The final published version is available at http://dx.doi.org/10.1109/ICME.2002.1035402.

EXTRACTING COARSE BOUNDARY FEATURES FOR VIDEO PROCESSING

Ramazan Savaş Aygün and Aidong Zhang

Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260-2000
{aygun,azhang}@cse.buffalo.edu

ABSTRACT

In this paper, we present a method to extract coarse boundary features from Discrete Cosine Transform (DCT) compressed blocks. Video sequences usually contain enormous data for video processing. Decompressing data and then processing uncompressed data is computational intensive. Moreover, this may yield erroneous results due to processing unnecessary data. Our goal is to extract coarse boundary features without decompressing original data, and then to eliminate insignificant blocks for decompressing. The features are extracted about the smoothness, the boundary visibility, and the boundary structure of a block. In the paper, we give an application of using coarse features for video object segmentation.

1. INTRODUCTION

The new video coding standard MPEG-4 [1] enables the content-based functionalities by introducing the concept of video object planes (VOP). The coding of video sequences that are segmented based on video contents, flexible reconstruction, and manipulation of video contents at the decoder have been the primary objective. Thus, video object segmentation, which emphasizes on partitioning the video frames into semantically video objects and the background, becomes a significant issue for the effective manipulation of MPEG-4 and MPEG-7. However, there has been significant amount of video that has been compressed using MPEG-1 and MPEG-2, which use Discrete Cosine Transform (DCT) to compress data. Decompressing data and then processing decompressed data is computational intensive. It is important to extract some coarse features to reduce decompression and to use these features in processing.

Feature extraction and edge detection in compressed DCT domain have been studied in [2, 3, 4]. Patterns of the first AC coefficients by zigzag ordering have been used for coarse edge detection [4]. The number of non-zero coefficients has been studied in [3]. First two AC coefficients are used for edge and shape detection in [2]. The general problem with the coarse edge detection problems is that most of the weight is given to coefficients that appear first according to the zigzag ordering. In this paper, AC coefficients are ordered according to their absolute magnitude. The features are obtained from the sorted AC coefficients.

A segmentation algorithm based on Hausdorff distance by extracting edges is proposed in [5]. A model based on detection and tracking of edges [6] discovers edges by using the edges in consecutive frames and in the background. Video object segmentation methods [5, 6] can detect the edges relying on the moving edges of the objects. These methods either require the raw data or need to decompress the original data. The video objects can be segmented using the background model. The blocks from the frames and the background model are compared to extract objects. The features extracted from the compressed data can be used to identify significant regions before the segmentation. Since the compression technique in MPEG-1, MPEG-2, and MPEG-4 is DCT, we propose a reliable method to extract significant blocks by extracting features about the smoothness and boundaries from DCT compressed blocks. Features about smoothness and structure of boundaries are evaluated to determine the significant blocks from compressed video for object segmentation.

This paper is organized as follows. The extraction of features from DCT compressed blocks is described in Section 2. The extraction of object boundaries and object segmentation are explained in Section 3. The last section concludes our paper.

2. EXTRACTION OF FEATURES FROM A DCT BLOCK

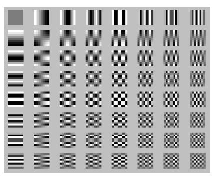
There has been significant video data that are compressed using DCT. The traditional methods decompress all the blocks and then apply video processing techniques. This increases computational complexity due to decompressing unnecessary blocks and processing these insignificant blocks. The

This research is supported by NSF grant IIS-9733730 and NSF grant IIS-9905603.

significant blocks can be determined by analyzing DCT blocks. In this section, we propose several features that can be extracted to compare two blocks. A DCT compressed block is obtained from 8x8 rectangular region of pixels. The major features that will be extracted are smoothness, the complexity of patterns, and the boundaries (or edges) inside a block.

2.1. Smoothness and Patterns

Each block is composed of DC and AC coefficients. DC coefficient of a block carries the most information about the block which is the 8 times of the average of the pixel values inside the block. However, DC coefficient provides no information on the structure of a block or how pixels are spread. If only DC coefficients are used for comparing blocks, different blocks may be assumed to be similar because of the equality in the average of the values in a block.



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure 1: 2D DCT basis images and their zigzag numbering.

AC coefficients must be used effectively to determine the contents of a block. Each AC coefficient is the coefficient of a basis image shown in Figure 1. The number of non-zero AC coefficients (NZ_{AC}) shows how complicated the block pattern is [3]. If NZ_{AC} is 0, the block pattern is smooth. This also indicates the absence of an edge in the block. As NZ_{AC} increases, smoothness in the block decreases and some patterns become visible.

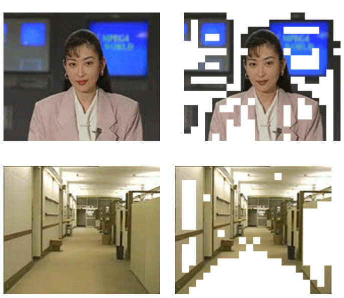


Figure 2: Blocks that contain edges.

The magnitude of AC coefficients is an indicator of edges or boundaries that may exist in the block. AC coefficients are sorted in descending order according to the absolute value of the AC coefficients. We sum up the absolute value of M highest AC coefficients, and the sum yields the block boundary visibility (BV). High values of BV indicate clear boundaries whereas low values indicate weak boundaries. We do not use all the coefficients since low coefficients determine the shape of the boundary rather than the visibility. This method does not rely on first coefficients with respect to zigzag ordering as in [4, 2]. Figure 2 shows the blocks that contain edges from the first frames of 'Akiyo' and 'Hall Monitor' test sequences.

2.2. Boundaries

The zigzag index (Figure 1) of the highest absolute magnitude of AC coefficient indicates the boundary type in the block. $Ordered_{AC}$ maintains the indices of AC coefficients in descending order according to their absolute values and $Ordered_{AC}(i)$ points the i^{th} $(0 \leq i < NZ_{AC})$ highest index. There is a *vertical* boundary if $Ordered_{AC}(0) \in \{1, 5, 6, 14, 15, 17, 28\}$ where $Ordered_{AC}(0)$ is the index of the highest AC coefficient. Basis images of these AC coefficients only have vertical borders. There is a *horizontal* boundary if $Ordered_{AC}(0) \in \{2, 3, 9, 10, 20, 21, 35\}$. If $Ordered_{AC}(0) \in \{4\}$, there is a *diagonal* boundary in the block.

The order of indices with respect to the absolute magnitude of AC coefficients indicates where the boundary is in the block. The sign of the AC coefficients indicates the dark and light regions in the block. Thus, we can extract coarse information by ordering of indices and using the signs of AC coefficients.

We extracted blocks that contain boundaries from the first image of 'Akiyo' test sequence. For example, $Ordered_{AC} =< -1, +6 >$ denotes that the index of the highest AC coefficient is 1 and it is negative, and the second highest index is 6 and it is positive. We deduce that the left part of the block is darker than its right side because the highest index 1 has a negative coefficient. The second index helps us locate the boundary. The second AC index is still a vertical AC index, therefore the block has a vertical boundary. Since the second AC index is $+6$, if basis images of -1 and $+6$ are overlapped, the boundary appears to be $2^{nd} - 3^{rd}$ column from the left. Some examples from the blocks of 'Akiyo' test sequence are shown in Figure 3. First two coefficients usually determine the position of the boundary and the third one indicates the slope in the boundary. Since AC coefficients having larger magnitudes are effective in determining the boundaries, this method is more reliable than the other methods that rely on the zigzag ordering of coefficients.

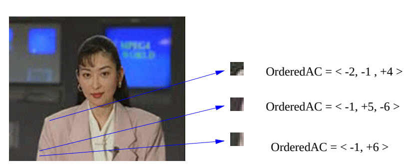


Figure 3: Boundaries in a block.

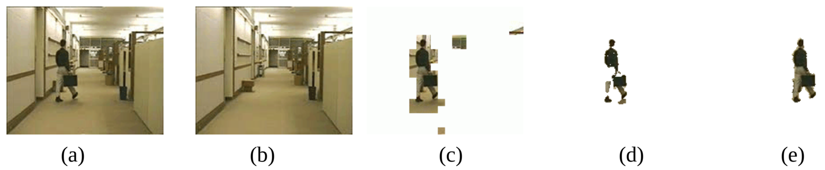


Figure 4: Experiments. (a) frame 43 (b) background image (c) decompressed (significant) blocks for edge detection (d) thresholded frame (e) video object.

The feature vector of a block is 5-tuple $B = \{DC, NZ, BV, BoundaryType, Darkness\}$. The comparison of two blocks is performed by comparing their feature vectors. The type of the boundary is important to compare the similarity of blocks. Sometimes, a block is compared with neighboring blocks to check the continuity of the boundary.

3. VIDEO OBJECT SEGMENTATION

The general method for segmentation of video objects is to detect the boundaries of the object. Therefore, if a region does not have a visible boundary, it is very unlikely that the region will contain the boundary for the object. The background model can be generated automatically or presented to the system. The feature vectors of blocks of the frames and the background model are compared to eliminate insignificant blocks. If the feature vectors are different, the frame block is likely to contain data about an object and considered as a significant block. The significant blocks are decompressed and used for further processing.

Firstly, the blocks in the background model and the frame are compared. Three thresholds, τ_{DC}, τ_{NZ}, and τ_{BV}, are used to compare DC coefficients, the number of non-zero AC coefficients, and boundary visibility of feature vectors. Let f_1 and f_2 be feature vectors of two blocks to be compared. If $|f_1(DC) - f_2(DC)| > \tau_{DC}$ or $|f_1(NZ) - f_2(NZ)| > \tau_{NZ}$, the blocks are considered as different blocks. If $f_1(BV) < \tau_{BV}$ and $f_2(BV) < \tau_{BV}$, blocks are assumed to have no visible boundaries. If only one of the blocks has a visible boundary (i.e., $\geq \tau_{BV}$), the blocks are considered different. If two blocks have visible boundaries, the boundary type and the darkness of boundaries are compared. If both are the same, they are treated as similar. The different blocks are considered as significant blocks. These blocks are decompressed and used in further processing. The frame 43 and background of 'Hall Monitor' test sequence are shown in Figure 4 (a) and (b), respectively. The significant blocks that are selected by comparing coarse boundary features are displayed in Figure 4 (c). Two macroblocks are misdetected as significant blocks due to their complex patterns.

Secondly, edges in the significant blocks are extracted to detect the boundaries of a video object. The Canny [7] edge detector is used to extract the edges. The most distinguished feature of an edge is its gradient, ∇g. Edge matrix E maintains the edges in an image, which is denoted by

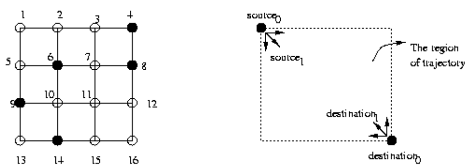
$$E = \begin{cases} E_{xy} = 1 & \text{if there is an edge at (x,y)} (\nabla g > \tau_{Edge}) \\ E_{xy} = 0 & \text{otherwise} \end{cases}$$

Let E^{β} and $E^{\mathcal{F}_i}$ represent the edge matrix for the background and frame i. The difference edge matrix (DE^i) holds the different edges that exist in the frame but not in the background where $DE^i = ((E^{\mathcal{F}_i} \text{ XOR } E^{\beta}) - E^{\beta})$. DE^i may also contain noisy edges due to the illumination change in the environment. $\Phi(DE^i)$ denotes the edges after noise removed. The removal of the noise is performed based on the image produced with thresholding (Figure 4 (d)). It is possible that significant blocks may be missed in the initial comparison phase. A significant block may be missed because of a weak boundary close to the borders of a block. This kind of a block will have similar DC, NZ_{AC}, and BV values with the background model. If there is an edge on the border of an insignificant block, the insignificant block is also decompressed before further processing.

Thirdly, as a result of the previous steps the edges may be disconnected. The edges of the video object may be removed since they overlap with the background edges or may be removed as noise mistakenly. If the gradients of nodes are below the threshold for edge, the edges may again be disconnected. Linking of edges will create new edges and is composed of two phases: linking of close edges and linking of distant edges. Let $\Phi(DE^i_{xy}) = 1$ and $\Phi(DE^i_{pq}) = 0$ where (x, y) and (p, q) are coordinates, $x - 1 \leq p \leq x + 1$, and $y - 1 \leq q \leq y + 1$. Let N_{pq} denote the set of 8-connected nodes of a node at (p, q). For simplicity, assume that $E = \Phi(DE^i)$. E_{pq} is converted to an edge if

$$E_{xy} = 1 \wedge \exists p \exists q (E_{pq} = 0 \wedge \exists s \exists t (E_{st} = 1 \wedge (E_{st} \in N_{pq}) \\ \wedge \neg \exists u \exists v (E_{uv} = 1 \wedge (E_{uv} \in N_{xy}) \wedge (E_{st} \in N_{uv}))$$

is true. Informally, a node is converted to an edge if the node connects edges that are not connected directly or through their neighbors. In this context, a node is an edge if its gradient is higher than the edge threshold. For example, in Figure 5 (a), nodes 4, 6, 8, 9 and 14 denote the edges. We want to link node 6 to another node. The 8-connected neighbors of node 6 are nodes 1, 2, 3, 5, 7, 9, 10 and 11. Node 9 is already an edge. Nodes 1 and 2 do not connect node 6 to any other edge, therefore these are ignored. If node 3, 7 or 11 becomes an edge, then nodes 4 and 8 are reachable from node 6. Nodes 10 and 11 connect node 6 to node 14 but node 14 is already reachable from node 6 through node 9. So, the only candidates are nodes 3, 7 and 11. Once one of nodes 3, 7 or 11 is chosen, all the edges in the figure are reachable from node 6.



(a) (b)

Figure 5: Edge Linking. (a) close edge linking, (b) distant edge linking.

Linking distant edges is different since none of the neighbors connect the edges to the distant edges. Assume that we want to link $source_{xy}$ to $destination_{pq}$. The main idea is to follow the nodes having high ∇g. There are 2 significant problems: 1) if the edge is also connected to a strong edge in the background, the trajectory from the source may stray, and 2) if we start from the source and try to reach the destination by following the gradients of nodes, the edges nearby the destination may be misdetected and the real edges may be eliminated. We use a heuristic to solve the first problem. If the source is at (x, y), destination is at (p, q), and (s, t) is an edge on the trajectory from the source to the destination, then $min(p, x) \leq s \leq max(p, x)$ and $min(q, y) \leq t \leq max(q, y)$. To solve the second problem, instead of starting from the source and reaching the destination, a single step from the source to destination and a single step from the destination to the source is taken at each iteration. A single step is to visit an neighbor of an edge (Figure 5 (b)). We use the Manhattan distance to measure the distance between edges. The condition $distance(source_{i+1}, destination_{i+1})$ $< distance(source_i, destination_i)$ must be satisfied after each iteration. The iteration continues until the source and the destination are 8-connected. The snake model [8] is proposed to extract open and closed contours of objects. It needs initial points for the contours and requires weights for adjusting the curves of the contours. The distant edge linking can also be performed using the snake model. Since the focus of the paper is the extraction of the coarse boundary features, experience with the snake model is not discussed in this paper. Finally, the regions within the boundary of the object are filled to obtain the video object (Figure 4 (e)).

In our experiments, we observed that sorted AC coefficients are significant in detection and extraction of coarse boundaries. Since first two AC coefficients are usually the highest AC coefficients, the methods that depend on these coefficients may also yield good results in some cases. However, during comparisons of video frames, there are also blocks that have other significant coefficients which cannot be distinguished using the first 2 AC coefficients. This increased the number of blocks that are assumed to be similar in the preprocessing. NZ_{AC} is also not enough to describe the coarse boundary features when NZ_{AC} is quite small (i.e. less than 6). If NZ_{AC} is high, it is likely that there are many AC coefficients having small magnitude. NZ_{AC} is useful in comparisons if the difference between the number of non-zero AC coefficients is significant. By using combination of the prestated features, we obtained satisfactory results.

4. CONCLUSION

In this paper, we presented a method to extract coarse boundary features from DCT compressed blocks. We showed that DC coefficient, smoothness, boundary visibility, boundary type, and darkness are good features to determine significant blocks. The boundary features are used to eliminate the insignificant blocks for video processing. We also gave an example on how these features can be used in video object segmentation. In this paper, the boundary types that are covered are vertical, horizontal, and diagonal. Other types of boundaries like curved boundaries can also be detected using other AC coefficients but this is left as a further research. In those cases, the boundary type is represented with the index of the highest AC coefficient.

5. REFERENCES

[1] T. Sikora, "The mpeg-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technology*, vol. 7, pp. 19–31, February 1997.

[2] C.W. Ngo, T. C. Pong, and R. T. Chin, "Exploiting image indexing techniques in dct domain," *Pattern Recognition*, 2000.

[3] F. Arman and *et al.*, "Image processing on compressed data for large video databases," in *Proc. ACM Int. Conf. Multimedia*, June 1993.

[4] Bo Shen and Ishwar K. Sethi, "Direct feature extraction from compressed images," in *Proc. SPIE Storage, Retrieval for Image and and Video Databases IV*, 1996, vol. 2670.

[5] T. Meier and K.N. Ngan, "Video object plane segmentation using a morphological filter and hausdorff object tracking," in *ICIP*, October 1998.

[6] C. Kim and J-N. Hwang, "An integrated scheme for object-based video abstraction," in *ACM*, October 2000, pp. 303–311.

[7] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 679–698, November 1986.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Proc. of First International Conf. on Computer Vision*, pp. 259–268, 1987.