# Object-based Video Coding Using Pixel State Analysis

Takayuki Nishi
Department of Computer Science
Chubu University
Kasugai, Aichi 487-8501, JAPAN
tk_w@vision.cs.chubu.ac.jp

Hironobu Fujiyoshi
Department of Computer Science
Chubu University
Kasugai, Aichi 487-8501, JAPAN
hf@cs.chubu.ac.jp

## Abstract

*In archiving video for surveillance, frame-based coding has been used and it makes storage size large because the whole image is stored even if there is no object in the image. On the other hand, object-based coding has the capability to make storage size small, because it distinguishes between the foreground and the background regions of the image, and stores only foreground objects such as people. This paper describes object-based coding by pixel state analysis. In our method, pixel state analysis detects the foreground objects and background regions in video frames. Furthermore, it distinguishes foreground object pixels as* stationary *or* transient *pixels. For* stationary *pixels, it is possible to restore the color intensity by refering to the same pixel location in the last frame. Therefore, our method makes the storage size smaller. Additionally, the* transient *pixels of foreground objects are compressed using LZH codec. Since LZH codec uses lossless compression, the object region can be compressed with lesser loss in image quality. We have evaluated our system over 9 test sequences and obtained an improvement of 8% in compression ratio and better quality for the moving parts of the object region compared to MPEG-4.*

## 1. Introduction

Time-lapse methods has been used for video archiving in surveillance systems. It makes storage size large, because the whole image is stored by frame-based coding even if there is no object in the image. As a new technology for solving this problem, Clark[1] and Vetro et al.[2] have proposed object-based coding algorithms. In comparison with frame-based coding, object-based coding can choose to code important foreground objects such as people with higher quality than the other parts of the scene. In the first method [1], the algorithm called ObjectVideo automatically detects and segments moving foreground and background objects. The foreground video objects are coded into a MPEG-4 main profile compliant bit stream. In the latter method [2], a method similar to the ObjectVideo algorithm is used for object segmentation. This method consists of two processes; background subtraction part and motion vector analysis part. For removing the surviving false positive error due to lighting changes, swaying branches or leaves, and diffused reflection on the surface of water etc., motion vector analysis is used to extract moving objects. In both methods, foreground objects are compressed using MPEG-4 object-based encoding which is based on hybrid motion-compensated DCT coding.

To get higher compression and higher quality of video from a fixed camera, we present a video coding technique based on pixel state analysis. At first, pixel state analysis detects foreground objects and the background using object detection. Furthermore, pixels of foreground object are classified as *stationary* or *transient*. Even though the object is moving, the foreground object has stationary pixels because it has same texture and color. For stationary pixels, it is not neccessary to store the pixel intensity. It is possible to restore the pixel intensity of these stationary pixels by using their values from the last frame. For transient pixels, it is neccessary to store the pixel intensity. Transient pixels of a foreground object are compressed by LZH (Lempel-Ziv-Huffman) codec which guarantees to restore the pixel's intensity entirely.

In this paper, Section 2 introduces a new approach to object-based coding by pixel state analysis. Section 3 describes an algorithm for video encoding and decoding by our proposed method. Section 4 describes experimental results and presents a comparison of our method with MPEG-4.

## 2. Object-based Encoding by Pixel State Analysis

This section describes the concept of object-based encoding and presents the algorithm for pixel state analysis.

### 2.1. Image Representation with Foreground and Background

In surveillance video from a fixed camera, most pixels of the image belong to the background, where the pixel intensity is estimated with the use of a stationary background model. So, the whole image is segmented into background and foreground pixels determined by using background subtraction. This brings us to a new concept of object-based encoding illustrated in Figure 1.

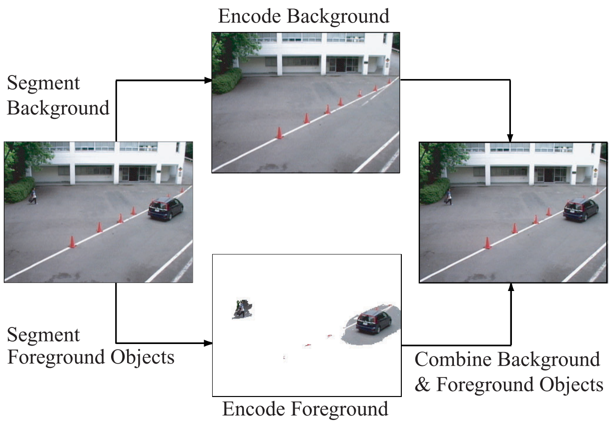



**Figure 1. Concept of object-based encoding**

When an object enters the surveillance area, object regions are classified as foreground and the other regions are classified as background. Foreground objects and background are encoded individually. At the stage of decoding, background and foreground objects are combined to restore the original image. To achieve higher compression based on object-encoding, we aim to determine the pixel state transition, assuming that a pixel has three states, viz., *background*, *transient* and *stationary* as illustrated in Figure 2. Stationary pixel which occures frequently when the object is stopped, is not needed to update its pixel intensity. By analyzing pixel state, higher compression based on object-encoding is expected.

### 2.2. Algorithm of Pixel State Analysis

To capture the transition of state in pixel intensity profiles, two factors are important: the existence of a significant step change in intensity, and the intensity value to which the profile stabilizes after passing through a period of instability[3]. To interpret the meaning of a step change (e.g. object passing through, stopping at, or leaving the pixel), we need to observe the intensity curve re-stabilizing after the step change. This introduces a time-delay into the process. In particular, current decisions are made about pixel events $k$ frames in the past. In our implementation $k$ is set to correspond to one second of video.

$$T = \max\left\{ |I_t - I_{(t-j)}|,\ \forall j \in [1,5] \right\} \tag{1}$$

The stability measure is the variance of the intensity profile from time $t$ to the present:

$$S = \frac{k\sum_{j=0}^{k} {I_{(t+j)}}^2 - (\sum_{j=0}^{k} I_{(t+j)})^2}{k(k-1)} \tag{2}$$

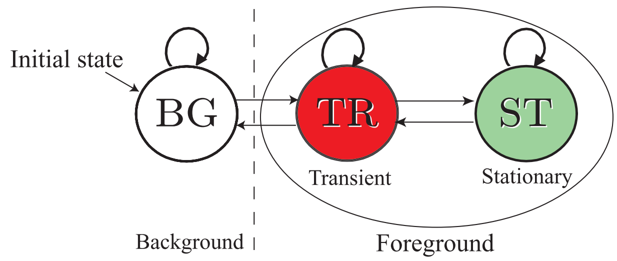



**Figure 2. State transition diagram of the pixel**

At this point a transience map of pixel state $M$ can be defined for each pixel by the following algorithm, where the pixel state can take three possible values: background=BG; transient=TR and stationary=ST. Background intensity is prepared in advance as a background image.

```
While all the pixels have not been classified
When M is equal to ST or BG and T is higher than threshold
    assign TR to M
When M is equal to TR and S is lower than threshold
    If the pixel intensity is nearly background
        assign BG to M
    else
        assign ST to M
end
```

Background is updated by an Infinite Impulse Response (running average) filter to accommodate slow lighting changes and noise in the imagery, as well as to compute statistically significant step-change thresholds [4].

$$B(t) = \alpha I(t) + (1 - \alpha)B(t - 1) \tag{3}$$

The constant $\alpha$ determines how fast the background is allowed to change.

Figure 3 shows a sample of pixel state analysis for a stopped object and a moving object. We see that the moving object consists of a lot of transient pixels and a few stationary pixels.

## 3. Object-based Encoding and Decoding

This section describes the process of object-based encoding and decoding using pixel state analysis.

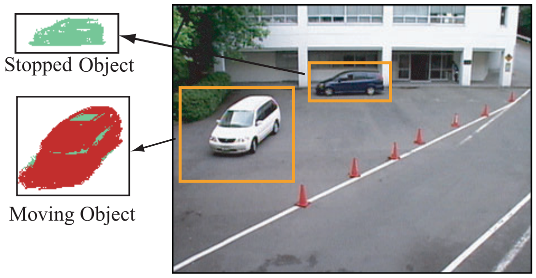
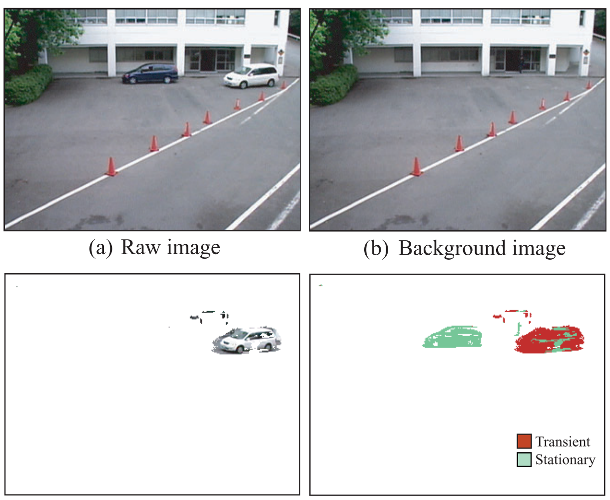



**Figure 3. Example of pixel state analysis**

### 3.1. Encoding Process

Each frame of video is decomposed by pixel state analysis into three elements: background image, pixel state map, and intensity of transient pixels as shown in Figure 4. Each element is encoded by the following algorithm.




(a) Raw image (b) Background image

(c) Color intensity of transient pixels (d) Pixel state map

**Figure 4. Three elements decomposed by pixel state analysis**

**Background image** The first frame of video is set as an initial background image when there is no object in the scene. The background image is compressed by LZH (Lempel-Ziv-Huffman)[5].

**Pixel state map** Pixel state analysis outputs a transience map of pixel state, which indicates the state of (BG/ST/TR). This pixel state can be represented by 2 bits at each pixel. A chain of these pixel states is encoded by the run length method shown in Figure 5. Additionally, the compressed data obtained by using run length method is encoded by LZH.

**Intensity of transient pixel color** The intensity information corresponding to the transient pixels is compressed by LZH. Since LZH is lossless compression, it is possible to restore the pixel intensity of transient pixels which mainly belong to the moving part of the foreground object.

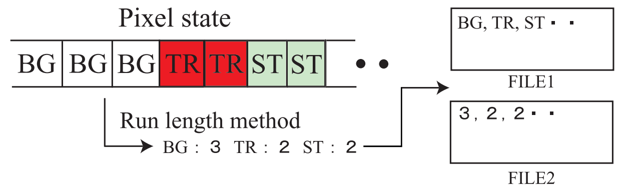



**Figure 5. Applying run length method to pixel state map**

### 3.2. Decoding Process

For restoring an image, the decoder overlays the value of the color intensity of transient pixels over the background image by refering to the pixel state map $M$. For stationary pixels, the decoder overlays the color intensity of pixel over the background image by refering to its value in the restored last frame. To adapt to ambient illumination change, the background image is updated every 8 seconds in the same way as MPEG-4's keyframe.

(a) Original image (b) Restored image

**Figure 6. Sample of original and restored image by proposed method**

## 4. Experimental Results

Test sequences for evaluation were recorded in a couple of different outdoor surveillance sites. The frame resolution is $320 \times 240$ with RGB color (24bits). We evaluate our method in aspects of compression rate and image quality comparing MPEG-4.

### 4.1. Evaluation in Compression Rate

Table 1 shows the results achieved by using our proposed method in terms of the compressed file size and the percentage of bit-saving against the values obtained by MPEG-4 for 9 test sequences of varying duration and content. It is clear that our proposed method has 8% bit-saving ratio higher

**Table 1. Compressed file size**

| Seq. | Original [MB] | MPEG* [MB] | Proposed method [MB] | Bit-saving ratio [%] |
|---|---|---|---|---|
| A | 184 | 5.30 | 3.46 | 34.72 |
| B | 169 | 4.81 | 4.67 | 2.91 |
| C | 150 | 5.05 | 6.29 | -24.55 |
| D | 429 | 12.57 | 9.43 | 24.98 |
| E | 468 | 15.29 | 13.47 | 11.90 |
| F | 521 | 16.83 | 15.03 | 10.70 |
| G | 1,930 | 83.70 | 68.39 | 18.29 |
| H | 3,860 | 104.49 | 94.49 | 9.57 |
| I | 3,860 | 83.93 | 96.86 | -15.41 |
| Ave. | 1,286 | 36.89 | 34.68 | 8.12 |

*MPEG-4 configuration
Key Frame: 8 sec/frame, Compression Control: 75
Bit Rate : Variable Bit Rate(Max:3,000kbps)

**Table 2. PSNR of restored image**

| Seq. | MPEG[dB] | | | Proposed Method[dB] | | |
|---|---|---|---|---|---|---|
| | whole image | object region | moving pixels | whole image | object region | moving pixels |
| A | 34.2 | 32.0 | 31.7 | 32.9 | 32.7 | 36.2 |
| B | 34.3 | 32.8 | 31.9 | 32.8 | 33.2 | 36.6 |
| C | 34.3 | 32.5 | 31.8 | 33.2 | 33.6 | 37.2 |
| D | 34.2 | 33.0 | 32.6 | 33.3 | 33.7 | 37.4 |
| E | 34.2 | 32.5 | 32.0 | 33.2 | 32.9 | 36.3 |
| F | 34.2 | 32.5 | 31.6 | 33.2 | 33.1 | 36.6 |
| G | 33.9 | 32.1 | 31.1 | 32.3 | 33.6 | 37.5 |
| H | 34.9 | 33.3 | 33.0 | 33.8 | 34.0 | 37.9 |
| I | 35.7 | 33.8 | 33.5 | 34.9 | 32.4 | 36.1 |
| Ave. | 34.4 | 32.7 | 32.1 | 33.3 | 33.2 | 36.9 |

than MPEG-4. We also can see that it does however vary as a function of the sequence from as much as -24.6% with Seq. C to as much as 34.7% with Seq. A. These differences are mainly due to the amount of moving objects. Test sequence of Seq. C contains many moving objects with small area. Therefore, the file size will be large in order to store the color intensities of transient pixels belonging to moving objects.

### 4.2. Evaluation in Image Quality

In the evaluation of image quality, we calculate PSNR (Peak Signal to Noise Ratio) of restored image by MPEG-4 and proposed method. PSNR is calculated as a logarithm of MSE (Mean Squared Error) as follows;

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\mathrm{MSE}}[\mathrm{dB}]$$

where

$$\mathrm{MSE} = \sqrt{\frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}(f(i,j)-f'(i,j))^2}{WH}}.$$

$W$: image width $H$: image height
$f(i,j)$: original image $f'(i,j)$: restored image

Table 2 shows PSNR for the whole image, object regions and moving pixels of the object regions. We see that PSNR of MPEG-4 and proposed method have almost same performance at whole image and object regions. However, the average of PSNR for moving pixels in object region is 36.9 [dB], so the proposed method can restore color intensity of moving pixels with better quality compared to MPEG-4. Since human eyes focus on moving objects, getting better quality on the moving parts of the object region is effective for archiving video for surveillance.

## 5. Conclusion

We have presented an object-based encoding method using pixel state analysis for long-term video archiving. We showed that the proposed method has a bit saving ratio 8% higher than MPEG-4 and almost same image quality for restored whole image. Focusing only on the image quality of transient pixels (mainly belonging to moving parts of the object), we obtain better performance as we use LZH lossless compression.

Future work include indexing and extraction of key frame for object summarization of long-term video using the output of pixel state analysis.

## References


[1] J. Clark, "The Power of ObjectVideo", http://www.objectvideo.com/

[2] A. Vetro, T. Haga, K. Sumi and H. Sun, "Object-based Coding for Long-term Archive of Surveillance Video", *IEEE International Conference on Multimedia and Expo*, Jul, 2003.

[3] H. Fujiyoshi and T. Kanade, "Layered Detection for Multiple Overlapping Objects", *IEEE International Conference on Pattern Recognition*, vol. 4, pp.156-161, 2002.

[4] A. Lipton, H. Fujiyoshi and R. Patil, "Moving Target Classification and Tracking From Real-time Video" *Proc. of the Workshop on Application of Computer Vision, IEEE*, Oct, 1998, pp. 8-14.

[5] M. Nelson and J. Gailly, "The Data Compression Book, Second Edition", M&T Books, 1996.