

Role of Edge Detection in Video Semantics

Michael Lee, Surya Nepal, Uma Srinivasan
CSIRO Mathematical and Information Sciences
Locked Bag 17, North Ryde NSW 1670, Australia

{Michael.Lee, Surya.Nepal, Uma.Srinivasan}@csiro.au

Abstract

The *semantic gap* or *semantic chasm* is a well-known problem in content-based image and video retrieval. To address this problem, many techniques have been proposed in the literature. A more common approach is the use of low-level features such as colour, texture and shape for semantic analysis. Our focus in this paper is on the *edge* feature, which has not been exploited to the same extent as other low-level features for semantic analysis. In this paper, we present an algorithm for edge detection, and illustrate the usage of edges for semantic analysis of video content.

We first propose an algorithm for detecting edges within video frames directly on the MPEG format without a decompression process. The algorithm is based on a spatial-domain synthetic edge model, which is defined using interrelationship of two DCT edge features: horizontal and vertical. We use a multi-step approach to classify video sequences into meaningful semantic segments such as “goal”, “foul”, and “crowd” in basketball games using the “edgeness” criteria. We then show how an audio feature (“whistle”) can be used as a filter to enhance edge-based semantic classification for sports videos.

Keywords: Edge detection, video semantic analysis, MPEG video, DCT model.

1 Introduction

The amount of digital video available has increased dramatically in the last few years, and has been used in a wide range of multimedia application areas such as digital video archives, Video-on-Demand (VoD) systems, etc. Efficient and effective retrieval from large collections of videos forms an important component of such systems resulting in a need for content-based video retrieval. Current content management systems support retrieval using low-level features, such as motion, colour and texture. However, low-level features often have little meaning for the human users of these systems, who much prefer to identify content using high-level semantics. This creates a gap between the system and the user that must be bridged for these systems to be used effectively. This gap is referred as *semantic gap* or *semantic chasm*. Unlike most content-based video retrieval systems that use features such as colour, motion and texture, the research presented in this paper uses the edge feature to

identify semantic content. We first propose an efficient algorithm that uses a spatial-domain synthetic edge model to detect the edge feature, and then show how this feature is useful for semantic analysis in the sports domain.

Conventional processing of digital video requires the video to be de-compressed which is an additional overhead. To avoid this problem, we operate directly on the compressed data. MPEG offers an attractive low-cost possibility for the storage and transmission of digital video data as it uses the motion-based compression algorithms. In this paper, we present a fast algorithm for detecting edges of an object directly in MPEG compressed video. The algorithm uses both horizontal and vertical edge features derived from the DCT block within a frame.

In content-based video retrieval systems, edge information can be used as a low level feature for indexing and retrieval. For example, edge information has been used in face recognition [1] applications. The advantage of using such low-level features is that they are applicable to generic video data. However, for these to be useful, it is important to have techniques for automatic classification of video sequences to represent high-level semantics. This requires explicit domain knowledge of the application as well as the genre of the video.

There have been some recent attempts to exploit domain knowledge and inherent properties of low-level features for automatic detection of high-level concepts in MPEG sports videos. Most of them attempt to relate or map the low-level information measured from video data to high-level concepts [2]-[4]. Yoshitaka et al. [2] use spatio-temporal correlations of objects to detect a certain semantic content, which are commonly performed by soccer players, namely “wall pass”, “overlap”, “though pass”, and “zone press”. Saur et al. [3] present a method that use the low-level information available directly from MPEG compressed video of basketball as well as prior knowledge of basketball video structure to provide high-level content analysis like “close-up views”, “fast breaks”, “steals”, etc. Nepal and Srinivasan [4] present temporal models for detecting goal segments in basketball videos. These approaches use motion-based visual features such as pan and zoom along with other audio-visual features. In all the above cases context, edge information has not been exploited for semantic identification and classification. In this paper, we present how edge information, directly extracted from MPEG domain using our proposed algorithm, could be used to classify video sequences into high-level semantic categories.

The contributions of the paper can be summarized as follows:

Copyright ©2003, Australian Computer Society, Inc. This paper appeared at the Pan-Sydney Area Workshop on Visual Information Processing (VIP2002), Sydney, Australia. Conferences in Research and Practice in Information Technology, Vol. 22. J. S. Jin, P. Eades, D. D. Feng, H. Yan, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

- We propose an algorithm for edge detection directly from MPEG videos using pre-defined synthetic edge models. This is described in section 2.
- In section 3, using detected edge information, we propose a technique of classifying video into sequences, representing high-level semantics.

2 Edge Detection Algorithm

At present, the majority of researches use Canny algorithm [5] and Marr-Hildreth algorithm [6] for edge detection in the spatial domain. Canny algorithm is based on the gradient vectors of the image. Algorithms based on gradient vectors are the most popular in edge detection. Marr-Hildreth algorithm is based on the analysis of the second derivative. Other known edge detection algorithms that can be found in literature are works by Davis [7] and Haralick [8]. Note that these algorithms are defined for images and are not directly applicable to frames in compressed videos. Our algorithm is motivated by the result of [9] and is based on pre-defined synthetic edge models [10]. We first give a brief overview of MPEG video and then describe our edge detection algorithm for MPEG videos.

MPEG video consists of three basic frame types: I- (Intra-coded), P- (Predictive coded), and B- (Bi-directional predictive coded) frames. Each I-frame is divided into 16x16 macroblocks (MBs) and each MB consists of four 8x8 luminance (Y) blocks and two 8x8 chrominance (Cb and Cr) blocks. Each block is transformed into a DCT block, which consists of one DC and 63 AC coefficients. A P-frame is predictively coded with a past reference frame while a B-frame requires a future and past frames together for its prediction [11]. In this proposed algorithm, we use the AC coefficients of Y blocks in I-frames to detect edges.

Since the duration of a scene in which the object is present is usually longer than the duration of a group of pictures (GOP), which consists of 12 to 15 frames, we use GOP as the basic unit for analysis.

2.1 DCT Edge Features

We first define the two edge features and our spatial edge models; and then show how these two edge features are used to derive edges defined in the edge models. The horizontal and vertical edge features can be distinctly formed by the two-dimensional DCT of a block. As shown in Fig. 1, the edges in an 8x8 block can be represented by two edge feature sets [12]:

$$\begin{aligned} \text{Horizontal feature: } \mathbf{H} &= \{H_i; i = 1, 2, \dots, 7\} \\ \text{Vertical feature: } \mathbf{V} &= \{V_j; j = 1, 2, \dots, 7\} \end{aligned} \quad (1)$$

where H_i and V_j correspond to the DCT coefficients $F_{u,0}$ and $F_{0,v}$, for $u, v = 1, 2, \dots, 7$, in Eq. (2), which describes the 2-dimensional DCT:

$$F_{0,0} = \frac{1}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} \quad (2a)$$

$$F_{u,v} = \frac{2}{\sqrt{MN}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} \cos \frac{(2i+1)u\pi}{2M} \cos \frac{(2j+1)v\pi}{2N} \quad (2b)$$

where $u = 1, 2, \dots, M-1$, and $v = 1, 2, \dots, N-1$. For an 8x8 block, $M = N = 8$. Eqs. (2a) and (2b) describe DC and AC coefficients of DCT, respectively.

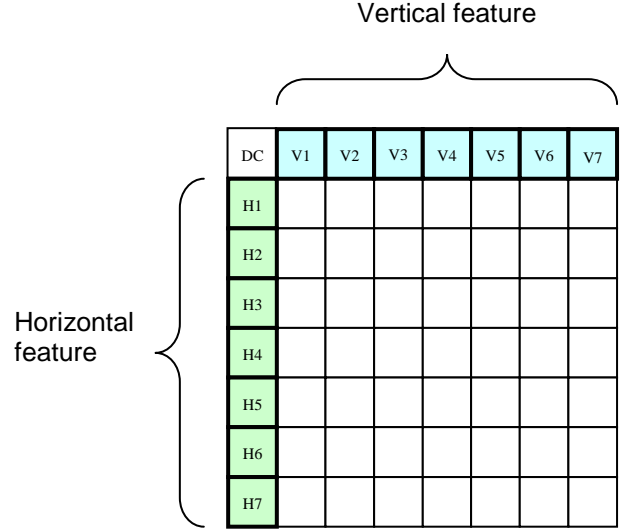


Fig. 1. Horizontal and vertical edge features of DCT coefficients in an 8x8 block.

From Eq. (2), the DCT of the 8x8 image vector is given by

$$\mathbf{F} = \begin{bmatrix} F_{0,0} & F_{0,1} & F_{0,2} & F_{0,3} & F_{0,4} & F_{0,5} & F_{0,6} & F_{0,7} \\ F_{1,0} & F_{1,1} & F_{1,2} & F_{1,3} & F_{1,4} & F_{1,5} & F_{1,6} & F_{1,7} \\ F_{2,0} & F_{2,1} & F_{2,2} & F_{2,3} & F_{2,4} & F_{2,5} & F_{2,6} & F_{2,7} \\ F_{3,0} & F_{3,1} & F_{3,2} & F_{3,3} & F_{3,4} & F_{3,5} & F_{3,6} & F_{3,7} \\ F_{4,0} & F_{4,1} & F_{4,2} & F_{4,3} & F_{4,4} & F_{4,5} & F_{4,6} & F_{4,7} \\ F_{5,0} & F_{5,1} & F_{5,2} & F_{5,3} & F_{5,4} & F_{5,5} & F_{5,6} & F_{5,7} \\ F_{6,0} & F_{6,1} & F_{6,2} & F_{6,3} & F_{6,4} & F_{6,5} & F_{6,6} & F_{6,7} \\ F_{7,0} & F_{7,1} & F_{7,2} & F_{7,3} & F_{7,4} & F_{7,5} & F_{7,6} & F_{7,7} \end{bmatrix}$$

It is well-known that the DCT has superior energy compaction property, and particularly, spatial edge information can be well represented by a certain group of DCT coefficients. In the DCT domain, the edge pattern of a block can be characterized with only one edge component, which is represented by projecting components in the vertical and horizontal directions, respectively. We can readily see these edge features from the DCT basis images in Fig. 2.

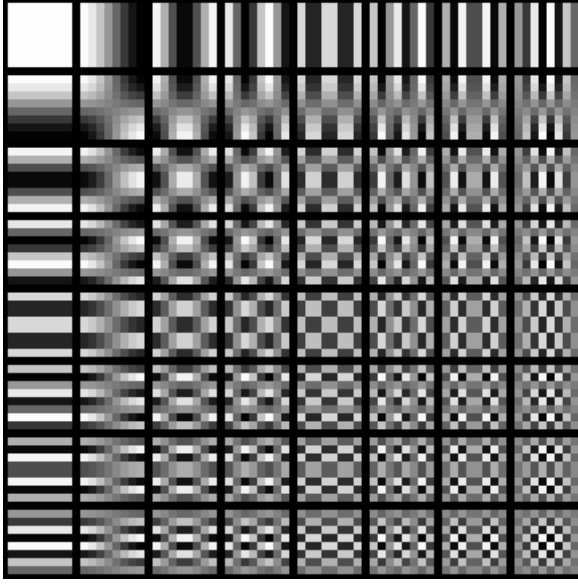


Fig. 2. DCT basis images for the 8x8 block size (intensity: -1.0 (black) ~ +1.0 (white)).

For these DCT basis images, the two-dimensional DCT kernel for the 8x8 block is defined as

$$g_{i,j,u,v} = \cos \frac{(2i+1)u\pi}{16} \cos \frac{(2j+1)v\pi}{16} \quad (3)$$

for $i, j, u, v = 0, 1, \dots, 7$. The basis images are pseudo-pictures for the two-dimensional DCT coefficients $F_{u,v}$, where each coefficient corresponds to the intensity transition across its respective basis image.

Using the horizontal and vertical edge features, we have defined typical spatial-domain synthetic edge models, shown in Fig. 3.

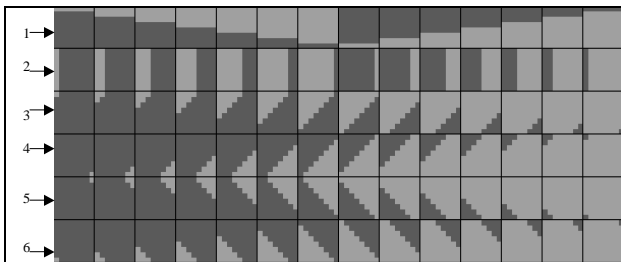


Fig. 3. Spatial-domain synthetic edge models for the 8x8 block size (Gray levels: 160 for bright side (HIGH), 90 for dark side (LOW); from the top line: Horizontal, vertical, 45-degree diagonal (HIGH-LOW), 45-degree diagonal (LOW-HIGH), 135-degree diagonal (HIGH-LOW), and 135-degree diagonal (LOW-HIGH) edges).

A total of 84 edge models are defined: 14 horizontal, 14 vertical, 28 45-degree diagonal, and 28 135-degree diagonal. We have used these models to analyse their corresponding DCT coefficients whose values are given by Eq. (2).

Fig. 4 shows DCT coefficients of the first column (1-6) of edge models shown in Fig. 3. Only those DCT coefficients of Fig. 4 that correspond to horizontal and vertical edge features in Eq. (1) are used in our edge detection algorithm. For the sake of clarity, in Fig. 4, we have marked the DCT coefficients that correspond to vertical edge features for the edge model 3. It is clearly seen that horizontal and vertical edges correspond to only the horizontal (H_i) and vertical (V_j) features (Refer edge models 1 and 2 in Fig. 4). The balanced coefficient values appear in both horizontal and vertical features for a diagonal edge (45-degree or 135-degree: refer edge models 3, 4, 5 and 6 in Fig. 4). A complete table of the corresponding DCT coefficients is given in [12].

| | | | | | | | |
|---|--------|--------|--------|--------|-------|-------|-------|
| <1: HORIZONTAL HIGH-LOW> | | | | | | | |
| 790.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 137.31 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 129.34 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 116.41 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 98.99 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 77.78 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 53.58 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| 27.31 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 | -0.00 | 0.00 |
| <2: VERTICAL HIGH-LOW> | | | | | | | |
| 790.00 | 137.31 | 129.34 | 116.41 | 98.99 | 77.78 | 53.58 | 27.31 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| <3: 45-D DIAGONAL HIGH-LOW> | | | | | | | |
| 728.75 | 17.16 | 16.17 | 14.55 | 12.37 | 9.72 | 6.70 | 3.41 |
| 17.16 | 16.83 | 15.86 | 14.27 | 12.14 | 9.54 | 6.57 | 3.35 |
| 16.17 | 15.86 | 14.94 | 13.44 | 11.43 | 8.98 | 6.19 | 3.15 |
| 14.55 | 14.27 | 13.44 | 12.10 | 10.29 | 8.08 | 5.57 | 2.84 |
| 12.37 | 12.14 | 11.43 | 10.29 | 8.75 | 6.87 | 4.74 | 2.41 |
| 9.72 | 9.54 | 8.98 | 8.08 | 6.87 | 5.40 | 3.72 | 1.90 |
| 6.70 | 6.57 | 6.19 | 5.57 | 4.74 | 3.72 | 2.56 | 1.31 |
| 3.41 | 3.35 | 3.15 | 2.84 | 2.41 | 1.90 | 1.31 | 0.67 |
| <4: 45-D DIAGONAL LOW-HIGH> | | | | | | | |
| 728.75 | -17.16 | 16.17 | -14.55 | 12.37 | -9.72 | 6.70 | -3.41 |
| -17.16 | 16.83 | -15.86 | 14.27 | -12.14 | 9.54 | -6.57 | 3.35 |
| 16.17 | -15.86 | 14.94 | -13.44 | 11.43 | -8.98 | 6.19 | -3.15 |
| -14.55 | 14.27 | -13.44 | 12.10 | -10.29 | 8.08 | -5.57 | 2.84 |
| 12.37 | -12.14 | 11.43 | -10.29 | 8.75 | -6.87 | 4.74 | -2.41 |
| -9.72 | 9.54 | -8.98 | 8.08 | -6.87 | 5.40 | -3.72 | 1.90 |
| 6.70 | -6.57 | 6.19 | -5.57 | 4.74 | -3.72 | 2.56 | -1.31 |
| -3.41 | 3.35 | -3.15 | 2.84 | -2.41 | 1.90 | -1.31 | 0.67 |
| <5: 135-D DIAGONAL HIGH-LOW> | | | | | | | |
| 728.75 | -17.16 | 16.17 | -14.55 | 12.37 | -9.72 | 6.70 | -3.41 |
| 17.16 | -16.83 | 15.86 | -14.27 | 12.14 | -9.54 | 6.57 | -3.35 |
| 16.17 | -15.86 | 14.94 | -13.44 | 11.43 | -8.98 | 6.19 | -3.15 |
| 14.55 | -14.27 | 13.44 | -12.10 | 10.29 | -8.08 | 5.57 | -2.84 |
| 12.37 | -12.14 | 11.43 | -10.29 | 8.75 | -6.87 | 4.74 | -2.41 |
| 9.72 | -9.54 | 8.98 | -8.08 | 6.87 | -5.40 | 3.72 | -1.90 |
| 6.70 | -6.57 | 6.19 | -5.57 | 4.74 | -3.72 | 2.56 | -1.31 |
| 3.41 | -3.35 | 3.15 | -2.84 | 2.41 | -1.90 | 1.31 | -0.67 |
| <6: 135-D DIAGONAL LOW-HIGH> | | | | | | | |
| 728.75 | 17.16 | 16.17 | 14.55 | 12.37 | 9.72 | 6.70 | 3.41 |
| -17.16 | 16.83 | -15.86 | -14.27 | -12.14 | -9.54 | -6.57 | -3.35 |
| 16.17 | 15.86 | 14.94 | 13.44 | 11.43 | 8.98 | 6.19 | 3.15 |
| -14.55 | -14.27 | -13.44 | -12.10 | -10.29 | -8.08 | -5.57 | -2.84 |
| 12.37 | 12.14 | 11.43 | 10.29 | 8.75 | 6.87 | 4.74 | 2.41 |
| -9.72 | -9.54 | -8.98 | -8.08 | -6.87 | -5.40 | -3.72 | -1.90 |
| 6.70 | 6.57 | 6.19 | 5.57 | 4.74 | 3.72 | 2.56 | 1.31 |
| -3.41 | -3.35 | -3.15 | -2.84 | -2.41 | -1.90 | -1.31 | -0.67 |

Fig. 4. Examples of the corresponding DCT coefficients for the first column (1-6) of the spatial-domain edge models in Fig. 3.

In order to determine edge features defined in our models, the following tests are performed on horizontal and vertical features:

$$\sum_{i=1}^7 |H_i| \geq Thr_horizontal \quad (3a)$$

$$\sum_{j=1}^7 |V_j| \geq Thr_vertical \quad (3b)$$

If the tests in Eqs. (3a) and (3b) are “true” and “false”, respectively, it is defined that the block contains a vertical edge. For a horizontal edge in the block, the converse is true, *i.e.* the tests have to be “false” and “true”, respectively. If both tests are “true”, the block contains a diagonal edge and it is further tested to determine its orientation using the polarities of the first coefficients: H_1 and V_1 .

The orientation of the diagonal edge is determined by the polarities of the first coefficients: V_1 and H_1 . That is, the coefficients have the same polarities ($V_1 \& H_1 = positive$, or $V_1 \& H_1 = negative$) for a 45-degree diagonal edge, and different polarities ($V_1 = positive$ and $H_1 = negative$, or $V_1 = negative$ and $H_1 = positive$) for a 135-degree diagonal edge.

2.2 Edge Orientation

In the viewpoint of edge angle, every edge can fall into one of the four edge groups, as shown in Fig. 5, which are determined by horizontal and vertical edge features in Fig. 1. That is, if $\sum |H_i| \leq \sum |V_j|$ and $\sum |H_i| \geq \sum |V_j|$, then the edges are vertical-dominant and horizontal-dominant, respectively.

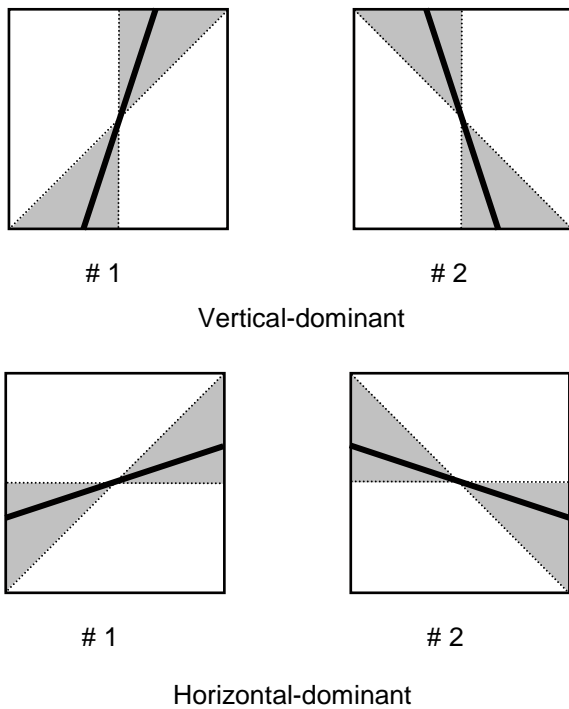


Fig. 5. Vertical- and horizontal-dominant edge groups (shaded areas indicate the ranges of edge angle).

The group 1 of both dominant edges can be roughly classified into the 45-degree diagonal edge group in terms of their gradients. Therefore, the edges of group 1 are determined when two first coefficients have the same polarities ($V_1 \& H_1 = positive$, or $V_1 \& H_1 = negative$), as mentioned in the previous section. On the other hand, the edges of group 2, classified into the 135-degree diagonal edge group, are also determined when the polarities of the two first coefficients are different ($V_1 = positive$ and $H_1 = negative$, or $V_1 = negative$ and $H_1 = positive$).

In fact, the expression of edge angle is limited by the block size of 8×8 . A maximum of 4 edge angles can be depicted for each edge group. Fig. 6 shows the 4 edge angles (θ_1 , θ_2 , θ_3 , and θ_4) defined in the group 1 of vertical-dominant edges. Each edge angle may represent a certain range of angle.

In this experiment, the angle ranges are given as follows:

$$\begin{aligned} 0^\circ &< \theta_1 \leq 15^\circ \\ 15^\circ &< \theta_2 \leq 27^\circ \\ 27^\circ &< \theta_3 \leq 37^\circ \\ 37^\circ &< \theta_4 \leq 45^\circ \end{aligned}$$

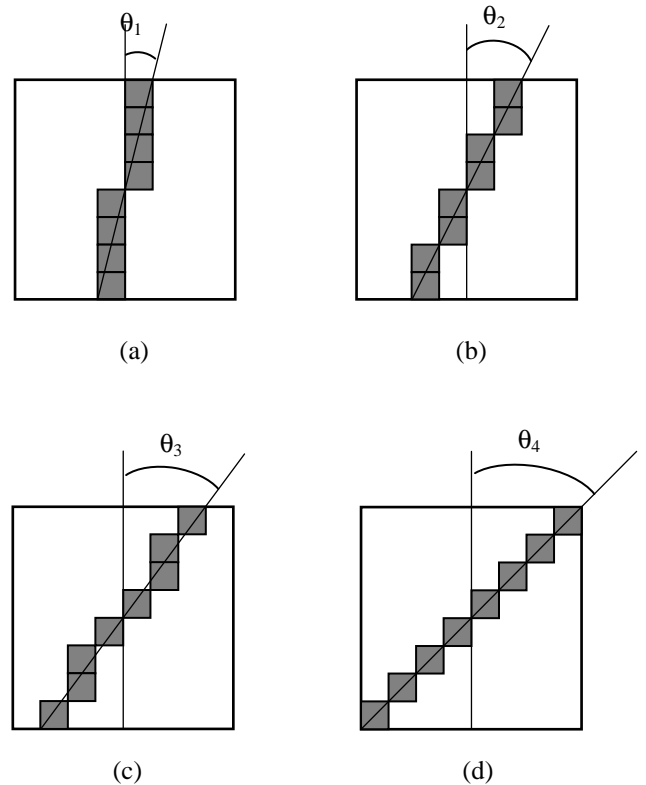


Fig. 6. Four edge angles defined in the group 1 of vertical-dominant edges. The actual values of the edge angles are: (a) $\theta_1 = \tan^{-1}(1/4) = 14.0^\circ$, (b) $\theta_2 = 26.5^\circ$, (c) $\theta_3 = 36.8^\circ$, (d) $\theta_4 = 45^\circ$.

Based on both vertical and horizontal edge features, we can derive a generic equation to generate an edge with a certain angle. In Fig. 7, the ratio of x (horizontal axis) and y (vertical axis) can be given from the two DCT edge features as

$$\tan \theta = \left(\sum_{i=1}^7 |H_i| \right) / \left(\sum_{j=1}^7 |V_j| \right) = \frac{x}{y} \quad (4)$$

From Eq. (4), an angle parameter is obtained as follows:

$$\begin{aligned} \varphi &= (\text{int})((\tan \theta + \alpha) / 0.25) * 2 - 1; \\ \text{if } (\varphi < 0) \varphi &= 0; \end{aligned}$$

where α is a compensation constant and is give as $\alpha < 0.25$. The values of angle parameter are computed as $\varphi = 0, 1, 3, 5, 7$.

Finally, x can be expressed as an equation of φ and y :

$$x = 7 - (\varphi * 10 * y / 7 / 10) + (\varphi * 10 * y / 7 \% 10 / 5) + (7 - \varphi) / 2; \quad (5)$$

where $y = 0, 1, \dots, 7$, and x is also given as an integer in the range, $0 \leq x \leq 7$.

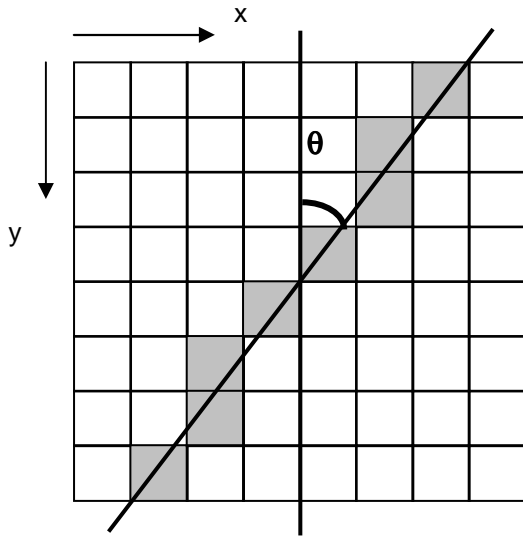


Fig. 7. Generating an edge with an angle θ in an 8x8 block.

2.3 Edge Location

As shown in Fig. 5, 7 locations can be defined for a vertical or horizontal edge while a diagonal edge can have up to 14 locations in an 8x8 block. A simple method to accurately find an edge location is defined as follows:

Let

$$L(V) = |V_1| / (|V_1| + |V_2|) \quad \text{for vertical-dominant edges,} \quad (6a)$$

$$L(H) = |H_1| / (|H_1| + |H_2|) \quad \text{for horizontal-dominant edges.} \quad (6b)$$

It is found that $L(V)$ or $L(H)$ varies in the range of 0.51 to 1.0, no matter what type of edge it is. That is, the maximum value 1.0 corresponds to the middle edge location while 0.51 corresponds to the first and last edge locations in a block. It is interesting to note that the second coefficient of the edge feature, V_2 or H_2 , is always “positive” for the first half and “negative” for the second half of the edge location group, respectively.

Fig. 8 shows an example of edge offset. The maximum offset range is defined ± 6 pixels for a 45-degree diagonal edge.

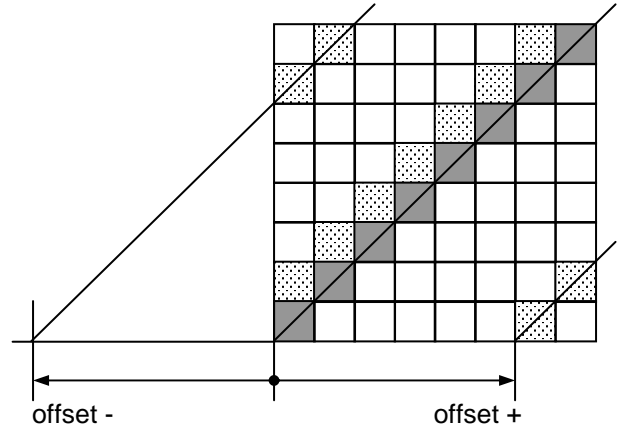


Fig. 8. Edge offset.

The edge offset for vertical-dominant edges is given as follows:

$$\text{Offset}(V) = (\text{int})((L(V) - 0.51) / \beta * (\tan \theta + 1.0)); \quad (7)$$

where β is given as $(1.0 - 0.51) / 3.0 = 0.163$ from the maximum and minimum values of $L(V)$ in Eq. (6a). From Eqs. (5) and (7), we can derive the following equation which depicts a vertical-dominant edge with an angle and location within the 8x8 block size:

$$\begin{aligned} &\text{if } (x \geq 0 \ \&\& \ x \leq 7) \\ &\{ \\ &\quad x = 7 - (\varphi * 10 * y / 7 / 10) + (\varphi * 10 * y / 7 \% 10 / 5) + \\ &\quad \quad (7 - \varphi) / 2 + \text{Offset}(V); \\ &\} \end{aligned}$$

In the same way, we can also obtain an equation for horizontal-dominant edges as follows:

```

if (y ≥ 0 && y ≤ 7)
{
  y = 7 - (φ * 10 * x / 7 / 10) + (φ * 10 * x / 7 % 10 / 5) +
    (7 - φ) / 2 + Offset(H);
}

```

where $\text{Offset}(H) = (\text{int})((L(H) - 0.51) / \beta * (\tan\theta + 1.0))$;

Having determined the edge location and orientation, we next describe some results with experimental data.

2.4 Edge Detection Results

We have tested the edge detection algorithm on different types of MPEG video sequences. Fig. 9 shows some examples.

The original image in Fig 9(a) shows an object (a face in this case) that has clear boundary lines, and Fig. 9(b) shows the edges detected. As can be seen, the edges are relatively well matched with the original. This shows that a face feature can be defined by a group of detected edges. In combination with a skin colour region detection technique [13], face regions can be more accurately detected, as the overlap of skin colour and face feature regions is expected to be a face region. However, we are not dealing with skin colour detection here.

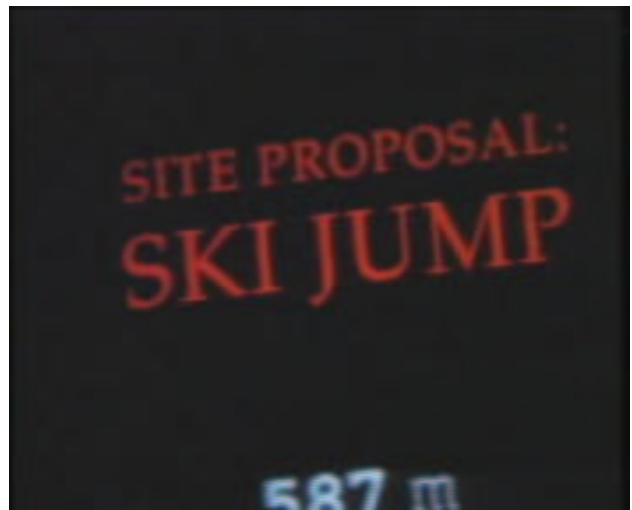
Fig. 9(d) shows a possibility that the edge detection technique can be extended to text detection. Some detected characters in Fig. 9(d) are readily recognizable although the detection tends to be limited to only larger sized characters. It is possible that this character detection can be enhanced by associating with the text detection algorithm that has been developed by Gu [14] for detecting the character regions directly in MPEG video sequences.



(a)



(b)



(c)



(d)

Fig. 9. (a) & (c) Original images; (b) & (d) Edge detection results of images (a) & (c), respectively (Image (a) is the frame 15 of a test video known as 'susie.mpg', of which the copyright is owned by David Sarnoff Research Center, Inc. and can be downloaded from <ftp://ftp.tek.com/tv/test/streams/Element/>. The usage of the video is granted for research and demonstration purposes).

Fig. 10 shows another example of edge detection on an MPEG video.

Next, we move on to the use of the detected edge features to determine higher-level semantics. For this, we define “edgeness” criteria.



(a)



(b)

Fig. 10. Edge detection in the MPEG domain: (a) Original I-frame of a basketball video, (b) Detected edges on the frame (© Basketball Association Australia).

3 Semantic Analysis

Our semantic analysis is based on the edge complexity, which we call “edgeness” that is defined as the number of edge pixels (the number of pixels of edges) in the frame. We then define a semantic s on a video sequence as a function of edgeness and duration as follows.

$$s = f(\text{edgeness}, \text{duration})$$

where duration is the number of frames where the edgeness remains within a certain threshold.

We develop a two-step approach of classifying video sequences into meaningful segments. The first step is a coarse-level semantic classification based on “edgeness”. The second step, fine-level semantic classification, is a further refinement of results from first step based on duration.

Let

e_i = edgeness of the i^{th} I-frame,

N = number of I frames in the video,

$$\mu = \text{mean edgeness over the video} = \frac{\sum_{i=1}^N e_i}{N},$$

σ_{avg} = average deviation from the mean

$$= \frac{\sum_{i=1}^N \text{abs}(e_i - \mu)}{N}.$$

We first define three coarse-level semantics of frames as follows.

High : if $e_i > \mu + \sigma_{avg}$

Medium : if $\mu + \sigma_{avg} \geq e_i \geq \mu - \sigma_{avg}$

Low : if $e_i < \mu - \sigma_{avg}$

Fig. 11 shows a course-level classification of a 24-second basketball video. The first step thus segments the video sequences into three different groups of: Low, Medium and High based on the “edgeness” criteria. We then define the following rules to provide meaning to different types of “edgeness” based on empirical investigations on sports videos:

- If the “edgeness” is high, then the frame represents a “crowded scene”
- If the “edgeness” is medium, then the frame represents a “normal play scene”
- If the “edgeness” is low, then the frame represents a “close-up scene”.

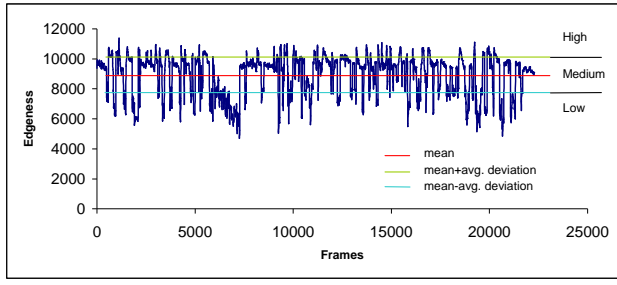


Fig. 11. The result of coarse-level classification in a basketball video.

Our empirical investigations consisted of observing basketball video clips, automatically categorizing the “edginess” into three levels, “Low”, “Medium”, and “High”. Video sequences of each edginess level were then reviewed and annotated with appropriate textual descriptions to describe events in the video clip. The terms used are subjective words to describe visual events. The observation results are shown in Table 1.

Table 1. Results of applying coarse-level classifications on sports videos.

| Sports | Coarse Level Semantic | Events |
|------------|-----------------------|--|
| Tennis | Low | Close-up views of players or umpires or a single spectator |
| | Medium | Normal play – focusing on players and grounds |
| | High | Crowds, Players with crowd on background |
| Basketball | Low | Close-up views of players or coaches or umpires |
| | Medium | Normal play |
| | High | Crowd, players with crowd on background |

In tennis videos, the low edginess based classification does not provide any distinct semantics. When we observed the video, we found that the “close-up views” contain some high edginess advertisement signs and texts, racket and nets. Similarly, normal play in basketball videos contains high edginess due to advertisement signs and texts. However, we think that it is within an acceptable range for coarse level classification.

In the second step, we incorporate temporal domain knowledge to further classify the frames and extract meaningful video sequences. To illustrate an example, we focus on only one edginess level, “Low”. In a basketball video, the low edginess indicates the “close-up views”. Our observation shows that there are different types of

“close-up views” depending on the events that have happened. These close-up views can be categorized as follows based on the temporal duration of the event:

- *Goal Close-up Views*: In a basketball game, the game becomes particularly interesting and gains momentum when there is a successful field goal. Such successful goals are marked by “close-up view” of the scoring player. It is also noticeable that such close-up views last for a short duration as the other team player immediately takes control of the ball and resumes normal play. It implies that a close-up view with short duration indicates “goal close-up view”.
- *Foul Close-up Views*: In basketball game, the play halts for sometime when a player commits a foul. Such fouls are marked by “close-up views” of the player who commits the foul. It is also noticeable that such close-up views last for a longer duration. It implies that a close-up view with long duration indicates a “foul close-up view”.
- *Other Close-up Views*: Apart from above two types of close-up views, there are other close-up views that occur during the game. Some of the other common close-up views are close-up views of coaches, close-up views of spectators (known public figures such as Bill Clinton), etc. Duration of such close-up views varies under different circumstances. However, we notice that due to the fast nature of the game such close-up views occur during the breaks.

Therefore, we use duration to classify close-up views into “Goal close-up view” and “Foul close-up view” in a basketball game as follows:

1. If the duration of the edginess remains “low” for less than 2 seconds, then the video sequence is classified as a “goal close-up view”.
2. If the duration of the edginess is greater than 2 seconds, then the video sequence is classified as “foul close-up view”.

Fig. 12 shows the low edginess sequences of the same basketball video shown in Fig. 11 with additional information.

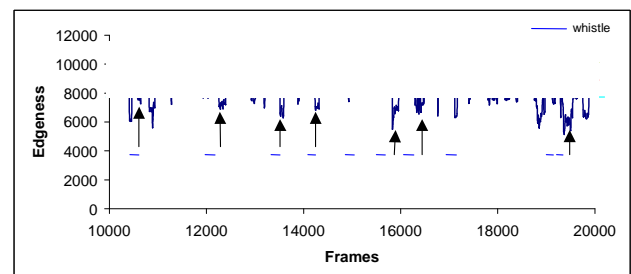


Fig. 12. A video segment of the basketball video (Fig. 11) from frame 10000 to 20000 with only low edginess sequences.

It is clear that most of the low edginess segments are of longer duration, which indicates “foul close-ups”. Our

automatic analysis shows that there are a total of 19 low edgeness sequences: 10 fouls, 1 goal and 8 others. Out of 19 low edgeness sequences 11 have duration greater than 2 seconds, of which 7 are foul close-up views and 4 are others. The correctness of the edgeness and its duration-based classification is about 65%.

In order to further improve the classification of the longer duration close-up views we use an audio feature, “whistle”.

In a basketball game, the umpire blows a whistle when a player commits a foul. We use an audio analysis tool, called MPEG Maaate [15], to determine whistles from an audio segment of the MPEG videos. Our whistle detection algorithm uses subband energy. The accuracy of the whistle detection algorithm is about 95%. Identifying “whistles” is outside the scope of this paper and is described in [15]. Our observation confirmed that the algorithm detects some non-whistle segments as whistle segments due to the presence of whistle when crowd was cheering for some interesting events. Such video sequences are marked by “crowded scene” based on edgeness. Therefore, whistle can be used to filter out longer duration “other close-up views” from “foul close-up views”.

We applied the whistle filter into the basketball game as shown in Fig. 12. The arrows indicate the “close-up views” that have duration greater than 2 seconds and follows “whistle” within 2 seconds. This filters out all 7 foul close-up views out of 11 close-up views with duration greater than 2 seconds. Thus, introduction of an additional feature, such as a whistle, as a filter increases the foul close-up classification rate from 65% to 100%. Thus, we conclude that edgeness feature, when used in conjunction with other audio-visual features, is useful to classify video sequences into meaningful semantic segments.

4 Conclusions

This paper addresses an interesting problem of identifying video sequences and classifying them into meaningful segments using edge features. The aim of the paper is in two fold: efficient detection of edges from MPEG videos and usage of edges for semantic classification.

We have defined two feature sets of DCT coefficients and used them to detect edges in MPEG video. Our algorithm relies on simple tests using DCT edge features, and is consequently very fast while it offers some visual accuracy. However, a major drawback is that the detected edges have poor connectivity with each other because each of them is individually processed within a block boundary. A post-processing technique could be further used to clean the edges of an object, that is, to link broken edge lines and to remove noise.

We have also shown a two-level classification and filtering approach to classify video sequences into high-level concepts using presence of edge information for certain temporal interval. In the first step, we have used the edgeness criteria and broadly classified video

sequences into three different semantic segments: low, medium and high edgeness, and corresponding semantics in sports videos. We have then used the duration of edgeness to further classify each segment into more concrete semantic segments such as “foul” and “goal”. We also show how we can improve the classification rate by using other features in conjunction with edge information.

5 References

- [1] H. Wang and S.F. Chang, “A high efficient system for automatic face region detection in MPEG video,” *IEEE Trans. on Circuits and System for Video Technology*, Vol.7, No.4, pp.615-628, 1997.
- [2] A. Yoshitaka, Y. Hosoda, M. Hirakawa, and T. Ichikawa, “Content-based retrieval of video data based on spatiotemporal correlation of objects,” *Proc. IEEE Multimedia Computing and Systems*, pp.208-213, 1998.
- [3] D.D. Saur, Y.-P. Tan, S.R. Kulkarni, and P.J. Ramadge, “Automatic analysis and annotation of basketball video,” *Proc. SPIE Storage and Retrieval for Image and Video Databases V*, Vol.3022, pp.176-187, Feb. 1997.
- [4] S. Nepal, U. Srinivasan and G. Reynolds, “Automatic detection of “Goal” segments in basketball videos,” *ACM Multimedia 2001*, pp.261-269, Sep.-Oct. 2001.
- [5] J.F. Canny, “Finding edges and lines in images,” MIT, Cambridge, Technical Report 720, June 1983.
- [6] D.C. Marr and E. Hildreth, “Theory of edge detection,” *Proc. of the Royal Society of London, Series B*, Vol. 207, 1980.
- [7] L.S. Davis, “A survey of edge detection techniques,” *Computer Graphics and Image Processing*, Vol. 4, pp. 248-270, 1975.
- [8] R.M. Haralick, “Digital step edges from zero crossing of second directional derivatives,” *IEEE Transactions on PAMI*, Vol. 11, No. 1, pp. 58-68, Jan. 1984.
- [9] B. Shen and I.K. Sethi, “Direct feature extraction from compressed images,” *Proc. SPIE Storage & Retrieval for Image and Video Databases IV*, Vol.2670, 1996.
- [10] M. Lee and G. Crebbin, “Classified vector quantisation with variable block-size DCT models,” *IEE Proc.-Vis. Image Signal Process.*, Vol.141, No.1, pp.39-48, Feb. 1994.
- [11] ISO/IEC 11172-2, “Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 2: Video,” 1993.
- [12] M. Lee and G. Reynolds, “Edge detection using DCT coefficients in MPEG video,” Technical Report 01/28, CSIRO Mathematical and Information Sciences, Feb. 2001.

- [13] L. Gu and D. Bone, "Skin colour region detection in MPEG video sequences," *International Conference on Image Analysis and Processing*, Venice, Italy, Sep. 1999.
- [14] L. Gu, "Text detection and extraction in MPEG video sequences," *Proc. Intl. Workshop on Content-Based Multimedia Indexing*, Italy, Sept. 2001.
- [15] MPEG Maaate:
<http://www.cmis.csiro.au/dmis/Maaate/>