OBJECT EXTRACTION BY SPATIO-TEMPORAL ASSEMBLING

Xiaoke Qin**, Liang Tang*† *and Jie Zhou**

*Department of Automation Tsinghua University, Beijing, China
†Hewlett-Packard Labs China
imqxk00@mails.tsinghua.edu.cn , liang.tang@hp.com , jzhou@tsinghua.edu.cn

ABSTRACT

Among various algorithms for vision-based traffic monitoring, spatio-temporal (ST) slice analysis is attractive by computing over a larger temporal scale. However, it is unsuitable for further pattern recognition, since the conventional ST slice cannot preserve the spatial relationship of the original object image. In this paper, we propose a novel algorithm for accurate traffic object extraction. Compared with previous ST algorithms depending on one line per frame, we assemble the object based on foreground strips obtained from each frame and carefully designed motion estimation. Thus, both the spatial and temporal information is used more effectively. Applications in real canal traffic scenes show the advantages of our algorithm.

Index Terms— Spatio-temporal assembling, Object extraction, Canal traffic surveillance

1. INTRODUCTION

Visual surveillance has been receiving an increasing interests from many fields in recent years. For example, automatic canal surveillance systems are required to reveal the spatial and temporal distribution of ships within canal networks in real time. In this case, the task of accurate object extraction is challenging for several reasons: ships might be too large to fit in the visual field of our camera, only a small part of the scene can be illuminated by a spotlight during nighttime and real-time performance usually can not be achieved because of complex background models.

A great deal of work has been done in the area of automatic traffic surveillance systems, and several commercial systems [1] have been developed. Most of these systems employ tracking techniques for traffic parameter estimation.
Beymer et al [2] describe a method to track and group corner features for robust and real-time vehicle detection. Gupte et al [3] track and classify vehicles via detecting and analyzing the foreground regions in the image sequence. Teal et al [4] construct an automated system for detecting and tracking small maritime objects, using a specially designed foreground segmentor. Another group of approaches fits models to objects. In [5], one of us performs vehicle tracking through matching parallelogram shape models on the resulting image of an SVM-based foreground classifier. In [6], Lou et al use 3D templates for vehicle segmentation and tracking. By testing the fitness of a given object against typical models of each kind of vehicle, this method obtains not only its motion information but also its type and pose. Unfortunately, it is difficult to track large objects like ships using above algorithms within limited visual field in presence of occlusion.

On the other hand, some researchers proposed another category of algorithms based on spatio-temporal (ST) image analysis. Nakanishi et al [7] present a method for vehicle extraction on the basis of the epi-polar image (EPI). They successfully solve some occlusion problems and classify vehicles using their silhouette, although the direction of the camera is restricted. In [8], the problem of vehicle counting and classification is addressed using two 2D ST images. The system operates effectively and robustly under many difficult situations in real time. Such a method is quite suitable for canal surveillance, since the building of ST images needs only a small portion of the input image and the complete objects can be easily extracted from ST images regardless of their size. However, [8] makes use of a limited amount of spatial information by processing only one scan line per frame to generate the ST slice, from which objects are extracted. It fails to preserve the spatial relationship of original objects. Thus, its extraction results for fast moving objects may have low quality, which makes more detailed classification unfeasible.

Actually, the appearance of most traffic objects keeps unchanged when they pass in front of the camera. Therefore, it is possible to reconstruct a ship using its partial images captured at different time instants. Based on this observation, we present a novel algorithm to extract complete ships via assembling foreground regions from several frames. These regions are detected in a pre-selected narrow window instead of a line in each frame. With carefully designed motion estimation and region labeling procedures, objects are assembled accurately. In this way, the new approach combines spatial and temporal information more effectively than conventional ST approaches.

This work was supported by the National Natural Science Foundation of China (No.60573062 and No.60673106) and the Specialized Research Fund for the Doctoral Program of Higher Education.

1-4244-1437-7/07/$20.00 ©2007 IEEE

The main merits of the proposed approach include:

1. Compared with conventional methods based on the ST slice, it preserves the spatial relationship of the original object image. So it is capable to extract objects moving at different speeds with higher and constant resolution.
2. It simplifies the complexity of background modeling and foreground detection. Since some part of the scene may be more stable to model or better lit during nighttime, the complexity will be reduced when we place our detection window there.
3. It demands less computational load. Like conventional ST algorithms, we only need to process part of the original image.

The paper is organized as follows: Section 2 introduces our algorithm in detail. Section 3 presents experimental results, following by conclusions given in Section 4.

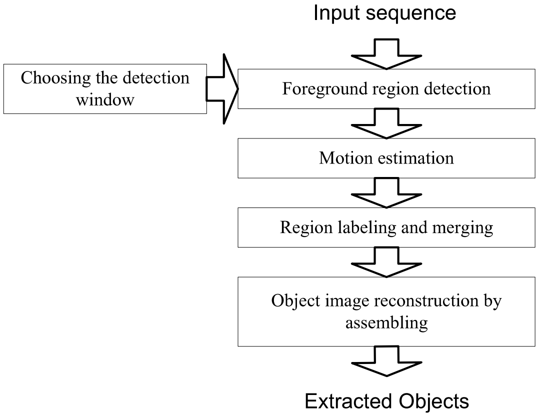
Input sequence
Choosing the detection window
Foreground region detection
Motion estimation
Region labeling and merging
Object image reconstruction by assembling
Extracted Objects

Fig. 1: Flowchart of the proposed algorithm

2. OBJECT EXTRACTION BY SPATIO-TEMPORAL ASSEMBLING

2.1. Overview of Our algorithm

Fig. 1 shows the flowchart of the proposed algorithm. First of all, we choose the suitable size and location of the detection window based on objects' speeds, the frame rate and the illumination condition. After that, foreground regions appearing within the window in each frame are extracted. Motion estimation is then applied to each foreground region. In the next stage, the foreground regions in successive frames belonging to the same object are identified according to estimated motion vectors. Finally, we assemble all these regions together to reconstruct the complete object. Note that if the perspective effect is remarkable, a preprocess stage should be employed in advance to counteract such effect.

2.2. Choosing the detection window

As mentioned above, the detection window should be placed properly in the scene, so that objects within it will be partially but clearly visible (see Fig.2 (a)). To reduce the computational load, a small window is preferred. However, it should be large enough to cover all river routes. And its width W_w should meet the following condition to guarantee that the overlapped area between two successive frames is large enough for motion estimation:

$$\frac{V_{max} * \Delta t}{W_w} < T_{overlap}, \tag{1}$$

where V_{max} is the maximal object speed in the image plane, and Δt is the time interval between two successive frames. $T_{overlap}$ is a threshold indicates the lower limit of the overlapped area of the same object within two successive frames. Theoretically, $T_{overlap}$ has to do with the richness of the texture on object images. In our implementation, we set $T_{overlap}$ 20%.

2.3. Foreground region detection and segmentation

We use a Gaussian mixture background model [9] in RGB space for background modeling. Foreground is detected on each frame via background subtraction, and a morphological filtering is adopted to remove noises. Then we get a binary motion mask M^k for the k^{th} frame (see Fig. 2 (b)).

Sometimes, a connected foreground region in M^k may consist of more than one object, because there is slight occlusion between objects in neighborhood. In this case, the following region based motion estimation and object reconstruction will not function correctly. Therefore, this region should be further segmented into smaller regions, each of which belongs to only one object. Assuming that most of the boundaries between ships are parallel to their velocities, we can approximate these boundaries by near horizontal lines. In our implementation, we perform Canny edge detection within M^k , and Hough transform on the resultant edge map (Fig. 2(c). Any detected horizontal line will be identified as a potential boundary, and used for region segmentation. Note that this process may result in over-segmentation, which will be solved in the region-merging step.

2.4. Motion estimation

Motion estimation aims to find the motion vector of a segmented foreground region. The motion vector $\mathbf{v}_j^k$ of region R_j^k (the j^{th} region in the k^{th} frame) can be estimated via minimizing the error measure E:

$$E = \frac{1}{N_j} \sum_{(x,y)\in R_j^k{'}} (I_k(x+dx, y+dy) - I_{k-1}(x,y))^2, \tag{2}$$

$$\mathbf{v}_j^k = (dx^*, dy^*) = \arg \min_{(dx,dy)} (E(dx, dy)), \tag{3}$$

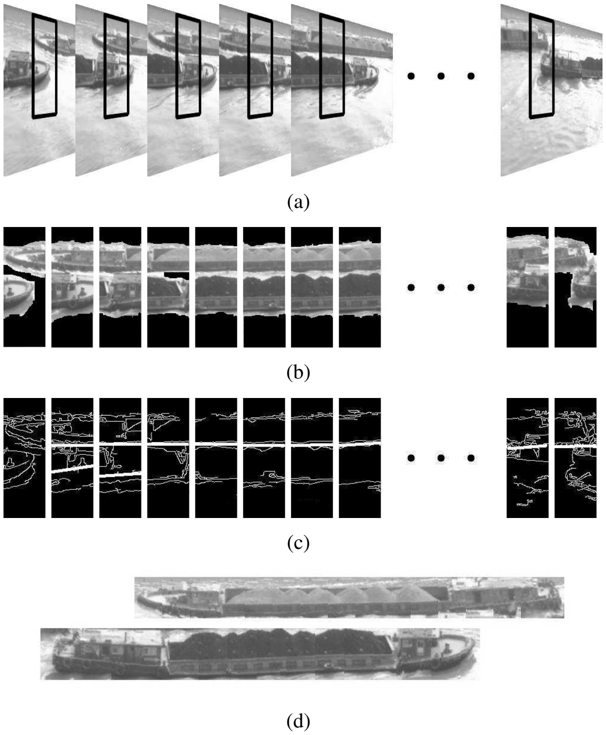

Fig. 2: Object extraction by spatio-temporal assembling. (a) Original video frames and the detection window, (b) Foreground regions, (c) Edge images and potential boundaries (horizontal white lines) and (d) Reconstructed ships.

where x-axis of the detection window denotes the direction parallel to the canal. dx and dy are bounded by the maximal interframe displacements of objects in the horizontal and vertical directions within the image plane. $R_j^{k'}$ is a part of R_j^k that still exists in the detection window when R_j^k is displaced by (dx, dy) . N_j is the number of pixels in $R_j^{k'}$. I_k is the k^{th} gray input image.

Note that if the texture or color differs insignificantly within a region, the estimated motion vector is not reliable. Besides, it is impossible to estimate the motion vector if an object appears within the detection window for the first time. To distinguish these results from reliable ones, we further define:

$$MEE(R_j^k) = mean(E(dx, dy))), \tag{4}$$

$$and \qquad MIE(R_j^k) = \min(E(dx, dy)), \tag{5}$$

and classify motion estimation results into three types:

1) $\mathbf{v}_j^k$ is valid and reliable, if $MIE(R_j^k) < T_1$ and $\frac{MIE(R_j^k)}{MEE(R_j^k)} < T_2$;

2) $\mathbf{v}_j^k$ is valid but unreliable, if $MIE(R_j^k) < T_1$ and $\frac{MIE(R_j^k)}{MEE(R_j^k)} \geq T_2$;

3) $\mathbf{v}_j^k$ is invalid, if $MIE(R_j^k) \geq T_1$.

T_1 indicates the minimal average intensity change of the same part of an object between two successive frames. T_2 guarantees that the estimation results is stable enough. Both the two thresholds should be determined according to specific scenes and appearances of objects. In our experiment, $T_1 = 20$ and $T_2 = 40\%$ empirically.

2.5. Region labeling and merging

Using estimated motion vectors, regions belonging to a same object in different frames can be identified and labeled. A region merging process is also required to counteract over-segmentation introduced in the foreground region segmentation stage.

For each foreground region R_j^k in the k^{th} frame with estimated motion vector $\mathbf{v}_j^k$, its label can be identified using following rules. (1) If $\mathbf{v}_j^k$ is invalid, assign R_j^k a new label, which indicts the emergence of a new object. (2) Otherwise, displace R_j^k by $\mathbf{v}_j^k$, and find R_l^{k-1} in the previous frame which overlaps most with R_j^k. These two regions should belong to the same object since the $\mathbf{v}_j^k$ is valid, which indicates that some corresponding parts of the two regions are identical. Therefore, we assign R_j^k the label of R_l^{k-1}. (3) Furthermore, if $\mathbf{v}_j^k$ is valid but unreliable, which means there is not enough appearance information available to estimate the correct motion vector, we set $\mathbf{v}_j^k$ the motion vector of R_l^{k-1}, assuming the object is moving at a consistent speed.

After that, we merge all regions with the same label together. For each resultant region, set its motion vector that of its largest component.

2.6. Object image reconstruction by assembling

At last, we reconstruct objects on the basis of merged regions and their motion vectors. Considering an object visible within the detection window from frame t_1 to t_2 , its complete image can be assembled, when we append all corresponding regions one after another with displacement $\sum_{i=t_1}^{t_2} \mathbf{v}^i$ respectively, where $\mathbf{v}^i$ is the motion vector of the region in the i^{th} frame. Fig. 2(d) shows the final results of our object extraction method. Since all motion vectors are estimated with pixel-precise accuracy in our implementation, there is no image blending technique required here currently.

3. EXPERIMENTAL RESULTS

The experiments are carried on real canal surveillance sequences. The input images (704 × 576) were acquired from a fixed camera aiming perpendicularly to the canal. The size of the detection window is 60 × 400. Some typical extraction results are shown in Fig. 2, 3 and 4. After the extraction, a rule-based ship classifier is employed to identify their types.

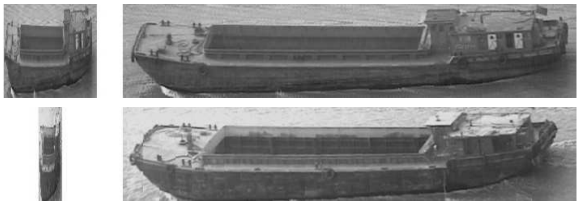

Fig. 3: Comparison results of the method in [8] (Left) and our approach (Right). The ship below is sailing three times faster than the ship above.

Our algorithm works robustly under various illumination conditions during day and night. On average, above 95% of ships are detected. 91% of detected ships are correctly classified. On a 3GHz computer with 1GB memory, our method processes one frame within 10-20 ms. The system has already met the requirements of commercial deployment.

We also make comparison with the ST slice algorithm [8]. Fig.3 shows the extraction results of two ships with different speeds. Their real lengths are similar, but the aspect ratio of images obtained from the ST slice [8] varies significantly due to the speed difference. Therefore, additional speed information is required to recover an object's length . In contrast, our approach successfully extracts the ship directly with correct aspect ratio, because the spatial relationship of the original object image is preserved.

Comparison of the resolution of extraction results is given in Fig.4. We resize the ship image extracted from the ST slice to its real size according to its speed estimated using EPI analysis. It can be concluded that the resolution of [8]'s results is not satisfying for fast ships. The ship ID is seriously blurred compared to our result. Since more details are revealed, our algorithm is more helpful for further analysis, such as OCR of ship ID and ship classification.

4. CONCLUSION

In this paper, a novel object extraction algorithm for canal traffic surveillance via spatio-temporal assembling is proposed. By suitable utilization of both spatial and temporal information, our approach improves the spatial resolution of extracted object image. Experimental results of canal traffic flow statistics prove the effectiveness of our algorithm.

Our work can be further improved by lessening the constraints imposed on object motion pattern. We believe the proposed algorithm, which combines spatial and temporal information in several successive frames, can be extended to other vision surveillance applications.

5. REFERENCES

[1] P.G. Michalopoulos, "Vehicle detection video through image processing: the autoscope system," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 21–29, 1991.

[2] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Proc. of CVPR*, 1997, pp. 495–501.

[3] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37–47, 2002.

[4] J.G. Sanderson, M.K. Teal, and T.J. Ellis, "Identification and tracking in maritime scenes," in *Proc. of IEE Int. Conference on Image Processing and its applications*, 1997, vol. 2, pp. 463–467.

[5] J. Zhou, D. Gao, and D. Zhang, "Robust moving vehicle detection for automatic traffic monitoring," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, 2007.

[6] J. Lou, T. Tan, W. Hu, H. Yang, and S.J. Maybank, "3-d model-based vehicle tracking," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1561–1569, 2005.

[7] T. Nakanishi and K. Ishii, "Automatic vehicle image extraction based on spatio-temporal image analysis," in *Proc. of ICPR*, 1992, pp. 500–504.

[8] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "Visatram: a real-time vision system for automatic traffic monitoring," *Image and Vision Computing*, vol. 18, pp. 781–794, 2000.

[9] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. of CVPR*, 1999, vol. 2, pp. 246–252.

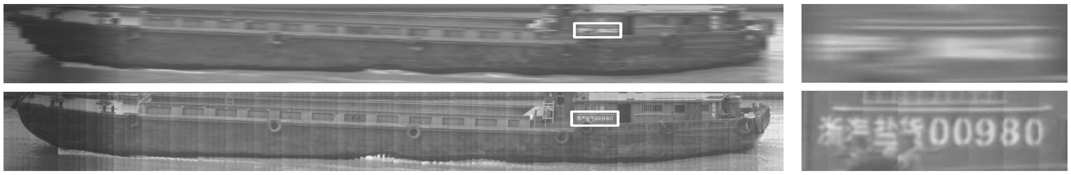

Fig. 4: Resolution comparison of the method in [8] (Up) and our approach (Bottom). The original size of extraction results (Left column) is 2273 × 226. Regions with the ship's ID are shown in their real size (Right column).