REAL-TIME COMPRESSED-DOMAIN SPATIOTEMPORAL VIDEO SEGMENTATION

Vasileios Mezaris[1,2], *Ioannis Kompatsiaris*[2], *Eleni Kokkinou*[1], *and Michael G. Strintzis*[1,2]

[1]Information Processing Laboratory
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece

[2]Informatics and Telematics Institute
1st Km Thermi-Panorama Rd,
Thessaloniki 57001, Greece
e-mail: strintzi@eng.auth.gr

ABSTRACT

In this paper, a novel algorithm for the real-time, unsupervised segmentation of image sequences in the compressed domain is proposed. The algorithm utilizes the motion information present in the compressed stream in the form of P-frame forward motion vectors, as well as basic color information in the form of DC coefficients present in I-frames. An iterative rejection scheme based on the bilinear motion model is used for performing foreground/background segmentation. Further examining the temporal consistency of the output of iterative rejection, clustering to connected regions and performing region tracking, results to foreground spatiotemporal objects being formed. Background segmentation to spatiotemporal objects is also performed. Experimental results on known sequences demonstrate the efficiency of the proposed approach and reveal the potential of employing it in content-based applications such as object-based video indexing and retrieval.

1. INTRODUCTION

Digital video is an integral part of many newly emerging multimedia applications. New image and video standards, such as MPEG-4 and MPEG-7, do not concentrate only on efficient compression methods but also on providing better ways to represent, integrate and exchange visual information [1]. These efforts aim to provide the user with greater flexibility for "content-based" access and manipulation of multimedia data. Multimedia applications, such as sophisticated query and retrieval of video, can benefit from this content-based approach. Although the standards will provide the needed functionalities in order to compose, manipulate and transmit the "object-based" information, the production of these objects is out of the scope of the standards and is left to the content developer. Thus, the success of any object-based approach depends largely on the accurate segmentation of the scene based on its contents.

This work was supported by the European IST project SCHEMA. The assistance of COST211 quat is also gratefully acknowledged.

Several approaches have been proposed in the literature for video segmentation. However, most of these approaches concentrate on segmentation in the raw, pixel domain, suffering of high computational complexity and requiring that the sequence is fully decoded before processing. To tackle these drawbacks of pixel-domain approaches, a few compressed domain methods have been proposed for spatiotemporal segmentation [2, 3, 4, 5, 6]. Some of these approaches, although significantly faster than most pixel-domain algorithms, cannot be executed in real-time [2, 5]. In [3, 6], pure translational motion is assumed, and motion vectors and dc coefficients are clustered. In [4], segmentation is performed using ac/dc DCT coefficients only; foreground/background classification is based on thresholding the average temporal change of each region, while the macroblock motion vectors are not used. In [7], a method for tracking manually identified moving objects in the compressed stream based on macroblock motion vectors is developed.

This work focuses on the real-time, unsupervised spatiotemporal segmentation of video sequences in the compressed domain. Only I- and P-frames are examined, since they contain all the information necessary for the proposed algorithm; this is also the case for most other compressed domain algorithms. The bilinear motion model is used for modelling the motion of the camera (equivalently, the perceived motion of static background) and, wherever necessary, the motion of the identified moving objects, as opposed to previous methods using simple clustering techniques which assume pure translational motion. Not only foreground spatiotemporal objects are identified, but also the background is segmented to background spatiotemporal objects; this would be useful, for example, in a retrieval application where instead of querying for a compound background, one would be allowed to query for its constituent objects, such as sky, sea, mountain, etc. The proposed spatiotemporal segmentation algorithm is applied to shots; shot detection is performed using the method of [8], due to its minimum computational complexity. An overview of the proposed scheme is presented in Fig. 1.

The paper is organized as follows: in section 2, the ex-

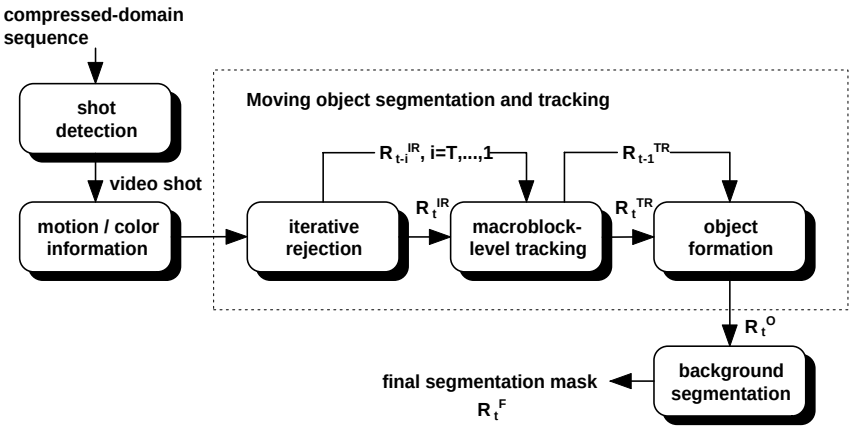


Fig. 1. Compressed-domain spatiotemporal segmentation algorithm overview.

traction of information (P-frame motion vectors, DC coefficients) used for the segmentation from the compressed stream is discussed. In section 3, moving object segmentation and tracking is developed. Section 4 deals with the segmentation of background. The application of the proposed algorithm in indexing and retrieval is briefly discussed in section 5. Section 6 contains an experimental evaluation of the developed methods, and finally, conclusions are drawn in section 7.

2. COMPRESSED-DOMAIN INFORMATION EXTRACTION

This work is focused on the fast and efficient spatiotemporal segmentation of MPEG-2 [9] compressed streams. The information used by the proposed algorithm is extracted from MPEG sequences during the decoding process. Specifically, motion vectors are extracted for the P-frames and are used for foreground/background segmentation and for subsequently identifying different foreground objects. Since P-frames are coded using motion information from I-frame to P-frame or from P-frame to P-frame, their motion information provides a clearer indication of the motion of an object in comparison to motion information derived from temporally adjacent frames. In order to derive motion information for the I-frames, averaging of the motion vectors of the closest previous and the closest next P-frame to the given I-frame is performed, rather than block matching employed in previous work on compressed-domain tracking [7].

In order to further segment the background to the different objects that it may be composed of (e.g. sky, grass, etc.), the use of color information is necessary; this is due to the fact that the background has already been identified, using the motion information, as a non-connected region of uniform motion. The color information extracted for the purpose of background segmentation is restricted to the DC coefficients of the macroblocks, corresponding to the Y, Cb and Cr components of the MPEG color space. One DC coefficient is used to describe luminance (Y) information for every macroblock. Since DC coefficients are present only in I-frames, motion information is used for temporal tracking in P-frames of the background regions formed in I-frames using the color information.

3. MOVING OBJECT SEGMENTATION AND TRACKING

The extraction of spatiotemporal moving objects is the key-challenge of any video segmentation algorithm. The proposed algorithm for moving object extraction is based on exploiting the motion information (motion vectors) of the macroblocks and consists of three main steps:

- Step 1. Iterative macroblock rejection is performed frame-wise to detect macroblocks with motion vectors deviating from the single rigid plane assumption. As a result, certain macroblocks of the current frame are activated (marked as possibly belonging to the foreground).

- Step 2. The temporal consistency of the output of iterative rejection over the last few frames is examined, to detect activated macroblocks of the current frame that cannot be tracked back to activated macroblocks for a few previous frames. These are excluded from further processing (deactivated). This process is based on temporal tracking of activated macroblocks using their motion vectors.

- Step 3. Macroblocks still activated after step 2 are clustered to connected regions and these are in turn

assigned to either preexisting or newly-appearing spatiotemporal objects, based on the motion vectors of their constituent macroblocks. Spatial and temporal constraints can also be applied to prevent the creation of spatiotemporal objects inconsistent with the human expectation (e.g single-macroblock objects or objects with undesirably small temporal duration).

These steps are explained in more detail in the sequel.

3.1. Iterative macroblock rejection

Iterative rejection is a method proposed in [10] for global motion estimation using the output of a block matching algorithm (BMA) and a four-parameter motion model. In [11], the method was extended to estimate the eight parameters of the bilinear motion model, for the purpose of retrieval of video clips based on their global motion characteristics. Iterative rejection is based on iteratively estimating the parameters of the global-motion model using least-square estimation and rejecting those blocks whose motion vectors result to an estimation error larger than the average estimation error. The iterative procedure is terminated when one iteration leaves the set of rejected blocks unaltered. The underlying assumption regarding the method of iterative rejection is that the area of the background is significantly larger than that of the moving objects; thus, applying the iterative rejection scheme to the entire frame results in motion vectors affected by local (object) motion being rejected, and global motion (camera motion or, equivalently, the perceived motion of still background) being estimated.

In this work, a byproduct of iterative rejection based on the bilinear motion model is used. Specifically, the mask R_t^{IR} indicating which macroblocks have been rejected at time t (or activated, from the segmentation objective's point of view), is employed as the first step to foreground / background segmentation. Rejected (activated) macroblocks are treated as potentially belonging to foreground objects. As opposed to methods based on examining the temporal change of color features, used for fast raw-domain foreground / background segmentation, the employed method of iterative rejection can efficiently deal with sequences captured by a moving camera.

Although this is a fast and relatively simple method for detecting macroblocks that may belong to the foreground, several of the activated macroblocks may have been falsely activated. This may be due to false motion vectors (motion vectors extracted from the compressed stream do not necessarily coincide with the true motion vectors expressing the object motion) or to motion-model inability to accurately capture the undergoing global motion. In order to identify and discard falsely activated macroblocks, the temporal consistency of the output of iterative rejection over the last few frames is examined, as discussed in the next subsection.

3.2. Macroblock-level tracking

In order to examine the temporal consistency of the output of iterative rejection, temporal tracking of activated macroblocks is performed using the motion information associated with them in the compressed stream, namely their motion vectors. The temporal tracking is based upon the work presented in [7], where objects were manually marked by selecting their constituent macroblocks and these objects were subsequently tracked in the compressed domain using the macroblock motion vectors. In the context of validating the output of the iterative rejection module, no human intervention is required, as opposed to [7]. A shortcoming of the method of [7], the need for performing block matching to extract motion features for the I-frames, is handled in this work by averaging the motion vectors of the closest previous and the closest next P-frame to the given I-frame, as already discussed in section 2.

Let R_t^{TR} denote the foreground/background mask derived from mask R_t^{IR} via macroblock-level tracking and let $\tau(.)$ be the tracking operator defined in [7], taking as input a macroblock at time t and outputting the corresponding macroblock or macroblocks at time $t+1$. Then, the operator $\mathcal{T}(.)$ can be defined as taking a foreground/background mask (such as R_t^{IR}, R_t^{TR}) as input, applying the $\tau(.)$ operator to all foreground macroblocks of that mask, and outputting the foreground/background mask at time $t+1$, as estimated by temporal tracking of all foreground macroblocks.

Using the operator $\mathcal{T}(.)$, examining the temporal consistency of the output of iterative rejection over T frames can be expressed as:

$$R_{t-T}^{temp} = R_{t-T}^{IR}$$

$$\text{for } i = T, \ldots, 1,\ R_{t-i+1}^{temp} = \mathcal{T}(R_{t-i}^{temp}) \cap R_{t-i+1}^{IR}$$

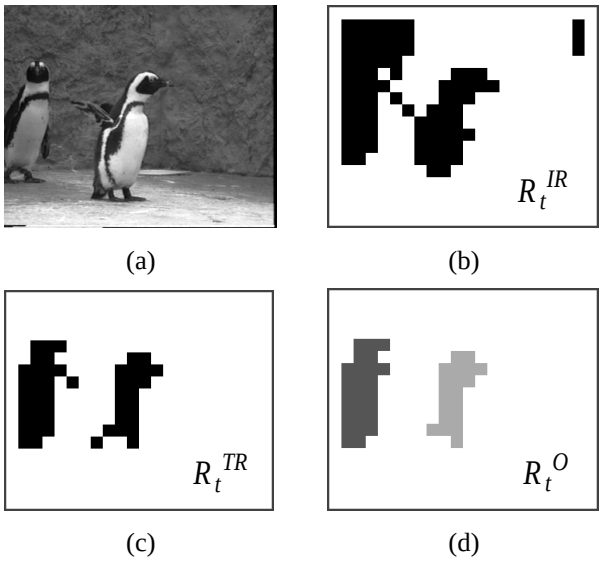
$$R_t^{TR} = R_t^{temp}$$

where $\cap$ denotes the intersection of foreground macroblocks.

It is important to observe that this process does not allow for infinite error propagation: if a macroblock is falsely assigned to the background, this will affect at most the T following frames, while it is possible that it affects no more that the current frame, since the tracking process, as explained in [7], results to tracked regions (in this case, the foreground part of the foreground/background mask) being inflated. The efficiency of macroblock-level tracking in rejecting falsely activated macroblocks is demonstrated in Fig. 2 for frame 220 of the "penguin" sequence.

3.3. Spatiotemporal object formation

As soon as falsely activated macroblocks have been rejected, as described in the previous subsection, the remaining macroblocks are clustered to connected foreground regions, using a four-connectivity component labelling algorithm [12];



(a) (b) (c) (d)

Fig. 2. Frame 220 of the "penguin" sequence: (a) original image, (b) output of iterative rejection, (c) activated macroblocks after macroblock-level tracking for $T = 4$, (d) final results showing the two spatiotemporal objects present in this frame. The usefulness of macroblock-level tracking in rejecting falsely activated macroblocks is evident.

this results to the creation of intermediate mask R_t^I. Note that this does not imply that each of these connected spatial regions in R_t^I belongs to a single spatiotemporal object o_q, as will be discussed in the sequel. Only for the first frame of the shot, in the absence of a previous object mask R_t^O (:output of the region formation and tracking step, expressing the spatiotemporal object membership of each macroblock) and for the sake of computational simplicity, each connected region is assumed to correspond to a single object:

$$\text{for } t = 0,\ R_t^O = R_t^I$$

To determine whether a given spatial region belongs to one or more pre-existing spatiotemporal objects or to a newly appearing one and eventually create the object mask R_t^O, motion projection is performed by applying the tracking operator $\tau(.)$ to the macroblocks of each spatiotemporal object of mask R_{t-1}^O. The resulting mask is denoted R_t^{MP}. Thus, every connected region s_k of mask R_t^I can be assigned to one of the following three categories:

- Cat. 1. A significant number of macroblocks of s_k have been marked in mask R_t^{MP} as belonging to spatiotemporal object o_q, and no macroblock of s_k has been marked in mask R_t^{MP} as belonging to a spatiotemporal object o_m, $m \neq q$.
- Cat. 2. A significant number of macroblocks of s_k have been marked in mask R_t^{MP} as belonging to spatiotemporal object o_q, and one or more macroblocks of s_k have been marked in mask R_t^{MP} as belonging to different spatiotemporal objects, namely o_m, $m = 1, \ldots, M$.
- Cat. 3. There is no spatiotemporal object o_q in mask R_t^{MP} having a significant number of macroblocks of s_k marked as belonging to it.

One thing that has to be defined is the exact meaning of the "significant number of macroblocks" cited in the definition of the above categories. This number is in fact estimated for every pair of a spatial region s_k of mask R_t^I and the motion projection of a spatiotemporal object o_q in mask R_t^{MP}. Let M_s, M_o denote the size of the examined pair (s_k and motion projection of o_q respectively) in macroblocks and $M_{s,o}$ the number of macroblocks of s_k that have been marked in mask R_t^{MP} as belonging to spatiotemporal object o_q. Then, the "significant number of macroblocks" requirement is satisfied if

$$M_{s,o} > a \cdot \frac{M_s + M_o}{2}, \tag{1}$$

The value of the parameter a was set to 0.5 on the basis of experimentation.

For the spatial regions s_k assigned to the third category, it is clear that they can not be associated with an existing spatiotemporal object; for that, each region of this category forms a new spatiotemporal object.

Similarly, the spatial regions s_k assigned to the first category can only be associated with a single spatiotemporal object o_q; however, more than one spatial regions may be associated with the same spatiotemporal object. In this case, the larger of these spatial regions becomes part of o_q, while the rest are discarded (their macroblocks are assigned to background). This procedure deals with objects breaking up; the fragments discarded at time t, if they actually correspond to moving objects, will clearly be assigned to category 3 at time $t + 1$, and in turn be identified as new spatiotemporal objects.

As for the regions s_k assigned to the second category, the initial correspondence of some of their macroblocks with spatiotemporal objects is employed to estimate the parameters of the bilinear motion model for each of the competing objects. Subsequently, each macroblock is assigned to the object for which the motion estimation error is minimized. This process elegantly handles moving objects that become spatially adjacent. The possibility of merging two adjacent objects, leaving unaltered any masks created at time $t - i$, $i > 0$, is also examined by estimating the parameters of their common motion model (at time t) and comparing the corresponding mean-square error with those of the motion models estimated for each object separately.

The process of moving object segmentation and tracking is terminated by imposing application-oriented restrictions regarding the size and temporal duration of valid moving objects, if any such restrictions exist. Generally, the removal of very small objects and objects of very small temporal duration, which are most likely to be false objects and are of little use in applications like indexing and retrieval, is beneficial.

3.4. Pixel-domain boundary refinement

In specific applications, the information that can be extracted from a segmentation mask of macroblock-level accuracy may be insufficient (e.g. for the extraction of shape descriptors of high accuracy). In this case, additional pixel-domain processing of a partially decompressed sequence may be required, to extract object masks of pixel accuracy. This can be performed using the color features of pixels in the area of each moving object and a Bayes classifier for two-class separation (moving object/background) to reclassify all pixels in that area or a portion of them, in a fashion similar to that of [13]. Pixel accuracy masks created using this refinement method are presented in the experimental results section.

4. BACKGROUND SEGMENTATION

Background segmentation, as soon as moving objects have been extracted, is based on classifying the remaining macroblocks (assigned to background) to one of a number of background spatiotemporal objects. This task is performed using two distinct steps, each dealing with one of the different types of the examined frames. These steps are preceded, at the beginning of the shot, by a procedure estimating the number of background objects that should be created. The result of the background segmentation is a final segmentation mask R_t^F.

Specifically, background segmentation is initiated by applying the maximin algorithm [14] to the color DC coefficients of the first frame, which is an I-frame. The maximin algorithm employs the Euclidean distance in the YCrCb colorspace to identify radically different colors; these indicate the presence of different background objects. Its output is the number of estimated objects and their corresponding colors; the latter can be used to initiate the clustering process.

In I-frames, background macroblocks are clustered to background objects using the K-means algorithm [15, 14], K being the number of objects estimated by the maximin algorithm. For the first frame of the shot, the color centers are initialized using the output of the maximin algorithm, while for the subsequent I-frames the color centers of the resulting objects in the previous I-frame are used for initialization. A recursive component labelling algorithm [12] is subsequently applied, to enforce the connectivity of the created objects and the merging of any non-connected parts of them; the connectivity constraint is useful in accurately estimating the position of each object, which could otherwise be made of non-connected parts scatterer in the entire frame.

In P-frames, the absence of color information can be dealt with by using the macroblock motion vectors and a previous final mask R_{t-1}^F. Temporal tracking is then performed as discussed in previous sections; macroblocks that are associated via the tracking process with more than one background objects are assigned to the one for which the motion information indicates a stronger association, while those not associated with any objects are assigned to one based on spatial proximity.

5. OBJECT-BASED INDEXING

The proposed algorithm is suitable for introducing object-based functionality to video indexing and retrieval applications, due to the creation of both foreground and background spatiotemporal objects, for which object-based descriptors in the context of the MPEG-7 Visual standard [16] can be extracted. Examples of such standardized descriptors include the *dominant color descriptor*, the *scalable color descriptor*, *contour-based* and *region-based* shape descriptors, *motion trajectory* and *parametric motion* descriptors. The use of such object-based descriptors would allow for more expressive queries to be processed and more efficient indexing and retrieval to be performed, compared to key-frame based indexing.

6. EXPERIMENTAL RESULTS

The proposed method was tested on a variety of video sequences. Here, results are presented for the "Penguin" (Fig. 3) and "Coast-guard" (Fig. 4) sequences. Segmentation masks both before (R_t^O, second column of results figures) and after the background segmentation (R_t^F, third column of results figures) are presented, to clearly demonstrate the foreground and background objects identified in the compressed stream by the proposed algorithm. Results after pixel-domain boundary refinement for the moving objects are also presented in Figs. 3 and 4. It can be seen from the results figures that the algorithm has succeeded in extracting the real foreground objects depicted in the sequences. No over-segmentation is caused by the proposed approach, thus facilitating the formation of semantically meaningful spatiotemporal objects. Additionally, very few false objects are created. Moving objects that have halted, as the rightmost penguin in Fig. 3, are assigned new labels when they resume moving.

Additionally, the proposed approach succeeds in adding as little computational overhead as possible to the computational complexity of a standard decoder. In particular, the compressed-domain segmentation algorithm presented (excluding any processes of the MPEG decoder and the storage of the segmentation masks, which is algorithm-independent and unnecessary in many cases, e.g. for the subsequent extraction of object-based indexing features) requires on average 5.02 msec per processed I/P-frame on an 800Mhz Pentium III. This translates to almost 600 frames per second considering the presence of two consecutive B-frames between every two I/P-frames, which is typical for MPEG-2 sequences and is the case for the employed test media. The pixel-domain boundary refinement of section 3.4 requires on average 0.48 sec. per processed I/P-frame.

7. CONCLUSIONS

An algorithm for the unsupervised segmentation of compressed domain image sequences was presented in this paper. The algorithm was shown to perform in real-time on a PC, producing semantically meaningful spatiotemporal objects both for the foreground and the background. Due to its real-time, unsupervised operation, the proposed algorithm is appropriate for content-based multimedia applications requiring the manipulation of large volumes of visual data, such as object-based video indexing and retrieval.

8. REFERENCES

[1] Tutorial issue. *Signal Processing:Image Communication, Tutorial issue on MPEG-4*, 15(4-5), 2000.

[2] R. Wang, H.-J. Zhang, and Y.-Q. Zhang. A confidence measure based moving object extraction system built for compressed domain. In *Proc. IEEE International Symposium on Circuits and Systems*, volume 5, pages 21–24, 2000.

[3] H.-L. Eng and K.-K. Ma. Spatiotemporal segmentation of moving video objects over MPEG compressed domain. In *Proc. IEEE International Conference on Multimedia and Expo*, volume 3, pages 1531–1534, 2000.

[4] O. Sukmarg and K.R. Rao. Fast object detection and segmentation in MPEG compressed domain. In *Proc. IEEE TENCON 2000*, volume 3, pages 364–368, 2000.

[5] R.V. Babu and K.R. Ramakrishnan. Compressed domain motion segmentation for video object extraction. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 3788–3791, 2002.

[6] N.V. Boulgouris, E. Kokkinou, and M.G. Strintzis. Fast Compressed-domain Segmentation For Video Indexing and Retrieval. In *Proc. Tyrrhenian International Workshop on Digital Communications (IWDC 2002)*, pages 295–300, Sept. 2002.

[7] L. Favalli, A. Mecocci, and F. Moschetti. Object tracking for retrieval applications in MPEG-2. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(3):427–432, Apr. 2000.

[8] V. Kobla, D.S. Doermann, and K.I. Lin. Archiving, indexing, and retrieval of video in the compressed domain. In *Proc. SPIE Conference on Multimedia Storage and Archiving Systems, vol. 2916*, pages 78–89, 1996.

[9] MPEG-2. Generic Coding of Moving Pictures and Associated Audio Information. Technical report, ISO/IEC 13818, 1996.

[10] G.B. Rath and A. Makur. Iterative least squares and compression based estimations for a four-parameter linear global motion model and global motion compensation. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(7):1075–1099, Oct. 1999.

[11] T. Yu and Y. Zhang. Retrieval of video clips using global motion information. *Electronics Letters*, 37(14):893–895, July 2001.

[12] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill International Editions, 1995.

[13] V. Mezaris, I. Kompatsiaris, and M.G. Strintzis. A framework for the efficient segmentation of large-format color images. In *Proc. International Conference on Image Processing*, volume 1, pages 761–764, 2002.

[14] N. V. Boulgouris, I. Kompatsiaris, V. Mezaris, D. Simitopoulos, and M.G. Strintzis. Segmentation and Content-based Watermarking for Color Image and Image Region Indexing and Retrieval. *EURASIP Journal on Applied Signal Processing*, April 2002.

[15] J. McQueen. Some Methods for Classification and Analyis of Multivariate Observations. In *5th Berkely Symp. on Math. Stat. and Prob.*, volume 1, pages 281–296, 1967.

[16] T. Sikora. The MPEG-7 Visual standard for content description - an overview. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6):696–702, June 2001.

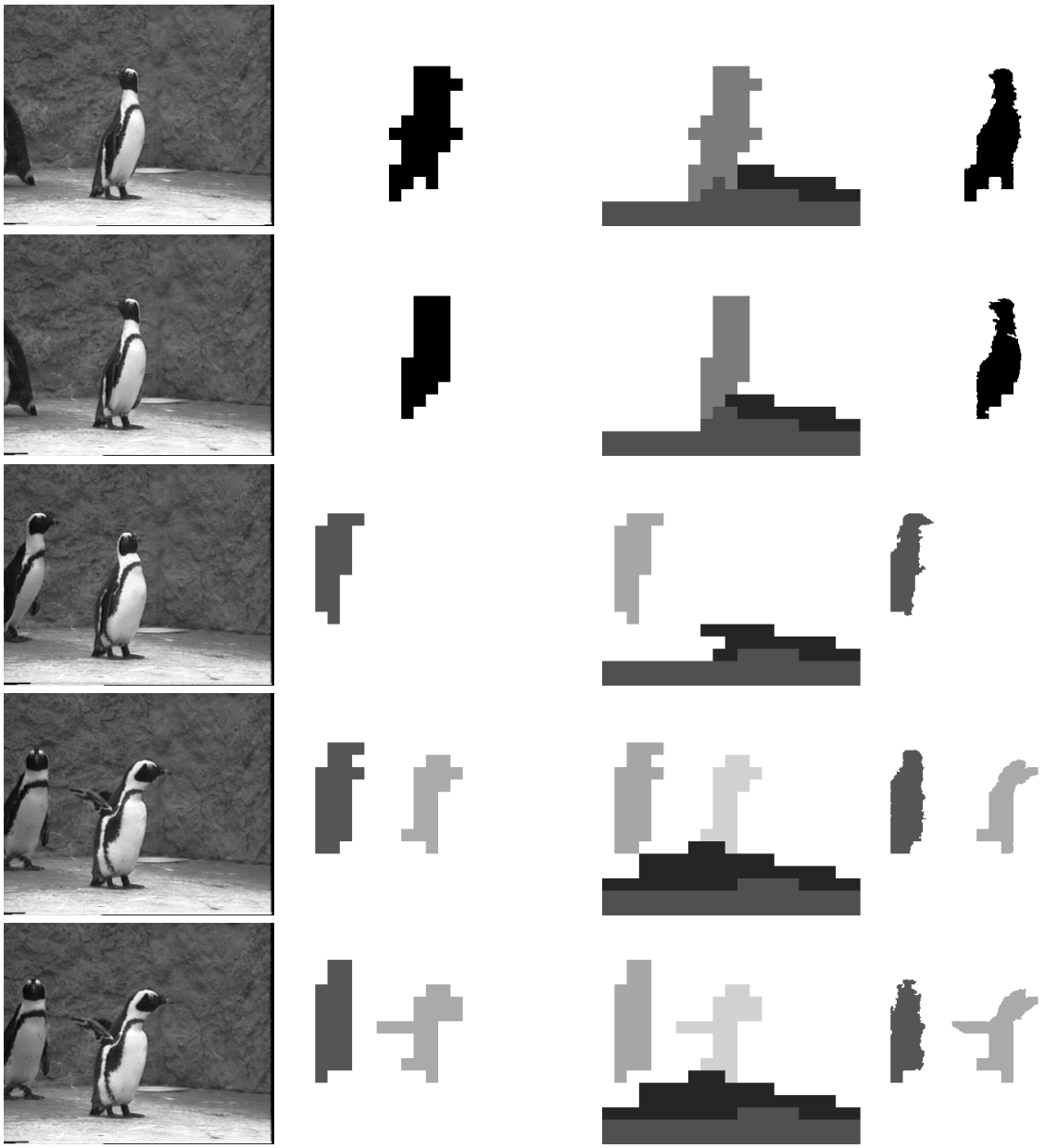


Fig. 3. Results of moving-object detection, final mask after background segmentation, and moving objects after pixel-domain boundary refinement for the “Penguin” sequence, frames 1, 7, 175, 220, 223.

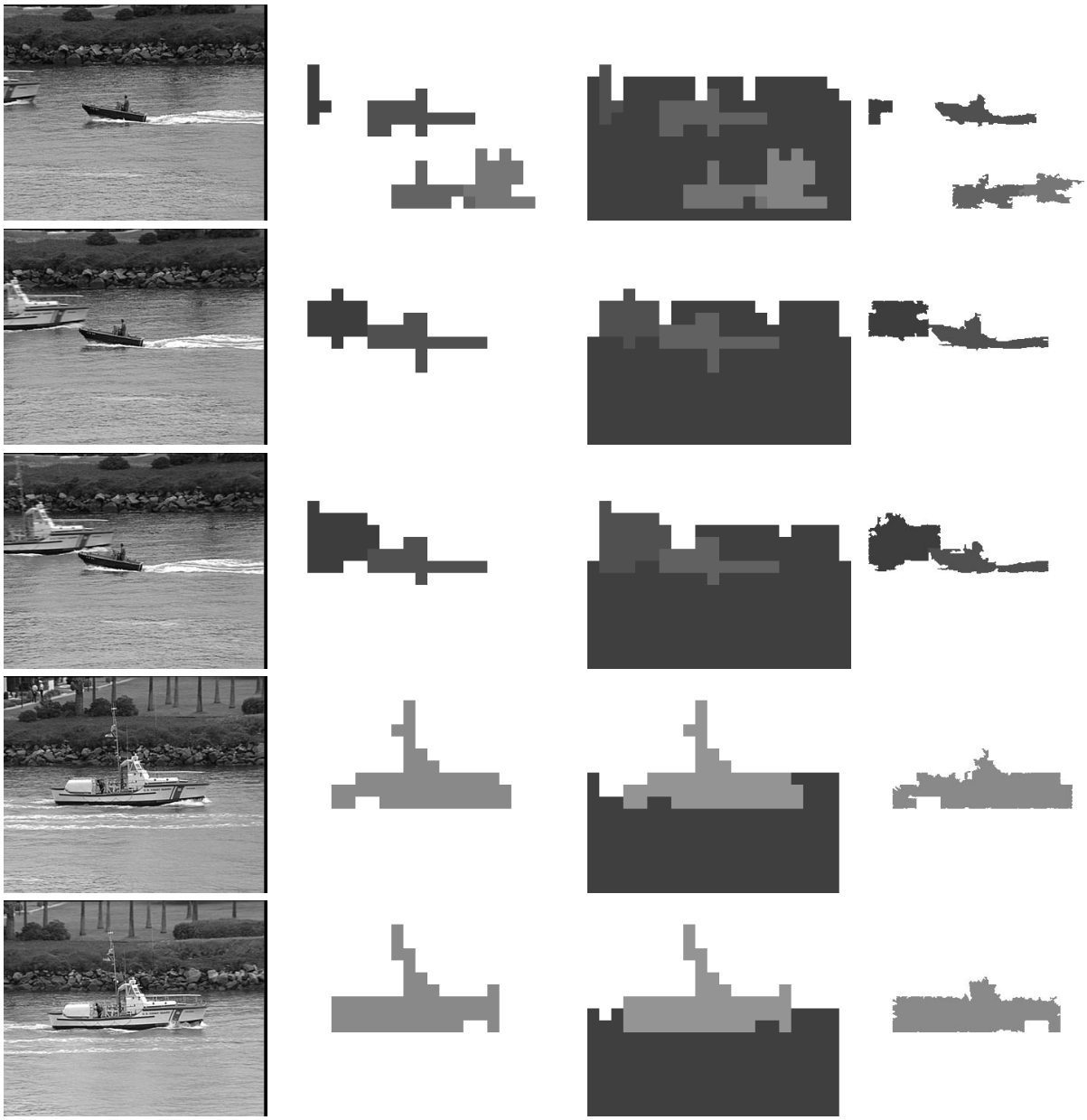


Fig. 4. Results of moving-object detection, final mask after background segmentation, and moving objects after pixel-domain boundary refinement for the “Coast-guard” sequence, frames 10, 28, 37, 202, 250.