An Improved CamShift Algorithm Using Stereo Vision For Object Tracking

J. Kovačević[*], S. Jurić-Kavelj[**] and I. Petrović[**]
[*] TEB-Elektronika d.o.o., Zagreb, Croatia
[**] University of Zagreb Faculty of Electrical Engineering and Computing, Zagreb, Croatia
josip.kovacevic@teb-elektronika.hr

Abstract - Stereo matching is a powerful method of image segmentation for applications such as moving objects tracking. The result of a stereo matching algorithm is a disparity image which shows the difference of the object location in the images produced by the left and right camera. In this paper, we present a tracking algorithm which is based on the CAMSHIFT (Continuously Adaptive Mean Shift) algorithm. Our algorithm operates on the stereo images, and the disparity image. Extending the CAMSHIFT algorithm with the scene depth information from the disparity image we are able to increase the tracking quality with acceptable losses in the execution time. The algorithm is implemented and evaluated in a people tracking system, where the experimental results show that our algorithm outperforms the conventional CAMSHIFT algorithm in tracking accuracy.

I. Introduction

The goal of tracking is to establish a stable track for each object of interest in successive frames. Designing a visual tracking system to track an object is a complex task because a large amount of video data must be transmitted and processed in real time. Real time stereo analysis, until recently, has been implemented in large custom hardware arrays. But computational power and algorithmic advances have made it possible to do such analysis on single processors. At the same time, increased density, speed and programmability of field programmable gate arrays (FPGA-s) [8] make custom hardware a viable alternative.

Stereo analysis is the process of measuring range to an object based on a comparison of the object projection on two or more images. The fundamental problem in stereo analysis is finding corresponding elements between the images. Correlation of image areas is disturbed by illumination, perspective, and imaging differences among images. Once the match is made, the range to the object can be computed using the image geometry.

In visual control systems, real time performance of object recognition with pose has been regarded as one of the most important issues for several decades. While modern recognition methods which are applicable to accurate pose estimation have been recently proposed, these are still inappropriate to be applied to vision based manipulation due to the high computational burden [10]. So, depending on the application requirements we have to use features simple enough, to get acceptable accuracy and real time computation.

II. Related work

The 3D information inherent in depth images affords better segmentation of objects from background and facilitates separation of objects undergoing close interaction. While some methods have tracked objects directly in the depth images themselves [2], others have found advantage in first projecting the 3D scene onto the ground plane and then tracking in the resulting “plan-view” images [3]. Because objects generally do not overlap much in the dimension normal to the ground plane in which they move, they are more easily separated and tracked in plan-view images than in the original “camera-view” images. 3D data allows plan-view object representations to be made invariant to object scale in camera-view. Despite its advantages, depth data produced by typical real-time stereo implementations contains many pixels whose values have low confidence due to stereo matching ambiguity at textureless regions or depth discontinuities. These problems increase with the distance of objects from the camera, and make it difficult to create depth-based models of tracked objects and features that are stable over successive frames.

Much progress has recently occurred in development of invariant descriptors of local features [4], and in applying these to object recognition. Augmenting stereo-based models of tracked objects with sparse local appearance features is proposed in [5]. Depth data complements sparse local features by informing correct assignment of features to objects, while tracking of stable local appearance features helps overcome distortion of object shape models due to depth noise and partial occlusion. Methods based on local features, assume that the tracked features are stable over a long period of time. So, in this approach feature representation and selection becomes a significant problem.

Color-based trackers [1, 6] have been proved robust and versatile for a modest computational cost. They are especially appealing for tracking tasks where the spatial structure of the tracked objects exhibits such a dramatic variability that trackers based on a space-dependent appearance reference would break down very fast. They have in particular been proved to be very useful for tracking tasks where the objects of interest can be of any kind, and exhibit in addition drastic changes of spatial structure through the sequence, due to pose changes, partial occlusions, etc. This type of tracking problem arises for instance in the context of video analysis and manipulation.

If the objects to be tracked are non-rigid, it is advisable to represent them with probability distributions. A straightforward way to derive a distribution model is by using histogram analysis. In this paper, we present a tracking algorithm which is based on the CAMSHIFT (Continuously Adaptive Mean Shift) algorithm. The techniques introduced independently by Bradski (CAMSHIFT) [1] and by Comaniciu (MEANSHIFT) [6] are based on the following principle: the current frame is searched for a region, a fixed-shape variable-size window, whose color content best matches a reference color model. The search is deterministic. Starting from the final location in the previous frame, it proceeds iteratively at each frame so as to minimize a distance measure to the reference color histogram. This deterministic search might however run into problems when parts of the background nearby exhibit similar colors or when the tracked object is completely occluded for a while.

Object tracking methods based on stereo cameras provide both color and depth data at each pixel. Our algorithm operates on both stereo images and the disparity image (which is converted to a depth map). Depth data complements color features giving the 3D size and location of objects, while tracking of stable color features helps overcome distortion of object shape and partial occlusion.

III. Proposed Approach

The modified CAMSHIFT algorithm can be summarized in the following steps:

1. Set the region of interest (ROI) of the probability distribution image to the entire image.
2. Select an initial location of the Mean Shift search window. The selected location is the target distribution to be tracked.
3.
    I. Calculate a color probability distribution of the region centred at the Mean Shift search window.

    II. Calculate a disparity probability distribution of the region centred at the Mean Shift search window.

    III. Calculate the final probability image combining the color and disparity probability distribution.
4. Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.
5. For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3.

The probability distribution image (PDI) may be determined using any method that associates a pixel value with a probability that the given pixel belongs to the target.

A common method is known as histogram back-projection. In order to generate the PDI, an initial histogram is computed at step 1 of the CAMSHIFT algorithm from the initial ROI of the filtered image. To calculate the histogram, we used the hue channel in HSV color space in order to isolate the pure color (Fig. 1). However multidimensional histograms from any color space may be used. Shadow-s and light seemingly changes colors which makes image processing difficult in practice. Pure color is difficult to determine if saturation or intensity values are low. Therefore, the algorithm allows setting thresholds for saturation and intensity values of colors that are included in the calculation of the PDI.

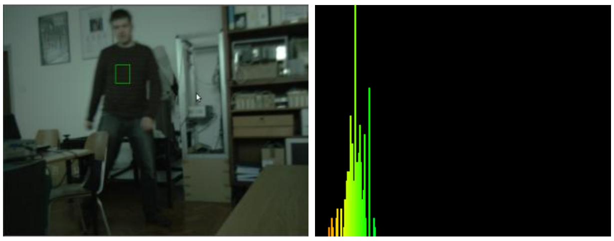

Figure 1. Marked object and its hue histogram

Histogram back-projection is a primitive operation that associates the pixel values in the image with the value of the corresponding histogram bin (Fig. 2).

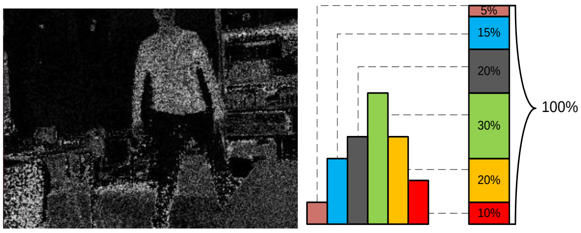

Figure 2. Histogram back-projection

In terms of statistics, the value of each output image pixel characterizes probability that the corresponding input pixel group belongs to the object whose histogram is used. This creates a black-and-white image (PDI) in which the objects are shown with lighter shades and the rest of the picture with darker shades.

In the original CAMSHIFT algorithm the color probability distribution given by the histogram back-projection represents the final probability distribution image. While the final probability distribution image in our algorithm depends on the color probability distribution (Fig. 2 left) and disparity probability distribution (Fig. 3).

The disparity probability distribution is in proportion to the difference of the disparities of the disparity image and the mean disparity of the tracked object (region centred at the Mean Shift search window). To determine the disparity probability distribution of the tracked object disparity has to be defined for each image pixel.

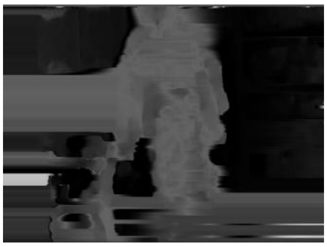

Figure 3. Disparity probability distribution

Disparity refers to the difference in image location of an object in two images. But we are not always able to figure out which parts of an image correspond to which parts of another image (correspondence problem). For example if we have an object that has a uniform color without any characteristic points it is difficult to find a pair of correspondent points for this object. Another problem are hidden points, when one point is visible from one view and not visible from another view.

Because of the correspondence problem we have to fill the disparity image. For filling disparity information in a gap along an epipolar line for which we have disparity information on the left (*disp(L)*) and right (*disp(R)*) we use a following formula to calculate the disparity:

$$disp(i) = (1 - y) \cdot disp(L) + y \cdot disp(R), \\ L < i < R \tag{1}$$

where *y* is given by:

$$y = 1 - \frac{\cos(x\pi) + 1}{2}, x \in [0,1] \tag{2}$$

An example of the resulting disparity image, after interpolation, is shown on Fig. 4.

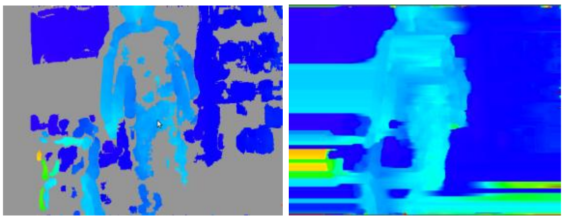

Figure 4. Disparity image before and after interpolation

The two probability distributions are multiplied and normalized to get the final distribution image (Fig. 5 left).

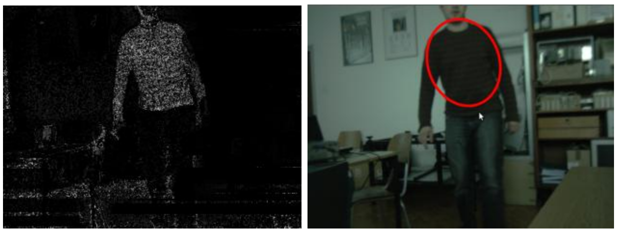

Figure 5. The final distribution image and marked new object position

The next two steps in our algorithm are the same as in the originall CAMSHIFT algrithm. We determinate the centroid of the probability image and finally mark the tracked object (Fig. 5 right).

IV. Implementation

The algorithm is implemented as a node in the ROS (Robot Operating System) framework. ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It is based on a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages.

The developed node receives image and disparity messages, and after processing, posts messages with the position information of the tracked object.

The initial position of the object to be tracked can be defined in two ways. The node allows marking of a rectangular area with in the incoming video stream (incoming image messages) which than represents the initial position of the object to be tracked. Alternatively, the node can receive a message with the initial position of the object to be tracked. These messages can be produced manually or by another processing node. Here, we developed a node which detects people faces in images, and posts messages with their position in the scene. Face detector is based on the standard Haar feature cascaded classifier [11],

To get real-time full-field distance information we used a Videre STOC (stereo-on-chip) color camera. The camera's onboard FPGA calculates the disparity image, reducing computation load on the computer and allowing for a higher resolution and refresh rate.

V. Results

The implementation of our algorithm runs at 28 Hz on average on a dual-processor 2.8GHz PC. We evaluated our method on long sequences (more than 2000 frames) captured at 30Hz and 640x480 resolution.

Our application of interest is tracking people moving around the camera (the camera is not fixed). The sequences contain objects of different types, including people, fixed and moving objects, partial occlusions, close interactions, and varying illumination.

Accuracy and performance of all test sequences were compared to the conventional CAMSHIFT algorithm. Our algorithm is on average about 1 millisecond per frame slower than the conventional CAMSHIFT algorithm with the same parameters.

Despite the complex interactions, partial occlusions, and non-rigid motions, our tracker can robustly track any objects and therefore outperforms the conventional CAMSHIFT algorithm in tracking accuracy.

VI. Conclusion

In order to overcome the disadvantages of the related works discussed, we proposed an algorithm that combines the nonparametric scale and rotation invariant CAMSHIFT algorithm with stereo vision.

Our proposed algorithm solves the problem of differentiate objects with similar color futures in wider range of situations, as it takes into account the distance of the objects to the camera. Tracking performance outperforms the original CAMSHIFT tracker.

These results suggest that our algorithm can be used for general-purpose object tracking using the disparity image and color features of the target.

Acknowledgement

This work was supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant No. 036-03630783-018.

References

[1] G.R. Bradski. Computer vision face tracking as a component of a perceptual user interface. In Workshop on Applications of Computer Vision, pages 214–219, Oct. 1998.

[2] T. Darell, D. Demirdjian, N. Checka, and J. Woodfill, Integrated person tracking using stereo, color, and pattern detection. IJCV, 37(2):175-185, June 2000.

[3] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. Real-time wide area multi-camera stereo tracking. In CVPR, pages 976–983, 2005.

[4] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In CVPR, pages II: 90–96, 2004.

[5] F. Tang, M. Harville, H. Tao, and I.N. Robinson, Fusion of local appearance with stereo depth for object tracking, Computer Vision and Pattern Recognition Workshops, CVPRW '08. IEEE Computer Society Conference on; pages 1-8, 2008.

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In Proc. Conf. Comp. Vision Pattern Rec., pages II:142–149, Hilton Head, SC, June 2000.

[7] G.R. Bradski and A. Kaehler; Learning OpenCV: Computer Vision with theOpenCV Library. O'Reilly, 2008.

[8] K. Konolige, Small vision systems: hardware and implementation, Robotics research-international symposium, pages 203-212, 1998.

[9] Seder, M. i Petrović, I. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles, 2007 IEEE International Conference on Robotics and Automation, pages 1986–1991, 2007.

[10] B. Yao and L. Fei-Fei, Modeling mutual context of object and human pose in human-object interaction activities, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 17-24, 2010.

[11] P. Viola and M. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features, 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01), 2001.