An Integrated Approach to Video Retrieval

Liang-Hua Chen[1] **Kuo-Hao Chin**[1] **Hong-Yuan Liao**[2]

[1] Department of Computer Science and Information Engineering,
Fu Jen University,
Hsinchuang, Taipei, TAIWAN.
Email: lchen@csie.fju.edu.tw

[2] Institute of Information Science,
Academia Sinica,
Nankang, Taipei, TAIWAN.

Abstract

The usefulness of a video database depends on whether the video of interest can be easily located. In this paper, we propose a video retrieval algorithm based on the integration of several visual cues. In contrast to key-frame based representation of shot, our approach analyzes all frames within a shot to construct a compact representation of video shot. In the video matching step, by integrating the color and motion features, a similarity measure is defined to locate the occurrence of similar video clips in the database. Therefore, our approach is able to fully exploit the spatio-temporal information contained in video. Experimental results indicate that the proposed approach is effective and outperforms some existing technique.

Keywords: Video retrieval, video database, video matching, similarity measure.

1 Introduction

The advances in low cost mass storage devices, higher transmission rates and improved compression techniques, have led to the widespread use and availability of digital video. Video data offers users of multimedia systems a wealth of information and also serves as a data source in many applications including digital libraries, publishing, entertainment, broadcasting and education. The usefulness of these applications depends largely on whether the video of interest can be retrieved accurately within a reasonable amount of time. Video query by keywords is inefficient, because it is not easy to describe video content in words. Alternatively, query-by-example is a more feasible approach that searches video database according to the visual content of query example.

The query example may be an image, a shot or a clip. A shot is a sequence of frames that was continuously captured by the same camera, while a clip is a series of shots describing a particular event. For example, a dialogue clip between two people may have a shot of speaker A, followed by a shot of the other speaker B, followed by a wide-angle shot of two parties involved. Video retrieval based on a single shot may not be practical since a shot itself is only a part of an event and does not convey full story. On the other hand, clip-based retrieval is more concise and convenient for most casual users. Thus, our problem can be formulated as: given a sample clip, find all occurrences of similar (or relevant) video clips in the database.

Current techniques for content-based video retrieval can be broadly classified into two categories: frame sequence matching(Mohan 1998, Tan et al. 1999, Naphade et al. 2000, Hoad & Zobel 2003, Ren & Singh 2004, Kim & Vasudev 2005, Toguro et al. 2005) and key-frame based shot matching(Liu et al. 1999, Jain et al. 1999, Lienhart et al. 2000, Kim & Park 2002, Diakopouos & Volmer 2003, Peng et al. 2003, Peng & Ngo 2004, Sze et al. 2005, Ho et al. 2006, Luo et al. 2007). The first one is derived from the sequential correlation matching widely used in the signal processing domain. These methods usually focus on frame-by-frame comparison between two clips in order to find sequences of frames that are consistently similar. The common drawback of these techniques is the heavy computational cost of the exhaustive search. Although there exist some techniques(Kashino et al. 2003, Yuan et al. 2004) to improve the linear scanning speed, their time complexity still remains at least linear to the size of database. Additionally, these approaches are susceptible to alignment problem when comparing clips of different encoding rates. In the second category, each video shot is represented by a key-frame compactly. To reduce computational cost, video sequence matching is achieved by comparing the visual features of key-frames. The problem with these approaches lies in that they all leave out the temporal variations and correlation between key-frames within an individual shot. Also, it is not clear as to which image should be used as the key-frame for a shot. To strike a good balance between searching accuracy and computational cost, in this paper, we propose an integrated approach for shot matching. In contrast to previous approaches, our approach analyzes all frames within a shot to extract more visual features for shot representation. Because there does not exist a single visual feature for the best representation of video content, we integrate several visual features to capture the spatio-temporal information more accurately.

The main issues regarding content-based video retrieval are: (1) how to select visual features to represent the content of a video clip and (2) how to define a distance metric to measure the visual similarity between two video clips. The next section of this paper describes the visual features used in our work. Then, the proposed shot similarity measure and video matching algorithm are described in Section 3. In Section 4, relevance feedback technique is introduced to improve the video retrieval result. The performance evaluation of our approach is reported in Section 5. Finally, some concluding remark is given in Section 6.

2 Visual Feature

Shot is the fundamental unit of a video. To facilitate subsequent video analysis, in our system, the query video clip and database video are segmented into shots. This task is achieved by applying shot boundary detection algorithm (Chen et al. 2003) to the original video sequence. A major requirement for shot matching is to define a content representation that captures the common aspects or characteristics of the shot. One common method is to select one key-frame from the shot and use the image features of the key-frame as an abstract representation of the shot. For shot with fast changing content, one key-frame per shot is not adequate. Besides, the content description it provides varies significantly with the key-frame selection criterion. To avoid these problems, a more feasible approach is to consider the visual content of all the frames within a shot for shot representation.

Color is one of the most widely used visual features in video content analysis, because it is an important source of information in visual content for discrimination. However, the amount of color information in video is vast. The raw data of video has to be transformed into compact feature representation that conveys only the most salient color aspects of the visual content. Color histogram is the most commonly used color feature representation. The histogram-based approach is relatively simple to calculate and can provide reasonable results. However, due to the statistical nature, color histogram does not capture spatial layout information of each color. When the image collection is large, two different content images are likely to have quite similar histograms. To remedy this deficiency, the distribution state of each single color in the spatial (image) domain needs to be taken into account.

The color histogram for an image is constructed by counting the number of pixels of each color. The main issues regarding the construction of color histogram involve the choice of color space and quantization of color space. The RGB color space is the most common color format for digital images, but it is not perceptually uniform. Uniform quantization of RGB space gives perceptually redundant bins and perceptual holes in color space. Therefore, the non-uniform quantization may be needed. Alternatively, HSV (hue, saturation,intensity) color space is chosen since it is nearly perceptually uniform. Thus, the similarity between two colors is determined by their proximity in the HSV color space. When a perceptually uniform color space is chosen, uniform quantization may be appropriate. Since the human visual system is more sensitive to hue than to saturation and intensity(Wan & Kuo 1998), H should be quantized finer than S and V. In our implementation, hue is quantized into 20 bins. Saturation and intensity are each quantized into 10 bins. This quantization provides $20 \times 10 \times 10$ distinct colors (bins), and each bin with non-zero count corresponds to a *color object*.

Since we are interested in the whole shot rather than single image frame, only one histogram is used to count the color distribution of all image frames within a shot. The use of one histogram as color descriptor for a group of frames has been accepted as the MPEG-7 standard(Sikora 2001). Then, each bin of the resulting histogram is divided by the number of frames in a shot to obtain the average color histogram. Next, several spatial features are calculated to characterize the distribution state of each color object in each image frame. Assuming a set of pixels $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ belong to color object c_i, k is the image size and m is the total number of 4-connected pixels in S. Then, we define

(i) density of distribution

$$f_{i1} = \frac{n}{k}$$

(ii) compactness of distribution

$$f_{i2} = \frac{m}{n}$$

(iii) scatter

$$f_{i3} = \frac{1}{n\sqrt{k}} \sum_{j=1}^{n} \sqrt{(x_j - x_\mu)^2 + (y_j - y_\mu)^2}$$

where $x_\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $y_\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$

To define the fourth feature, the image is equally partitioned into p blocks of size 16×16. A block is *active*, if it contains some subset of S. Let the number of active blocks in the image frame be q, we define
(iv) ratio of active block

$$f_{i4} = \frac{q}{p}$$

After the spatial features of all image frames within a shot are computed, we take average of these values respectively. Let $\overline{f_{i1}}, \overline{f_{i2}}, \overline{f_{i3}}$ and $\overline{f_{i4}}$ be the average feature values of color object c_i within a shot, for two color objects c_i and c_j, the difference of spatial distribution within a shot is defined as

$$\begin{aligned} D_s(c_i, c_j) = \; & \tfrac{1}{4}(|\overline{f_{i1}} - \overline{f_{j1}}| + \quad |\overline{f_{i2}} - \overline{f_{j2}}| + \\ & |\overline{f_{i3}} - \overline{f_{j3}}| + \quad |\overline{f_{i4}} - \overline{f_{j4}}|) \end{aligned} \tag{1}$$

Motion is another visual feature which is essential to capture temporal variation of video. It also reveals the correlations between frame sequences within a video shot. To complement the color histogram and convey the motion information contained in video, a 2-D motion histogram of motion vectors is constructed. In our implementation, the displacement in vertical and horizontal directions are all quantized into 121 bins (60 bins for positive, 60 bins for negative, and 1 bin for zero), and there are a total of 121×121 distinct bins for the 2-D motion histogram.

To obtain the motion vectors between successive frames, previous approaches are based on the feature matching(Wang & Mersereau 1999) or optical flow computation(Tzovaras & Strintzis 1998). These techniques are computationally intensive. Alternatively, we directly use the motion vectors encoded in the MPEG-1 video stream(Gall 1991). In MPEG video, each frame is partitioned into blocks of size 16×16 pixels called macro blocks (MBs). MPEG defines motion vector as the displacement from the Target (current frame) MB to the Prediction (reference frame) MB. In MPEG format, there are three types of frames: I, P and B frames. I frames are skipped because they are intra-coded and no motion information is available. P frames have forward motion prediction and B frames have both forward and backward motion prediction. In our system, only the forward motion vectors encoded in P frames are extracted and accumulated into the motion histogram. The final histogram is then normalized (i.e. divided by the number of frames in a shot) to obtain the average motion histogram to represent the motion feature of a shot. Since the resulting motion histogram is constructed from the average statistics of the entire shot, it is not sensitive to the error of motion estimation due to noise for a particular frame, and the separation of camera motion from object motion is avoided. For this reason, the application of this method is not limited to any specific type of scenes.

3 Video Matching

Our approach performs video matching at two levels. At the shot level, the objective is to evaluate the visual similarity between two shots with different durations (lengths). At the sequence level, the video matching is achieved by sliding the query video clip (a matching window) along the database video at one shot increment and computing the similarity metric for every window position.

Histogram intersection is a popular similarity measure for color-based image matching(Swain & Ballard 1991). It yields the number of pixels that have same color in two images. In our work, we extend this idea to shot matching. Let A,B be the set of all color objects in shot S_1 and S_2 respectively, for a given $u \in A$, its similar color object in B is some $v \in B$ such that $||u - v|| < \epsilon$, where $||u - v||$ denotes the Euclidean distance between u and v in the HSV color space and ϵ is a threshold (ϵ is set to be 3.). Then, (u, v) is called a *similar color pair*. Let $\Omega = \{(u, v)|(u, v) \in A \times B, (u, v)$ is a similar color pair$\}$, the shot similarity measure for color between S_1 (with the average color histogram $\overline{H_{C1}}$) and S_2 (with the average color histogram $\overline{H_{C2}}$) is defined as

$$\begin{aligned}\text{ShotSim\_C}(S_1, S_2) &= \frac{1}{k} \sum_{(u,v)\in\Omega} \{W(D_s(u, v)) \times \\ &\quad \min(\overline{H_{C1}}(u), \overline{H_{C2}}(v))\} \quad (2)\end{aligned}$$

where k is the image size, D_s is the difference of spatial features as defined in equation (1) and W is a weight function defined as

$$W(x) = \frac{1}{1 + e^{(ax+b)}}$$

The weight function W is the general form of sigmoid function which is frequently used in neural networks computation(Haykin 1999), where a and b are parameters. In our work, it is used to fuse spatial distribution information with histogram. The construction of this weight function is motivated by the psychophysical observation: the effect of spatial distribution on human perception is progressive(Boycott 2001). Only when the difference in spatial features is greater than a threshold, human perceive significant visual variation. The property of sigmoid function fulfills this requirement. As shown is Figure 1, the function value becomes small significantly for $x > 0.75$.

It is noted that a given color object in shot S_1 may have more than one similar color objects in shot S_2 as illustrated in Figure 2. To avoid the overlapping contribution in calculating shot similarity, after each step of $\min(\overline{H_{C1}}(u), \overline{H_{C2}}(v))$, $\overline{H_{C1}}(u)$ and $\overline{H_{C2}}(v)$ are all subtracted by $\min(\overline{H_{C1}}(u), \overline{H_{C2}}(v))$.

Assuming the vertical and horizontal displacements of motion vectors have values range from $-M$ to M, let $R = \{(x, y)| - M \leq x \leq M, -M \leq y \leq M, (x, y)$is a pair of motion vector$\}$. The shot similarity measure for motion between shot S_1 (with the average motion histogram $\overline{H_{M1}}$) and shot S_2 (with the average motion histogram $\overline{H_{M2}}$) is defined as

$$\begin{aligned}&\text{ShotSim\_M}(S_1, S_2) = \\ &\frac{1}{k} \sum_{(x,y)\in R} \min(\overline{H_{M1}}(x, y), \overline{H_{M2}}(x, y)) \quad (3)\end{aligned}$$

The overall shot similarity measure between shot S_1 and S_2 is then defined as the weighted sum of equation (2) and (3), i.e.,

$$\begin{aligned}\text{ShotSim}(S_1, S_2) = \; & W_c \times \text{ShotSim\_C}(S_1, S_2) + \\ & W_m \times \text{ShotSim\_M}(S_1, S_2) \quad (4)\end{aligned}$$

where W_c and W_m are the weights for the color and motion features, respectively. The setting of W_c and W_m is discussed in the next section.

Given the query video clip $Q = \{q_1, \cdots, q_m\}$ and the database video $V = \{v_1, \cdots, v_n\}$, where q_i and v_j denote the segmented shots, the similarity measure between the query clip and the database video segment starting at the i-th shot is defined as

$$D_i = \sum_{j=1}^{m} \text{ShotSim}(q_j, v_{i+j-1}) \quad (5)$$

If D_i is a local maxima and is also greater than a threshold T then a similar clip is detected at the i-th shot of database video. In our system, the threshold T is set to be 0.5. The smaller T value the more similar video clips are detected. Since our system is able to rank the similar video clips, the choice of threshold is irrelevant to the determination of the most similar video clip.

4 Relevance Feedback

The various techniques developed for content-based video retrieval are all efforts to try to map low level features to high level concepts. However, it is not easy to fill in the gap between these two levels in every case. In addition, different persons, or even the same person under different circumstances, may perceive the same visual content differently. Therefore, any method with a fixed set of visual feature representations and their corresponding weights cannot always effectively model high level concepts and user's subjective perceptions. To address this limitation, one possible solution is the *relevance feedback* (Rui et al. 1998). It is an interactive mechanism that involves a human as part of the retrieval process.

In the relevance feedback approach, under the assumption that high level concepts can be represented by low level features, the technique tries to establish the link between the two levels from a user's feedback. The user only needs to specify which video clips he or she thinks are relevant to the query. The weights embedded in the similarity measure are then dynamically updated to adjust the importance of the visual features used according to the user's subjective perceptions during each round of the retrieval process. As relevance feedback was applied to content-based image retrieval(Rui et al. 1998), we extend this technique to video retrieval. Our objective is to update the weights W_c and W_m in equation (4) to reflect the user's different emphasis on the feature representation in the overall similarity metric according to his or her feedback. This is done with the following algorithm.

Let R be the set containing the most similar N retrieved video clips according to the overall similarity value D_i, with W_c and W_m initially set to 0.5:

$$R = [R_1, \cdots, R_N]$$

Let $Score = [Score_1, \cdots, Score_N]$ be the set containing the relevance scores feedback by the user for each of the retrieved clips in R:

$$Score_i = \begin{cases} 3, & \text{highly relevant} \\ 1, & \text{relevant} \\ 0, & \text{no-opinion} \\ -1, & \text{non-relevant} \\ -3, & \text{highly non-relevant} \end{cases}$$

The choice of these numbers as the scores are arbitrary. The user may choose other numbers for their convenience.

Then, let R^c and R^m be the sets containing the most similar N clips to the query, according to only the color similarity measure and only the motion similarity measure, respectively.

$$R^c = [R_1^c, \cdots, R_N^c]$$

$$R^m = [R_1^m, \cdots, R_N^m]$$

Now, to calculate the new values for W_c and W_m, first set $W_c = 0$ and $W_m = 0$, then update these two weights using the following procedure:

$$W_k = \begin{cases} W_k + Score_i & \text{if} \quad R_i^k \in R \\ W_k + 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} i &= 1, \cdots, N \\ k &= c, m \end{aligned}$$

After this procedure, if $W_k < 0$, set it to be 0. The raw weights obtained by the above procedure are then normalized by the total weights to make the sum of the normalized weights equal to 1. It is observed that the more the overlap of relevant clips between R and R^k, the larger the weights of W_k. In other words, if a particular feature representation reflects a user's need, it receives more emphasis. Moreover, this algorithm can be repeated to iteratively fine-tune the retrieval results until the user is satisfied.

5 Experimental Results

To evaluate the performance of the proposed approach, we set up a database that consists of 3 hours of videos approximately. The genres of videos include home video, news, sports, movies and documentaries. The testing with different genres of videos would ensure that the overall performance of the algorithm is not biased toward a specific video category. Figure 3 shows an example of retrieving and ranking similar video clips with query clip (shown in the first row). In each row, sampled frames (one for each shot) are used to represent the content of video clip. As shown in Figure 3, the retrieved results are similar to the query clip, and they are ranked in descending order of similarity. Since the relevance feedback technique involves interactions with the user, it should be noted that the displayed clips are either results of a convergence after several iterations, or deemed "optimal" from a particular user's perspective. To demonstrate the effectiveness of the relevance feedback technique, retrieval results before and after the relevance feedback process are shown in Figure 4 and Figure 5, respectively. It is observed that improvement is made in ranking similar video clips.

The performance of video retrieval is usually measured by the following two metrics:

$$\text{Recall} = \frac{DC}{DB} \qquad \text{Precision} = \frac{DC}{DT}$$

where DC is the number of similar clips which are detected correctly, DB is the number of similar clips in the database and DT is the total number of detected clips. The ground truth of database, i.e., the decision whether a video clip is similar or not, is determined by human subjects. For performance comparison, we also implement the well known video retrieval algorithm proposed by Jain et al.(Jain et al. 1999). Their algorithm follows the key-frame based approach of identifying shots, selecting key-frames from a video, and then extracting image features (color, texture and motion) around the key frames. For each key frame in the query, a similar value is obtained with respect to the key frames in the database video. Consecutive key frames in the database video that are highly similar to the query key frames are then used to generate the set of retrieved video clips. To compare both approaches fairly, our system does not apply the relevance feedback technique at this stage. Table 1 gives the experimental results using five different query topics:

Table 1: Performance comparison for different queries.

Query #	Our Approach		Jain's Approach	
	Recall	Precision	Recall	Precision
1	0.75	0.86	0.45	0.63
2	0.80	0.80	0.65	0.75
3	0.75	0.86	0.53	0.75
4	0.83	0.83	0.60	0.63
5	0.75	0.63	0.55	0.46

1. Close-up interview shots.
2. Hot-air balloons.
3. Scene of a male character.
4. Free throw shots.
5. Classroom scene.

The performance of Jain's algorithm may be limited by the following factors:

- Only the image features of key-frame are used to represent the whole shot content.
- The color description is based on traditional histogram which does not capture spatial layout information of each color.
- The video similarity is measured by the Euclidean distance between feature histograms. However, two different bins may represent perceptually similar features but are not compared in this measure. It has been shown that histogram intersection distance is more effective than histogram Euclidean distance for image retrieval(Smith & Chang 1996).

6 Conclusion

We have presented a new video shot representation and a video similarity measure to achieve video retrieval task. Unlike key-frame based representation of shot, the proposed approach analyzes all frames within a shot to extract more visual features for shot representation. Our approach integrates color and motion features to fully exploit the spatio-temporal information contained in video. To improve the retrieval performance according to user's visual judgment, a technique called relevance feedback is also incorporated. Thus, the proposed system is able to resemble human similarity perception to some extent. Experimental results indicate that the proposed approach is effective and feasible in retrieving and ranking similar video clips. Finally, our future work should incorporate other video features, such as audio and text, for assessing video similarity.

References

Boycott, B. (2001), *Color Vision*, Cambridge University Press, Cambridge, U.K.

Chen, L., Su, C., Liao, H. & Shih, C. (2003), 'On the preview of digital movies', *Journal of Visual Communication and Image Representation* **14**(3), 357–367.

Diakopouos, N. & Volmer, S. (2003), Temporally tolerant video matching, *in* 'ACM SIGIR Workshop on Multimedia Information Retrieval', Toronto, Canada.

Gall, D. (1991), 'MPEG: A video compression standard for multimedia applications', *Communication of ACM* **34**(4), 46–58.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Prentice Hall, New Jersey, U.S.A.

Ho, Y., Lin, C., Chen, J. & Liao, H. (2006), 'Fast coarse-to-fine video retrieval using shot-level spatial-temporal statistics', *IEEE Transactions on Circuits and Systems for Video Technology* **16**(5), 642–648.

Hoad, T. & Zobel, J. (2003), Fast video matching with signature alignment, *in* 'ACM SIGMM International Workshop on Multimedia Information Retrieval', Berkeley, CA, pp. 262–269.

Jain, A., Vailaya, A. & Wei, X. (1999), 'Query by video clip', *Multimedia Systems* **7**, 369–384.

Kashino, K., Kurozumi, T. & Murase, H. (2003), 'A quick search method for audio and video signals based on histigram pruning', *IEEE Transactions on Multimedia* **5**(3), 348–357.

Kim, C. & Vasudev, B. (2005), 'Spatiotemporal sequence matching for efficient video copy detection', *IEEE Transactions on Circuits and Systems for Video Technology* **15**(1), 127–132.

Kim, S. & Park, R. (2002), 'An efficient algorithm for video sequence matching using the modified Hausdorff distance and the directed divergence', *IEEE Transactions on Circuits and Systems for Video Technology* **12**(7), 592–596.

Lienhart, R., Effelsberg, W. & Jain, R. (2000), 'VisualGREP: A systematic method to compare and retrieve video sequences', *Multimedia Tools and Applications* **10**(1), 47–72.

Liu, X., Zhung, Y. & Pan, Y. (1999), A new approach to retrieve video by example video clip, *in* 'ACM International Conference on Multimedia', pp. 41–44.

Luo, H., Fan, J., Satoh, S. & Ribarsky, W. (2007), Large scale news video database browsing and retrieval via information visualization, *in* 'ACM symposium on applied computing', Seoul, Korea, pp. 1086–1087.

Mohan, R. (1998), Video sequence matching, *in* 'Proceedings of International Conference on Acoustic, Speech and Signal Processing', pp. 3697–3700.

Naphade, M., Yeung, M. & Yeo, B. (2000), A novel scheme for fast and efficient video sequence matching using compact signature, *in* 'SPIE Conference on Storage and Retrieval for Media Database', pp. 564–572.

Peng, Y. & Ngo, C. (2004), Clip-based similarity measure for hierarchical video retrieval, *in* 'ACM SIGMM International Workshop on Multimedia Information Retrieval', pp. 53–60.

Peng, Y., Ngo, C., Dong, Q., Guo, Z. & Xiao, J. (2003), Video clip retrieval by maximal matching and optimal matching in graph theory, *in* 'International Conference on Multimedia and Expo', pp. 317–320.

Ren, W. & Singh, S. (2004), Video sequence matching with spatio-temporal constraints, *in* 'International Conference on Pattern Recognition', pp. 834–837.

Rui, Y., Huang, T., Ortega, M. & Mehrotra, S. (1998), 'Relevance Feedback: A power tool for interactive content-based image retrieval', *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5), 644–655.

Sikora, T. (2001), 'The MPEG-7 visual standard for content description - An overview', *IEEE Transactions on Circuits and Systems for Video Technology* **11**(6), 696–702.

Smith, J. & Chang, S. (1996), Tools and techniques for color image retrieval, *in* 'SPIE Conference on Storage & Retrieval for Image and Video Databases', San Jose CA, pp. 426–437.

Swain, M. & Ballard, D. (1991), 'Color indexing', *International Journal of Computer Vision* **7**(11), 11–32.

Sze, K., Lam, K. & Qiu, G. (2005), 'A new key frame representation for video segment retrieval', *IEEE Transactions on Circuits and Systems for Video Technology* **15**(9), 1148–1155.

Tan, Y., Kulkarni, S. & Ramadge, P. (1999), A framework for measuring video similarity and its application to video query by example, *in* 'International Conference on Image Processing', pp. 106–110.

Toguro, M., Suzuki, K., Hartono, P. & Hashimoto, S. (2005), Video stream retrieval based on temporal feature of frame difference, *in* 'Proceedings of International Conference on Acoustic, Speech and Signal Processing', Volume 2, pp. 445–448.

Tzovaras, D. & Strintzis, M. (1998), 'Motion and disparity field estimation using rate-distortion optimization', *IEEE Transactions on Circuits and Systems for Video Technology* **8**(2), 171–180.

Wan, X. & Kuo, C. (1998), 'A new approach to image retrieval with hierarchical color clustering', *IEEE Transactions on Circuits and Systems for Video Technology* **8**(5), 628–643.

Wang, H. & Mersereau, R. (1999), 'Fast algorithms for the estimation of motion vectors', *IEEE Transactions on Image Processing* **8**(3), 435–438.

Yuan, J., Tian, Q. & Ranganath, S. (2004), Fast and robust search method for short video clips from large video collection, *in* 'International Conference on Pattern Recognition', pp. 866–869.

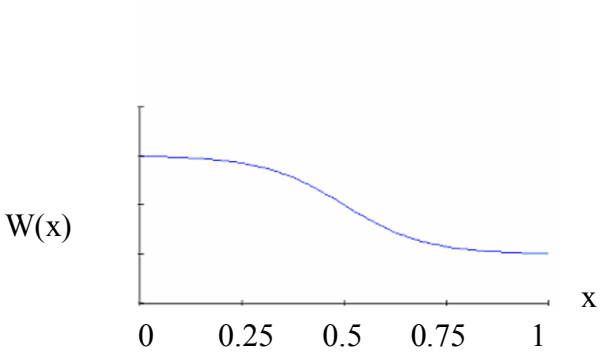

Figure 1: Sigmoid function with parameters a=10 and b=−5.

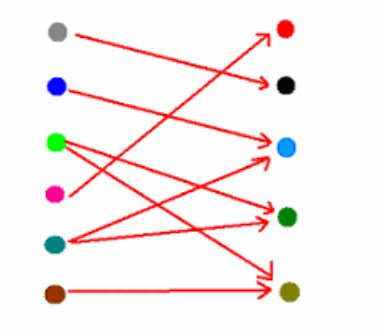

Figure 2: Finding similar color object pairs.

Figure 3: Retrieval result for a ``hot-air balloon'' query.

Figure 4: Retrieval result for a ``free throw'' query without relevance feedback.

Figure 5: Retrieval result for a ``free throw'' query with relevance feedback.