tracking result. The user can check the output trajectory, modify or add some new key frames in problematic video sequences and restart the tracking process. Therefore, calculation time each of new tracking iteration is critical for interactive usage of the system.

A fast detector is used to speed up the tracking process in the system. The detector is based on a color histogram-based feature called *boosted color bin* (Section 3.1) that quickly rejects large majority of non-object regions. For the final trajectory calculation, an algorithm called *temporal trajectory growing* (Section 4.2) is used which exploits the temporal coherence of object motion to reject remaining non-object regions and find the object in frames where it was missed. The detector alone is not reliable enough to correctly track the object in all video frames, but the combination of the detection in sparse frames and application of the trajectory growing algorithm between these frames seems to be efficient enough to create a fast interactive object tracking system.

3. OBJECT DETECTOR

As was said earlier, the detector uses the feature called boosted color bin. This feature is based on color histograms extracted from the key frames. The color histogram has been proved very robust for generic object tracking because it is rotation and scale invariant [1]. The combination of multiple 1D color histograms is used for fast extraction of the feature.

3.1. BOOSTED COLOR BIN

A 1D color histogram is extracted from a 3D RGB color space of an object rectangle. All pixel colors of the rectangle are projected on one-dimensional line in the color space and the 1D histogram of these projected values is calculated. One 1D histogram is weak for robust object classification, but a combination of multiple histograms calculated from multiple lines where each line has a different direction, is reliable enough. The detector uses a set of 13 one-dimensional lines that pass through the point (127,127,127); each calculated 1D color histogram has only 8 bins (Figure 1) to get higher detection robustness. The color histograms are extracted in a short constant time for any frame rectangle by using *integral histogram* data structure.

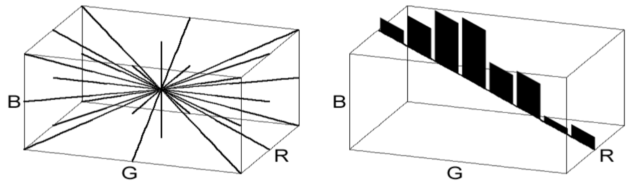

Figure 1: (left) 13 one-dimensional lines in color space, (right) a 1D color histogram of a line.

All $13 \cdot 8 = 104$ color bins are then combined in an effective way by *adaboost* algorithm which creates a strong and highly discriminative detector. To train the detector, adaboost takes the color bins as input weak classifiers and an output strong classifier is called boosted color bin. The number of bins used to successful tracking of common objects usually varies from 2 to 10.

3.2. TRAINING

The detector is trained by using positive and negative samples of an object. These samples are obtained from the key frames. Positive samples are generated from the object rectangle where the position and dimension of the rectangle is slightly varied to generate more positive samples. The colors of the object are also slightly scaled to anticipate possible appearance changes. Negative samples are evenly generated from the key frames' backgrounds by changing only the object rectangle position. The amount of generated samples is about 1000 for positive and 2000-3000 for negative samples per k-frame.

4. TRAJECTORY CREATION

The detector is not capable to perform successful object tracking alone because of false detections and partial occlusions. Therefore, the detector is combined with another algorithm which is called temporal trajectory growing. The algorithm uses the temporal coherence of an object's motion to join positively classified samples in sparse video frames into the object trajectory.

4.1. INITIAL CANDIDATES EXTRACTION

Object detection for all video frames is too time-consuming, so the detector is used only in every ten frames in the system (these frames are called *i-frames*) and these i-frames are sampled with an interval of 1/4 of object size. A *Bhattacharyya distance* is then calculated for each positively classified sample and the samples with the distance lower than 0.6 are discarded. The positions of remaining samples are moved in a gradient ascent way by *mean shift* algorithm [1] and after convergence, the samples with too close positions are merged. Finally, only ten best object candidates are stored in memory for each frame.

4.2. TEMPORAL TRAJECTORY GROWING

Trajectory growing algorithm starts from the best object candidate in each i-frame and tracks it forward and backward through the video by the mean shift algorithm. The trajectory grow is stopped when minimally other ten better trajectory hypotheses already exist in actual frame or when the current trajectory is too close to an existing better one. The best trajectory hypothesis is then shown as the system output.

5. IMPLEMENTATION

My tracking application is implemented using C++ programming language and OpenCV 2.1 library. The application now achieves 16-20 fps on a 320×240px video. Fps rate is low because the only fully implemented part of the tracker is the detector (all frames are handled as i-frames). The complete system is capable to achieve 60-100 fps according to [2].

5.1. RESULTS

Figure 2 shows an example of the detector's work on a complex scene with large background changes and a number of object pose variations. White dots represent centers of positively classified frame samples. Key frames #000, #060 and #360 are specified by the user and the detector uses 8 color bins for classification. Frames #039, #089 and #297 contain a few false positive detections. The true object is missed in #297 but the detection is successful again in a few frames later. False positives and missed objects will be handled by the temporal trajectory growing algorithm.

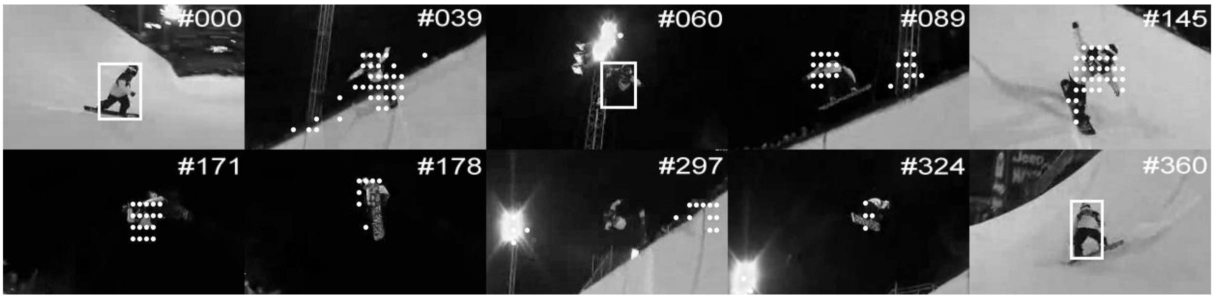

Figure 2: Detection example of a snowboarder on a superpipe.

The number of selected color bins varies from 2 to 10 in all tested videos. Calculation time of the integral histogram per i-frame is now about 50ms. Extraction of the boosted color bin from generated samples and the training alone takes less than one second. The detector is capable to reject up to 98% of non-object regions and runs at 1000 - 2000 fps (100 - 200 fps on i-frames) which is dependent on the number of selected color bins.

6. CONCLUSION

This paper describes an effective interactive tracking system for generic color objects in offline video. The system uses an efficient combination of color histogram bins for object detection and exploits the temporal coherence to generate some possible object trajectories. An implemented application can handle object detection even in complicated scenes with only a few k-frames. The speed of the tracker is now largely limited by calculation time of the integral histogram per i-frame. Future work could be focused on interactive tracking of multiple objects.

REFERENCES

[1] Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In CVPR, 2000

[2] Wei, Y., Sun, J., Tang, X., Shum, H.: Interactive Offline Tracking for Color Objects, iccv, pp.1-8, 2007 IEEE 11th International Conference on Computer Vision, 2007