A LVQ-BASED TEMPORAL TRACKING FOR SEMI-AUTOMATIC VIDEO OBJECT SEGMENTATION

Mochamad Hariadi and Mauridhi Hery Purnomo[*]

ABSTRACT
This paper presents a Learning Vector Quantization (LVQ)-based temporal tracking method for semi-automatic video object segmentation. A semantic video object is initialized using user assistance in a reference frame to give initial classification of video object and its background regions. The LVQ training approximates video object and background classification and use them for automatic segmentation of the video object on the following frames thus performing temporal tracking. For LVQ training input, we sampling each pixel of a video frame as a 5-dimensional vector combining 2-dimensional pixel position (X,Y) and 3-dimensional HSV color space. This paper also demonstrates experiments using some MPEG-4 standard test video sequences to evaluate the accuracy of the proposed method.
Keywords: LVQ, HSV, MPEG-4, Video Object Segmentation.

ABSTRAK
Paper ini membahas tentang metode temporal tracking berbasis Learning Vector Quantization untuk segmentasi video object secara semi-automatic. Suatu semantic video object diinitialisasi menggunakan bantuan user pada sebuah reference frame, untuk memberi klasifikasi patokan dari object yang diinginkan serta backgroundnya. LVQ training mengaproksimasi video object dan backgroundnya menggunakannya untuk men-segmentasi video object pada frame-frame berikutnya, hal ini dikenal dengan istilah *t*emporal tracking. Untuk mentraining LVQ, diinputkan secara sampling setiap pixel dari sebuah video frame dalam bentuk vector 5-dimensi, yang mengkombinasikan 2-dimensi posisi pixel pada koordinat (X,Y) dan 3-dimensi color space HSV. Paper ini juga menampilkan hasil experiment menggunakan standar test video MPEG-4 untuk mengevaluasi keakuratan dari methode yang kami usulkan.
Kata kunci: LVQ, HSV, MPEG-4, Segmentasi Video Object.

1. INTRODUCTION

Nowadays, extracting the shape information of semantic objects from video sequences is the key operation for multimedia content description, content-based representation (Sikora 1997), image retrieval, multimedia database, movie manipulation etc. An example of video object segmentation is the application in movie special effect, where seamless integration of natural video objects with synthetic elements (like cartoons) requires flexible video representation schemes. However, a video sequence does not provide the shape information of its semantic object. Recent developments in video object segmentation lead to two types of algorithms, i.e. automatic segmentation (Guo and Kim 1999), and semi-automatic segmentation (Gu and Lee 1998; Castagno and Kunt 1998). Automatic segmentation tracks the object by using some invariant parameters like color, pattern and motion. The main problem of this method is the difficulty for automatically segmenting the semantically meaningful object. Tekalp (Bovik 1998) said that until now, there is no guarantee that any of the resulting automatic segmentation will be semantically meaningful, since a semantically meaningful object may have multiple colors, pattern and multiple motions. There is still some possibility of achieving fully automatic motion segmentation even though the accuracy will be limited. Therefore, semantically meaningful video object segmentation generally requires user interaction to define the object of interest in at least one key frame. Due to the ill-posed definition of a semantically meaningful object itself, semi-automatic segmentation methods that incorporate the user's interaction become more popular. In semi-automatic algorithms, human assistance is required to identify semantic objects of interest to the segmentation system. These object of interest regions are tracked temporally for each of the following frames based on the segmentation result obtained from its previous frame.

This paper employs semi-automatic segmentation method, which incorporates human assistance to define the semantic video objects. Until now, only human knows the meaning of "semantic", there is no computer program that can fully understand the meaning of semantic. Therefore, human can give the information of semantic objects regions to the computer program for tracking the object regions on consecutive frames

[*] Jurusan Teknik Elektro, FTI ITS, Jl. Arief Rahman Hakim, Surabaya (60111)
E-mail: mochar@elect-eng.its.ac.id

1.1 Temporal Tracking Method for Video Object Segmentation

Due to strong association between consecutive frames, it is possible to track the similarity of some visual properties using a reference frame. The frames within a single shot are usually inclined to share similar visual properties, which are often semantically meaningful. Similarities within consecutive frames can be defined in terms of regions and objects similarities. The comparison may be carried out on the entire frame, or limited to certain object or region of interest in each frame. In recent years, most temporal tracking methods use color homogeneity to separate multiple regions included in a single frame. An example of such color homogeneity is skin color (Wu 1998; Kjeldsen 1996). Other possible characteristics include object shapes such as human or car, single pattern objects such as the skin of animals, road pattern, etc. On the other hand, as we have mentioned before, a semantic object may contain multiple regions with different colors, textures and motions. Therefore, the use of a single color or texture for object segmentation cannot lead to a satisfactory result. Another research using motion segmentation provides a coarse mask of moving objects where the object boundary is too rough to lead to an accurate extraction.

1.2 Object Segmentation by 5-Dimensional Feature Vectors

Some constraints of video object segmentation come from complex and cluttered background, lighting perspective, deformation of object, color similarities between object and background, video sequence image noise, etc. Therefore, this paper proposes pixel-wise object-based temporal tracking aiming for accuracy in pixel level. Here, each pixel of video frame is considered as 5-dimensional data vector combining pixel position coordinates and HSV color space components. There are some reasons of using the combination of pixel position and color information. If the approach uses *pixel color information only*, background complexity, image noise and color similarity between the object class and background class will make problems to determine the pixels of object of interest class. If the

approach uses *pixel position only,* the deformation of object class will make problems. Almost all semantic objects are not rigid objects. A rigid object preserves its shape and changes its position only by translation or rotation. Block matching techniques can track rigid objects. On the other hand, a non-rigid object has a tendency to change its outline whether it is moving or not. The 5-dimensional vector components have different coordinate spaces. As the geometric type of HSV color space is hexcone (or cylindrical) , we need to convert it into Cartesian coordinate space. This combination will act as one single vector

$(x; y; S \cos H; S \sin H; V)^T$ ....(1)

This vector should be normalized to prevent one feature domination (Hariadi et al. 2002).

2. LEARNING VECTOR QUANTIZATION (LVQ)

LVQ is found by Teuvo Kohonen, and it is closely related to SOM (Self Organizing Feature Maps) and classic VQ (Vector Quantization) (Kohonen 2001). While SOM and classic VQ are unsupervised clustering and learning methods, LVQ is supervised clustering and learning method. Unlike SOM, LVQ is without topological structure. It means that LVQ will only provide the information of each neuron instead of preserving the topological structure. The aim of LVQ is to define class regions decision or statistical pattern classification in the input data space.

Classical VQ tends to approximate the input data space $\Re^n$ by forming a quantized approximation to the input data vector $x \in \Re^n$ using a finite number of codebook vector $m_i \in \Re^n, i = 1, 2, \ldots, N$.

Assume that vector $\boldsymbol{m_c}$ is the nearest codebook vector to vector $\boldsymbol{x}$. To find the codebook vector $\boldsymbol{m_c}$ that approximates vector $\boldsymbol{x}$ in the input space, we can use a Euclidean distance measurement:

$$c = \arg\min_i \{\|x - m_i\|\},$$ ....(2)

where $\boldsymbol{c}$ is the index of the closes codebook vector $\boldsymbol{m_i}$ to input vector $\boldsymbol{x}$. Since LVQ is meant for statistical pattern classification, it constructs the VQ codebook vectors by classifying them into classes or categories. This paper employs the basic LVQ1 algorithm as follows:

if x and m_c belong to the same class, then:

$$m_c(t+1) = m_c(t) + \alpha(t)\,[x(t) - m_c(t)],$$

if x and m_c belong to the different classes then:

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)],$$

the other codebook vectors remain the same

$$m_i(t+1) = m_i(t) \text{ for } i \neq c.$$ ......(3)

It means that if the class label of the

codebook vector $\boldsymbol{m_i}$ matches the class label of the training sample $\boldsymbol{x}$, then the codebook vector moves towards $\boldsymbol{x}$. Otherwise, it moves away from the inputs sample. The other codebook vectors will remain the same where $0 < \alpha(t) < 1$ is the corresponding learning rate. In this paper we employ LVQ1 with its optimization of learning rates α.

The basic LVQ1 algorithm can be extended in such away that a different learning rate $\alpha_I(t)$ is assigned to each m_i Thus, the learning process from eq. (3) can be express as follows:

$$m_c(t+1) = [1-s(t)\alpha_c(t)]m_c(t)+s(t)\alpha(t)x(t), \quad (4)$$

where $s(t) = +1$, if the classification is correct, and $s(t) = -1$ if the classification is wrong. The 'optimal" values of $\alpha_i(t)$ for fast convergence of eq. (4) are determined in the following recursion equation

$$\alpha(t) = \frac{\alpha(t-1)}{1+s(t)\alpha(t-1)}. \quad (5)$$

3. VIDEO OBJECT SEGMENTATION SYSTEM

This section describes how the video object segmentation system works. Fig.1 shows the flowchart of the video object segmentation system.

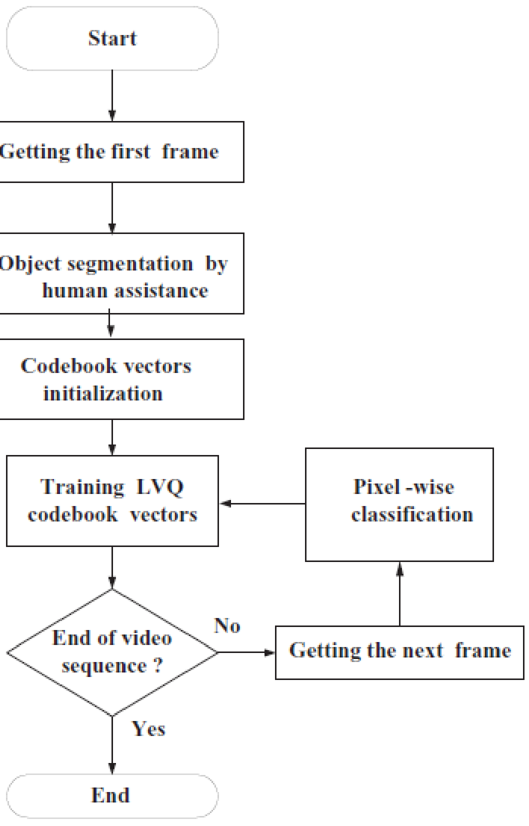
Start
Getting the first frame
Object segmentation by human assistance
Codebook vectors initialization
Training LVQ codebook vectors
Pixel -wise classification
End of video sequence ?
No
Getting the next frame
Yes
End

Fig. 1. Video object segmentation flowchart

3.1 Semantic User Interaction

The semi-automatic type of video object segmentation incorporates user interaction for generating the semantically meaningful object of interest. The temporal tracking for semantic object at consecutive frames uses this object of interest as reference frame. In this paper, we aim to use only the first frame as the reference frame. To create the segmentation of the object of interest, we use Adobe Photoshop 6 software to cut the object of interest and separate it from its background

3.2 Codebook Vectors Initialization and Classification

Codebook vectors initialization and classification employs the object of interest created by human assistance as reference frame, followed by creating a window called the data sample window surrounding the object of interest. The data sample window is created by finding the maximum and minimum pixels position (boundary) of the object of interest class in (x,y) coordinates, and given a certain margin from those pixels' position. This window could save the computation time and to increase the density of codebook vectors quantization. Codebook vectors initialization begins with randomly distributing them in the pixel position domain inside the data sample window. The color components of the codebook vectors shall be their $(H; S; V)$ color information at their pixel positions. After that, create the nearest neighbor regions for each codebook vectors using eq. (2). The classification of each codebook vector depends on the class label of the majority pixels inside its nearest neighbor region. If the number of pixels with object class label is greater than the number of pixels with background class label, then the codebook vector classification is object class codebook vector and vice versa. Refer to Fig.2 algorithm.

3.3 Temporal Object Tracking

The codebook vectors are trained using LVQ1 algorithm eq. (3) and OLVQ1 algorithm for learning rate optimizer eq. (5). The key idea is to adjust the codebook vectors' values such that they can give the optimal class region decision between the object of interest class and the background class. The LVQ training begins by randomly selecting a set of sample input pixels inside the data sample window. Using eq. 2), the closest codebook vector to each sample pixel input vector is found sequentially. Each codebook vector winner is updated relative to each nearest input vector using LVQ1 algorithm (eq. (3)).

For each iteration, the corresponding learning rate ® is updated using OLVQ1 (eq. (5)). The LVQ learning process updates the codebook vectors for the succeeding frames. It is difficult to find the optimal number of iterations for convergence (Kohonen 2001), so for this implementation we employ an experimentally determined number. Codebook vectors training algorithm is in Fig.3.

3.4 Pixel-Wise Classification

Pixel-wise classification will create the result of video object segmentation. Classification is done by finding the nearest neighbor regions for object class and background class pixel respectively. In other words, the object segmentation is created by calculating the quantization of each codebook vector, and labeling all pixels the class of the nearest codebook vector (Fig. 4). Pixels outside the data sample window will be considered as background class pixels.

4. EXPERIMENT

The experiment demonstrates the implementation of video object segmentation for MPEG-4 standard test video sequences. We choose 100 frames of each video sequence which content with specific characteristics of semantic objects. We also test the segmentation program using only one frame as reference (the 1st frame of video sequence), and multiple frames as references. The LVQ algorithm uses LVQ1 with OLVQ1 as the learning rate optimizer. Experiment is run with this common parameter: Codebook vectors ***N*= 400** (1-dimension). Learning time **T = 20000.** Initialization of learning rate **α_0= 0.3**. Data sample window margin = **20** pixels.

```
Procedure Codebook vectors classification
   1. begin
   2.      N := 400; {codebook vectors amount}
   3.    Load segmented image frame;
   4.    Load target frame;
   5.    Create data sample window at target image;
   6.    Choose N codebook vectors randomly inside
         data sample window;
   7.    Find nearest neighbor region for
         each codebook vector using (eq. 2);
         {labelling codebook vectors}
   8.    for each nearest neighbor region do
   9.    class :=0 {flag for class label }
   10.    begin
   11.      for each pixel do
   12.        begin
   13.          if the pixel label of segmented image
             is object then
   14.          class := class + 1
   15.        else
   16.          class := class – 1
   17.        end
   18.      if class > 0
   19.        codebook vector label is object
   20.      else
   21.        codebook vector label is background
   22.    end;
   23. end;
```

Fig. 2. Algorithm for codebook vectors classification

Until now, it is very difficult to evaluate the accuracy of video object segmentation since it can only be observed by human eyes. Anyway, an accuracy evaluation is required to test whether the algorithm works or not. Therefore, in this video object segmentation experiment, the evaluation of segmentation accuracy is done by comparing the segmentation result of the proposed method to object segmentation using human assistance.

```
Procedure Codebook vector Training using LVQ1 with learning rate optimizer OLVQ1
{Assume all codebook vector has been initialize and classified }
    1. begin
    2.     N := 400; {initialize N codebook vectors}
    3.     T := 20000; {initialize LVQ learning time}
    4.     for t :=1 to T do
    5.       begin
    6.         Choose an input vector randomly inside data sample window;
    7.         Find the closest codebook vector to the input vector using eq. 2);
    8.         Update the closest codebook vector using LVQ1 rule (eq. 3);
    9.         Update $\alpha$ as corresponding learning rates using OLVQ1 (eq.5);
    10.      end;
    12. end;
```

Fig. 3. Algorithm for codebook vectors training using LVQ1 rule with OLVQ1 for learning rate optimizer.

We do the experiment in two ways, the first one with only single reference frame, and the other one with multiple reference frames, which giving the segmentation system the user assistance every 10 consecutive frames. The accuracy of the segmentation system for each MPEG-4 test video sequences are different.

```
Procedure Creating segmentation result
    1. begin
    2.     N := 400; {nitialize N codebook vectors}
    3.     Load target frame;
    4.     Load codebook vector from training step;
    5.     Find the nearest neighbor region for each codebook vector using eq. 2);
           {create object segmentation}
    6.     Give background label to all pixel outside data sample window;
    7.     for each nearest neighbor region do
    8.       begin
    9.         for each pixel do
    10.          begin
    11.            if codebook vector label is object then
    12.              pixel label is object
    13.          else
    14.              pixel label is background;
    15.          end;
    16.    end;
    17. end;
```

Fig. 4. Algorithm for pixel-wise classification for creating object segmentation result

$$matches = O_s \cap O_m \qquad (6)$$

$$losses = O_m - (O_s \cap O_m) \qquad (7)$$

$$overlaps = O_s - (O_s \cap O_m) \qquad (8)$$

$$error = \frac{losses + overlap}{O_m} x100\% \qquad (9)$$

where:
O_s **=object class from video object segmentation**
O_m **=object class from manual segmentation**

For video sequences with still background like in Foreman (Fig.9), the video object is well segmented. The difference of error percentage between single reference and multiple reference frames are not so significant (Fig. 5). The same result is occur also in coastguard video sequence, the object is rigid, only the background is moving. See the result in (Fig. 11) and the error percentage in (Fig. 6). The Mobile video sequence (Fig.12) which background is moving, some errors occur due to the very similar color between the moving ball and the calendar in very near position. Error percentage also increases in the last frames of Mobile video sequence due to occlusion (another object covering the moving ball). Therefore the difference of error percentage between single reference and multiple reference frame is significant at the last frame (Fig. 7). The color similarity between object and background is also occur in hall monitor. See the segmentation result in (Fig. 10) and error percentage at (Fig. 8).

5. CONCLUSION AND FUTURE PLANS

In this paper, the video object segmentation is developed based on Learning Vector Quantization (LVQ). The video object segmentation system use semiautomatic method, hence it requires human assistance to classify object of interest. The data input vectors are generated from the pixel value of video sequence frames. Each pixel of video frames is considered as a 5-dimensional vector which components are the combination of pixel position information in (*x,y*) coordinate and HSV color space (*H, S, V*). The experiment results demonstrate satisfactory segmentation even if the background color similar to object of interest. Nevertheless, for some cases in MPEG-4 video sequences test, the results are not so satisfactory due to color similarity
in very near position between object of interest and background pixels. To avoid misclassification of object class due to color similarity, pattern similarity between objects, in our future research we will improve the combination of pixel position and color information in 5-dimensional vector by adding motion vector as one of the vector components.

6. REFERENCES

Bovik, A.C. (1998), **The Hand Book of Image and Video Processing**, Academic Press Limited, 1st edition.

Castagno, R.T.E. and Kunt, M. (1998), 'Video Segmentation Based on Multiple Features for Interactive Multimedia Applications', *IEEE. Trans. Circuits Syst. Video Technology*.

Gu, C. and Lee, M.-C. (1998), 'Semiautomatic Segmentation and Tracking of Semantic Video Objects', *IEEE. Trans. Circuits Syst. Video Technol.*

Guo, J., Kim, J.C.-C. J. K. (1999), 'Fast and Accurate Moving Object Extraction Technique for MPEG-4 Object-based Video Coding', *SPIE Visual Communication and Image Proc.'99*, San Jose, CA.

Hariadi, M., Harada, A., Aoki, T. and Higuchi, T. (2002), 'An LVQ-based Human Motion Segmentation', *IEEE APCCAS 2002*, Bali, Indonesia.

Kohonen, T. (2001), **Self-organizing Maps**, The 3rd Edition, Springer-Verlag, Berlin Heidelberg, Germany.

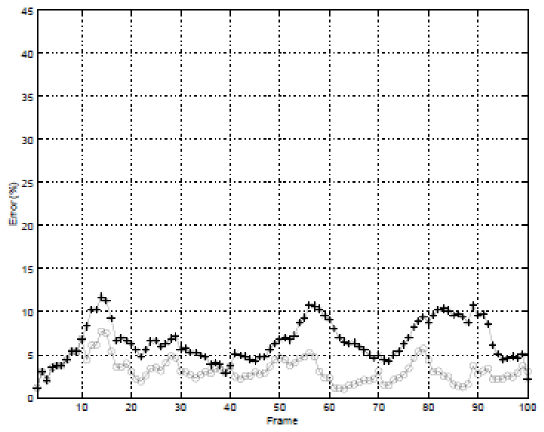

Fig. 5. Error percentage of "foreman" for frame 0 - 100, (-*) line uses only first frame as reference frame, (-o) line uses new refence frame every 10 frames

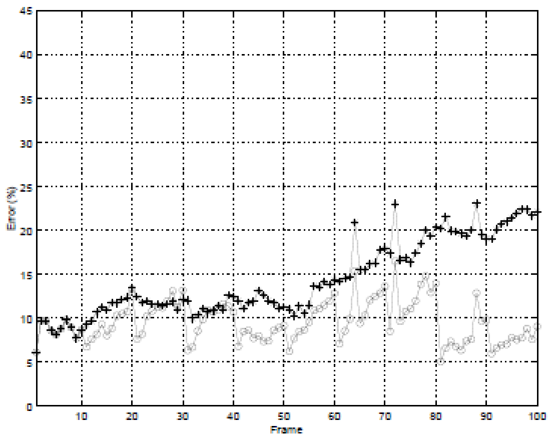

Fig. 6. Error percentage of "coast guard" for frame 0 - 100, (-*) line uses only first frame as reference frame, (-o) line uses new refence every 10 frames

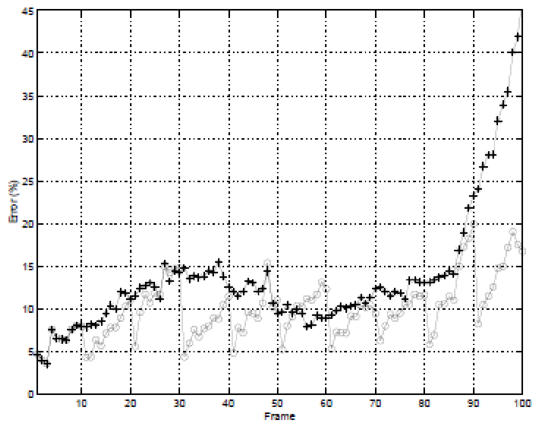

Fig. 7. Error percentage of "mobile" for frame 0 - 100, -* line uses only first frame as reference frame, -o line uses new refence every 10 frame

Diterima: 22 Januari 2007
Disetujui untuk diterbitkan: 10 Mei 2007

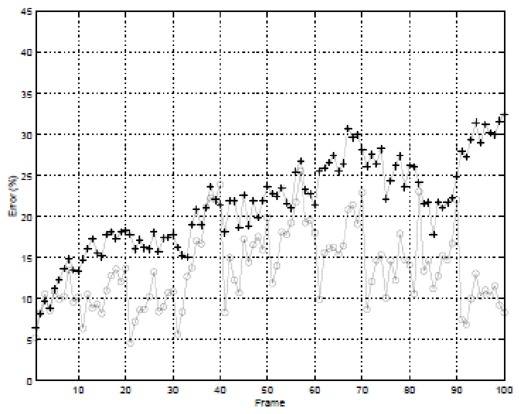

Fig. 8. Error percentage of "hall monitor" for frame 0 - 100, (-*) line uses only first frame as reference frame, (-o) line uses new refence every 10 frames

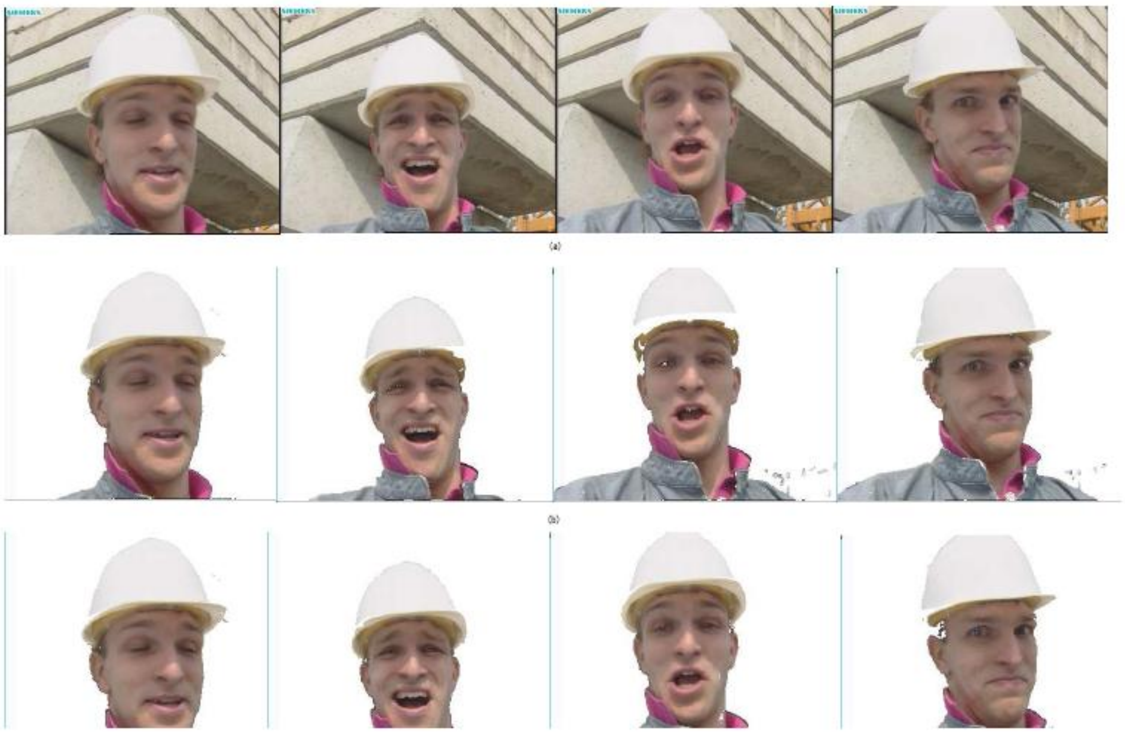

Fig. 9. MPEG-4 video sequence test: Foreman (a) original sequence, (b) segmentation result with 1 reference frame, (c) segmentation result with new reference frame every 10 frames

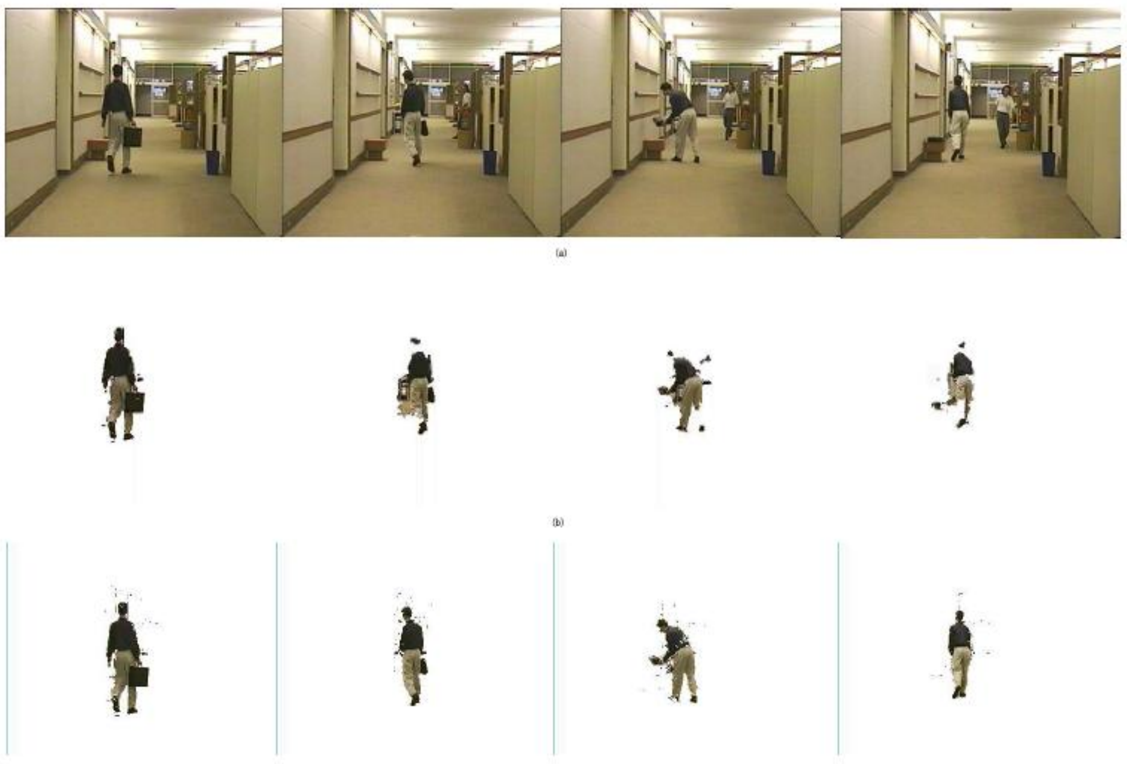

Fig. 10. Mpeg-4 video sequence test: Hall monitor (a) original sequence,(b) segmentation result with 1 reference frame, (c) segmentation result with new reference frame every 10 frames

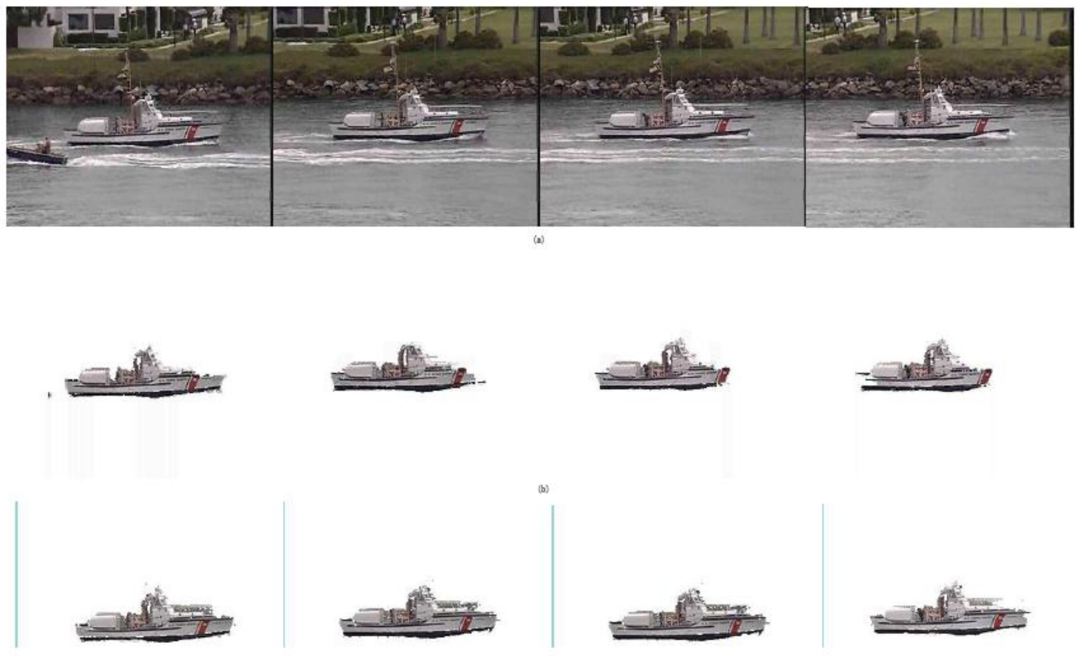

Fig. 11. Mpeg-4 video sequence test: Coast guard (a) original sequence,(b) segmentation result with 1 reference frame, (c) segmentation result with new reference frame every 10 frames

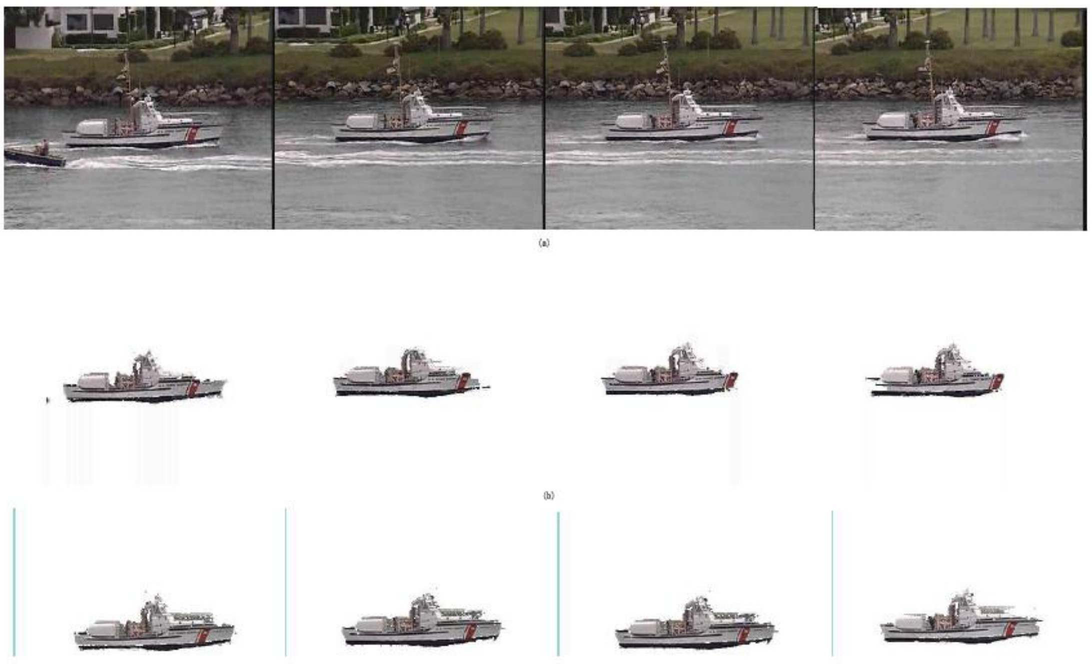

Fig. 11. Mpeg-4 video sequence test: Coast guard (a) original sequence,(b) segmentation result with 1 reference frame, (c) segmentation result with new reference frame every 10 frames