VisionGo: Towards video retrieval with joint exploration of human and computer

Huanbo Luan a, Yan-Tao Zheng b, Meng Wang a,⇑, Tat-Seng Chua a

a School of Computing, National University of Singapore, Singapore 117590, Singapore
b Institute for Infocomm Research, Singapore 138632, Singapore

ARTICLE INFO

Article history:
Received 2 January 2011
Received in revised form 3 April 2011
Accepted 21 May 2011
Available online 30 May 2011

Keywords:
Interactive video retrieval
Relevance feedback
Active learning

ABSTRACT

This paper introduces an effective interactive video retrieval system named VisionGo. It jointly explores human and computer to accomplish video retrieval with high effectiveness and efficiency. It assists the interactive video retrieval process in different aspects: (1) it maximizes the interaction efficiency between human and computer by providing a user interface that supports highly effective user annotation and an intuitive visualization of retrieval results; (2) it employs a multiple feedback technique that assists users in choosing proper method to enhance relevance feedback performance; and (3) it facilitates users to assess the retrieval results of motion-related queries by using motion-icons instead of static keyframes. Experimental results based on over 160 h of news video shows demonstrate the effectiveness of the VisionGo system.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

The amount of multimedia information has been growing exponentially throughout the years. This is especially true for video data such as online videos, broadcast news, movies, and documentaries. Thus video retrieval has been an active and important research area since 1990s. The massive amount of video data requires intelligent systems which are capable of performing precise retrieval according to users' demand. Recently there is a shift from conventional text-based video retrieval to content-based or concept-based video retrieval that incorporates video content analysis as this is believed to better satisfy users' information needs [23,26,43,44,58,64–66].

The major challenge of video retrieval is to cope with the semantic gap [7,12] between raw signals and high-level semantics. This is because the inference of links between low-level features and high-level semantics is often a difficult task. In order to bridge the semantic gap, various multi-modal low-level features such as automatic speech recognition (ASR) text, audio and video features are introduced and extracted to give a richer representation of the video [11,21,41]. Furthermore, good high-level feature (HLF, or high-level concept) detectors need to be developed to add more semantics into the retrieval process and thereby improve retrieval performance [12,28,29,40,62,63]. However, existing video retrieval techniques that utilize ASR, audio/video features and HLF still cannot perform well enough for difficult queries. There are two major reasons for this: (a) the imprecision of the extracted feature representation and (b) the ambiguity of user queries. The low accuracy of some extracted features, especially from HLF, severely affects the retrieval performance. In addition, most existing retrieval systems require users to give well-specified queries for good performance. As compared to expert users, this is usually a very difficult task for general users [48]. Due to these reasons, fully automatic video retrieval provides relatively poor performance and is still far from becoming practical for accurate routine retrieval by most users demand.

⇑ Corresponding author. Tel.: +65 65164654.
E-mail addresses: luanhuanbo@gmail.com (H. Luan), eric.mengwang@gmail.com (M. Wang).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.05.018
Another available solution to reduce the semantic gap is through interactive video retrieval [3,22,42,51,52,57]. This is a process whereby the interactions between the user and system are used to enhance the performance. In such a process, a given query is first analyzed and used to retrieve an initial rank list of videos. Then the system permits the user to do some assessment over the set of video clips by specifying them as relevant or irrelevant to the query. This process can repeat for various iterations. The user’s reaction reflects subjective preference for knowledge demand over time. Based on the successive judgments on each cycle of the retrieval results, the original query will be gradually refined to achieve better retrieval results for the user or future queries. Such kind of feedback mechanism often includes the usage of common relevance feedback (RF) [34–36] or active learning techniques [13,14,49]. The main objective is to capture the user’s intention more precisely with user-supplied feedback and thus minimizing the semantic gap problem.

Interactive video retrieval has been established as a task of TRECVID [38,60], an annual international benchmarking campaign organized by National Institute of Standards and Technology (NIST) since 2001. The target of the task is to retrieve as many relevant shots as possible for a given open-domain query with the help of human interaction. The task definition which involves user interaction over a stipulated time frame of 15 min is shown in Fig. 1 from TRECVID guideline [61]. Current research on interactive video retrieval mainly follows the guideline of TRECVID. Generally, an interactive video retrieval scheme should contain two more components in comparison with automatic retrieval: (a) methods for effective using relevance feedback; and (b) information visualization which efficiently conveys the information content to the user. However, most systems generally focus on one direction and neglect the other. Besides, there are limited feedback strategies to choose, which forces most users to be in an impasse situation when a feedback strategy fails.

Catering to these problems, we propose an interactive video retrieval system named VisionGo. The system assists the interactive video retrieval process in different aspects: (a) it maximizes the interaction efficiency between human and computer by providing a user interface that supports highly effective user annotation and an intuitive visualization of retrieval results; (b) it employs a multiple feedback technique that assists users in choosing proper method to enhance relevance feedback performance; and (c) it uses motion-icon, a kind of short clips to summarize video information, such that the retrieval results of motion-related queries can be better visualized. To our knowledge, this is the first work that introduces an interactive video retrieval system with comprehensive details, including feature, indexing, ranking, interaction and interface. We emphasize that the employment of multiple relevance feedback and motion-icon in interactive video retrieval is novel.

Details can be found in the next section and in experiments we will validate them empirically.

The rest of the paper is organized as follows. The related work on interactive video retrieval is discussed in Section 2. Section 3 gives an overview on our interactive video retrieval system. Section 4 focuses on the inherent automatic retrieval engine, and Section 5 presents detailed techniques for the complete interactive process. Experiments are presented in Section 6, and we derive the conclusion in Section 7.

2. Related work

Compared with text retrieval, video retrieval is a much more complicated work. This is because it includes not only text retrieval but also image and video understanding [10]. Thus, interactive video retrieval has attracted great interests of many video research groups in recent years. Several state-of-the-art interactive video retrieval systems have emerged during the TRECVID evaluation competitions [60]. It is well recognized that, typically an interactive video retrieval system consists of three essential elements: (a) a well-performed automatic retrieval engine; (b) several effective feedback techniques; and (c) an efficient user interface (UI) for good interaction and visualization.

Some researchers have tried to investigate multi-modal features, like ASR, VOCR (video optical character recognition), and HLF, in automatic video retrieval [1,5,55]. Some other approaches have been studied on how to assist retrieval with semantic concepts and the optimal number of concepts for narrowing the semantic gap [5,24,28,39,50]. Re-ranking algorithms [17,18] have also been proposed in order to boost retrieval performance. For example, Hsu et al. [19] introduced a shot re-ranking retrieval algorithm on the basis of a rigorous information bottleneck principle. Yan and Hauptmann [56] brought forward a boosted re-ranking algorithm called co-retrieval to automatically select the most useful features from multiple modalities of video retrieval. Besides these interesting research works, others devoted themselves to multimedia-based ontology construction and reasoning for video retrieval. One good example is [54]. The authors presented a novel model, namely ontology-enriched semantic space (OSS), to provide a computable platform for modeling and reasoning concepts in a linear space.

Different from automatic retrieval, interactive retrieval allows the system to learn user’s information requirements through an interactive way. Hereinto, relevance feedback (RF) [34,35] is an important tool to improve the performance of
content-based information retrieval. RF was first proposed in content-based image retrieval (CBIR) and since then various relevance methods have been developed. The main idea of relevance feedback is for the system to understand the user's information needs [53]. The early approaches to RF belong to a category of “query point movement”, which is performed by finding a better query point as close as possible to the assumed ontological “ideal query point”, together with adjusting the weights of various features [6]. Further analysis on the density of positive feedback examples is interesting as well [2]. However, these methods only work well under certain limitations or strong assumptions. In general, RF can be viewed as a particular type of pattern classification that regards positive and negative samples as two different groups. Thus RF in CBIR is now an online learning problem. Many learning algorithms are applied within RF loop or cycles [8,15,46]. Among these learning methods, Support Vector Machines (SVM) based RF [15,59] is the most popular method because of its inherent advantages such as fast learning, multi-chains of kernel, and relying on only support vectors. However, SVM based RF usually meets with the following challenges in interactive video retrieval: (a) the amount of positive samples are not sufficient and the SVM classifier is unstable with small-size training set; (b) the imbalance between positive and the negative samples; and (c) feature dimension is usually greater than the number of training samples. To solve these problems, the authors in [47] designed a new asymmetric bagging and random subspace mechanism for SVM based RF. Besides, semi-supervised learning and active learning are also investigated. Semi-supervised learning is an approach that tries to learn a classification model from the mixture of labeled and unlabeled examples [14,45]. Another solution is active learning [13,14], which chooses the most informative unlabeled examples for manual labeling. Tong et al. [49] presented the use of SVM active learning with several sampling strategies for conducting RF. In spite of its promising performance from previous studies, semi-supervised learning suffers from high computation cost.

For effective interaction, it is not enough to only make good use of RF techniques. A well designed interface for communication between the users and the system is also important. CMU and MediaMill’s successes demonstrate that efficient UI is crucial for interactions which jointly maximize the performance of both human and computer. Hauptmann et al. [10] proposed an extreme video retrieval system that maximizes only human annotation efforts. They introduced an efficient interface and an extensive annotation strategy based effectively on human reaction time and set a surprising record of annotating 5000 shots in 15 min. The main drawback of their system is that it cannot supply good visualization for users. Rooij et al. [33] focused on building an elaborate interface with flexible display capabilities to interact with the users. They proposed several impressive interfaces such as CrossBrowser, RotorBrowser and ForkBrowser, which give intuitive visualization so as to make it easy for users to obtain video structure and visual content. With these efficient interfaces, these groups achieved good performance in TRECVID evaluations. Despite encouraging performance has been shown, they do not provide enough feedback techniques to refine the retrieval engine which can further enhance the performance. Furthermore, their user interfaces usually focus on either visualization or annotation efforts, failing to combine the two main aspects that can maximize the user’s interaction efforts.

Through the studies on past interactive video retrieval systems, we can summarize that there are still challenges that need to be solved. First, most interfaces focus on either annotation efforts or visualization. Focusing only on one aspect is insufficient to maximize interaction efforts. Second, feedback strategies are too limited to support effective refinement for the base retrieval engine. Some systems require users to do mainly annotation throughout the whole retrieval process and provide no feedback techniques to assist user’s intelligence or initiative. Third, current systems with feedback generally let users judge only the static keyframes for video shots. However, a video consists of many frames and the static keyframes itself is not sufficient. This is especially true for queries related to action or motion. Static keyframes are obviously not able to show motion information for them. The proposed VisionGo system can tackle these problems by emphasizing on: (a) designing an intuitive user interface, which combines good visualization and user’s annotation efforts to maximize the interaction efficiency between human and computer; (b) providing multiple feedback techniques to assist users in choosing the proper way for enhancing retrieval performance in real time; and (c) supporting motion-icons, which contain motion information to handle motion queries. Details of these strategies are given in the later sections.

3. Framework

3.1. Overall framework for interactive video retrieval

The overall framework of our interactive video retrieval system is shown in Fig. 2. There are two main stages: the automatic retrieval stage and the interactive retrieval stage. The retrieval starts with the user’s query, which could be a free text query or a combination of text and image/video examples (multimedia query). The automatic retrieval will perform query analysis and multi-modal fusion to retrieve a ranked list of results deemed most relevant to the query, where the multi-modal fusion uses a combination of ASR, HLF, low-level visual and motion features. The user then screens the returned results and indicates whether they are relevant through a well-designed interface, which provides an intuitive visualization to maximize user's annotation efforts. This purpose is to provide enough positive and negative samples to refine the query and improve the subsequent searches. At this stage, the user can interact with the system to choose which kind of feedback strategy to use in different situation. Meanwhile, the proposed motion-icons provide users a better idea of motions and content variations within the shots. Thus this interactive stage enhances the retrieval performance by maximizing the interaction between the user and the system.
Video retrieval has primarily started as a form of text-directed retrieval based solely on ASR text. ASR is extraordinarily important in news video as video speech at which the news event is conveyed usually describes the essential context where an event is occurring. With advances in image and video processing, recent approaches attempt to integrate various multi-modal features, including low-level visual features (e.g., color, edge, motion) and high-level concept features (e.g., trees, fire, explosion and applause), to support more precise retrieval. The available visual features can complement with ASR to improve the retrieval performance for a wider range of queries. Our used multi-modal features detail as follows.

3.2.1. Text (video speech)
ASR is extracted from the audio stream of news video. For non-English videos, an additional step of automated machine translation (MT) is taken. Even with good quality speech utterances from news video, it is well known that ASR seldom has a perfect performance; and MT output is likely to be even more erroneous due to the unavoidable translation errors. We further extract known named entities (NE) such as time, date, location, subjects and activities from text at story level. NEs have been found to be good descriptors especially for news. This is because news reports are events in nature; and events are usually location-centered, subject-centered or activity-centered. Although shot boundary detection performance is already fairly good, we choose to align text to pseudo story [28] as they allow higher coherency.

3.2.2. High-level features (HLF)
HLF denotes a set of predefined concepts: (a) objects like cars, buildings; (b) audio-genre like cheering, silence, music; (c) shot-genre in news like political, weather, financial; (d) person-related features like face, people walking, people marching; and (e) scenes like desert, vegetation, and sky. For example, a retrieval on “fire” can be retrieved precisely at shot level if “fire” is one of the available HLF. To determine the presence of certain HLF in news video, machine learning is the most typical approach to learning a detector for each HLF based on an annotated corpus of video clips. Such features are extremely useful as they provide additional semantics which are usually not available from text.
3.2.3. Low-level visual features

Low-level visual features such as color, texture, edge, motion and SIFT points are useful for finding similar keyframes through the use of image matching techniques. One problem concerning the usage of too many visual features is the curse of high dimensionality, which may degrade performance and introduce large time cost. In order to cater to real-time retrieval, we restrict the feature size to a 116-feature vector for each key frame, consisting of 27-dimension color moment features (including 1st, 2nd, and 3rd moments) obtained at a $3 \times 3$ block, 80-dimension normalized local edge histogram texture feature, eight directional motion features and one global motion feature [25].

4. Automatic retrieval component

As an essential part of interactive video retrieval, a well-performed automatic retrieval engine can supply a high-quality initial rank list for the next feedback step so that it could attract much attention. We employ our automatic retrieval techniques based on our previous work [5] to perform the initial retrieval. We focus on multimedia query analysis and two-level retrieval (pseudo story retrieval and shot level re-ranking). The overall framework is highlighted in Fig. 3.

4.1. Multimedia query analysis

Besides effective features, query analysis and formulation are also essential as they help in understanding users’ intention. Queries on news are often time-sensitive, featuring new persons, events, corporations and entities. It is therefore necessary to induce additional query context from original user’s query in order to retrieve most relevant answers. We induce and extract query information like Query-term, Query-class and Query-HLF from the text query; as well as Query-HLF and image level features named Query-ImageVector from the visual counterparts if available. This query-information is then used in pseudo story-level retrieval and shot-level re-ranking.

**Query-term** $Q_T$, which provides the necessary initial context for retrieval, is employed. As user’s text query is usually short and imprecise, it is necessary to carry out inference to gather extra context. We therefore propose query expansion by generating additional query terms in the following 2-step approach [30]: (a) using the original query to retrieve relevant news articles from parallel news corpus, and (b) extracting terms from relevant news articles which have high mutual information (MI) with the original query-terms. For example, queries like: “Find stories related to flood” or “Find shots containing building covered in flood water” will extract highly related terms like “rain, flood, hurricane, yellow river ...”

**Query-class** $Q_C$ is another important feature, as it has been shown in many previous works as a guide to fuse multi-modal features effectively. This query-class is determined by a set of filtering rules for each class: {PERSON, SPORTS, FINANCE, WEATHER, DISASTER, HEALTH, POLITICAL, MILITARY, GENERAL}. The GENERAL-class is created to accommodate queries that do not belong to any of the first eight classes. The main reason for this classification scheme is to create an explicit mapping of query-class for video program-genre. For example: the answer shots for sports questions are normally found in sport news; and it is the same case as far as financial news and weather news are concerned. These nine classes are also chosen because they can be easily classified by using simple heuristic rules based on textual information alone. This is important as it is impossible to perform complex query classification for short text queries. In addition, we allow a user’s question to be classified into a maximum of two classes. This is reasonable as there are questions which can belong to more than one class. One such typical question is: “Which states are affected by tornado?” can be classified into WEATHER or DISASTER.

**Query-HLF** $Q_{HLF}$ on the other hand, suggests possible HLFs that are important to the query in terms of visual requirements. It is clear that a visual-oriented query like: “Find shots of George Bush speaking” requires substantial visual evidences in the shot which cannot be achieved with only text features. We approach this by employing morphological analysis.
followed by selective expansion using the WordNet lexical database on both the feature descriptions of HLFs and user’s query [28]. The stronger the match between the HLF descriptions and the query, the more important the HLF is to the query. Besides relying on lexical relationship, we also infer query-HLF from sample images when they are available from users’ multimedia queries. A sample image containing one of the HLFs explicitly means that the particular HLF could be important.

**Query-Image Vector** $Q_{IMG}$ is the corresponding set of image-level features extracted from sample images. It follows the 116-feature vector as described in Section 3.2.

4.2. Retrieval process

4.2.1. Pseudo story retrieval

It employs similarity matching by using query terms on the ASR text of news video. Here we use pseudo story-level segment as the basic retrieval unit where the pseudo story boundaries were determined using the heuristics outlined in [16]. The text score of a particular story, $V_d$, with respect to query $Q$ was computed using the vector space model [37] as shown in Eq. (1).

$$\text{Sim}_{\text{text}}(Q, \text{story}_d) = \frac{q \cdot \mathbf{v}_d}{||q|| \cdot ||\mathbf{v}_d||} \tag{1}$$

where $q$ is the vector of terms in query, and $\mathbf{v}_d = [w_{1,d}, w_{2,d}, \ldots, w_{c,d}]^T$ denotes the terms appear in ASR text of story $d$ with $w_{c,d} = tf \cdot \text{idf}$. $tf$ refers to the term frequency of term $w$ in the document and $\text{idf}$ is the inverse document frequency of $t$ which indicates how “rare” is the term in the whole set of stories. The resulting similarity yields a value of between 0 (no similarity) and 1 (identical) and it is used for ranking stories. Note that a story may contain multiple shots.

4.2.2. Shot re-ranking

As multiple continuous shots are returned during story retrieval, it is necessary to identify shots relevant to user’s query by performing shot-level analysis. Here we employ Query-HLF and the Query-ImageVector of each shot are employed as visual-oriented features to analyze the contents of shots. These visual-oriented features, in conjunction with text similarity derived from Eq. (1), are used for shot re-ranking as shown in Eq. (2).

$$\text{Sim}_{\text{shot}}(Q, \text{shot}_j) = \alpha_{c} \cdot \text{Sim}_{\text{text}}(Q, \mathbf{v}_n|\text{shot}_j \in \mathbf{v}_n) + \beta_{c} \cdot \sum_{\text{HLFm} \in \text{shot}_j} \left[ \text{conf}(\text{HLF}_m) \times \text{Sim}_{\text{HLF}}(Q, \text{HLF}_m) \right] + \delta_{c} \cdot \max(\text{img}_\text{sim}(Q_{\text{image}}, s_j)) \tag{2}$$

The first term makes use of the story retrieval function from Eq. (1); the second term measures the similarity of high-level feature scores [28], and the third term measures the image similarity using the low-level visual features. Here $\text{img}_\text{sim}()$ computes a normalized Euclidean distance between the feature vector of low-level visual features extracted from the available query-images and the shot keyframes. $\alpha_c$, $\beta_c$ and $\delta_c$ are weights used for combining the three similarity values with $\alpha_c + \beta_c + \delta_c = 1$. As with [27], the values of $\alpha_c$, $\beta_c$ and $\delta_c$ are Query-class dependent, determined using EM training from ground-truth results under in Query-class (for more details, see [27]). For illustration, the weights for the queries belong to the PERSON class are set to: $\alpha_c = 0.7$, $\beta_c = 0.2$, $\delta_c = 0.1$. Note that the scores of text entities are much higher than high-level visual features since story retrieval tends to invoke contextual aspects for the PERSON class. The similarity measure Eq. (2) is used as the basis to automatically retrieve a rank list of shots to the users.

5. Retrieval with user interaction

To solve the above mentioned problems of current systems, our proposed VisionGo system is designed to maximize the effectiveness of human annotators through the use of: (a) effective UI (User Interface); (b) options for multiple feedback strategies; and (c) motion icons for previewing key motion characteristics in videos. We are emphasizing not only retrieval performance but also user’s annotation efforts, advanced visualization for a good interface and the influence of various interaction mechanisms. The system first provides user the results from the automatic retrieval for initial feedback. After that, the user can make use of our retrieval interface to refine the retrieval results with a variety of relevance feedback techniques. Through motion-icons, users are also able to see a dynamic series of keyframes instead of a single static keyframe during relevance assessment. This helps them in well handling motion-related queries.

5.1. Interface for video annotation

Most existing interactive UIs designed for traditional CBIR systems mainly aim at displaying a large number of keyframe images in a display area. By observing these keyframes, the user can make simultaneous annotation by mouse-clicking. A large number of dazzling images are displayed in a limited area at the same time, which make the user giddy and tired quickly and results in low-efficiency in annotation. Some interfaces such as CMU’s extreme video retrieval, and MediaMill’s various browsers have achieved great success in interactive video retrieval. However, they merely focus on maximizing either annotation efforts or visualization. Failure in combining the two aspects together makes these systems unable to
maximize the interactions between the users and the system and further maximizing the retrieval performance. Inspired by these novel interfaces, we have designed our efficient UI, as shown in Fig. 4, to combine these two factors, bearing in mind the balance between retrieval effectiveness and efficiency. The interface has two main characteristics: (a) the UI is designed for fast feedback based on keyboard hitting and fast updates with advance pre-viewing of previous and subsequent sets of shots in the ranked list; and (b) the UI contains several auto update/learning buttons to permit users to quickly try different learning or relevance feedback functions when there are few relevant shots to be found.

From the screen dump of our UI, user interface developed by Macromedia Flash is user-friendly and impressive. Similar with other systems, one keyframe represents its corresponding shot. The viewed keyframes with ticks show that these shots are judged as relevant while the viewed keyframes without ticks show that these shots are judged as irrelevant. For the rest of the candidate keyframes in the rank list, these shots are still not judged. Some UI design bases are explained as follows:

5.1. Fast perception and quick previews

Our design displays three images at a time in a central active row, while supporting advance pre-viewing of previous and subsequent sets of shots in the ranked list to facilitate fast updates. We have carried out the experiment with various configurations of display and found that the quickest user reaction time appears when annotating three images at a time. The purpose is to eliminate both unnecessary eye movements and to avoid too many dazzling images in a limited area, which makes the user confused and disoriented. As a result, the user can locate his eyes on three candidates in the central active row and determine the keyframes' relevance to the query in the most rapid way. With the help of pre-viewing of previous and subsequent sets of shots, the user is able to make a timely modification or capture the visual content of next set of shots in advance. The user will determine these three images' relevance to the query and annotate the positive ones. After accessing current row, the system captures the user’s input and automatically refreshes itself to display the next row of new keyframes in the rank list.

5.1.2. Fast keystroke actions

CMU’s success in Extreme Video retrieval demonstrates that the operation speed of keyboard-hitting is much faster than that of mouse clicking. In addition, keystroke can help to maximize the human annotation throughput. Likewise, it has also been found in many online game interfaces, in which keyboard provides fast interaction modes. This is the mode preferred by almost all young gamers. Inspired by these experiences, our system prefers keystroke actions over the traditional mouse-clicking. The user performs the annotation by hitting the pre-defined keys on the keyboard, as shown in Fig. 5. The buttons “F” and “D” are designed to control the forward or reverse scrolling for browsing the rank list of keyframes. The buttons “J”, “K”, and “L” are assigned for labeling the three corresponding images in the active display row, the pressing of which means the corresponding keyframe shot is labeled as positive. Besides, some hot keys have been predefined to perform certain
essential feedback functions quickly instead of clicking the function button. For example, in a case where no image is relevant to the query, the user can hit the “Space” key to skip a row. Meanwhile, the “Space” can also be pressed and held to “fast forward”. Alternatively, the “Backspace” key is used to undo changes and also backtrack when the user needs to perform corrections. But it is worth noting that the fast keystroke actions are used only in the user interaction process.

5.1.3. Flexible annotation modes

Because users have different labeling habits, e.g., some users think that scrolling by pressing assigned keys is too boring especially when the retrieval stage lasts a long time. In order to fit different users’ different labeling habits and to achieve considerable annotation throughput, we provide three types of scrolling modes as follows:

- **Manual-scroll**: Scrolling up images by hitting some assigned keyboards;
- **Semi-scroll**: Scrolling up images automatically and immediately in an adjustable interval after three images in the current row are labeled;
- **Auto-scroll**: Scrolling up images automatically in a pre-defined static speed, being able to be adjusted for different users.

5.1.4. Retrieval statistics

We designed a retrieval statistics panel in our UI to display real-time running information of the system. By experiments, we find that it is a very necessary and helpful design for users to learn the exact system running status and estimate the approximated performance so as to adjust his feedback strategy and to further improve the retrieval performance.

5.1.5. Modular design

Another characteristic is our UI is modular, which is entirely separated from the retrieval engine. The retrieval engine can be installed in a host sever and the UI can be used in any place to perform the retrieval, as long as they are connected over any form of network. This separated design enables the interactive retrieval system to be easily used in different settings and operating environments. Besides, the UI developed by Macromedia Flash can be easily embedded into the internet browser, which is in line with current trends in web search systems.

5.2. Multiple feedback strategies

In order to improve the accuracy of video retrieval, many existing systems have applied certain forms of automated feedback methods such as pseudo relevance feedback [31]. However, most systems usually rely on only a specific type of relevance feedback method. Due to the complexity and variety of multi-modal features in videos, it is usually insufficient to apply a single feedback method. There is no single method which could be equally good or generic across different video corpuses or for different retrieval queries. Therefore, we segregated feedback into three parts as the following:

5.2.1. Recall-driven relevance feedback

Given a large corpus, re-ranking in the later stages might be redundant if relevant shots are not presented in the initial retrieval. Hence, in order to maximize recall performance and minimize computational complexity, we choose to employ feedback using general features such as text and HLF. First, our recall was enhanced by representing each shot with an entire ASR phrase segment which overlaps across shot boundaries on a time scale. These phrases usually contain coherent
obtained in [28]. The feedback process comprises of three steps:

Step 1: Given the set of positively annotated shots \( N_p \), the feedback process utilizes the text and HLF scores to iteratively adjust the retrieval function. The text from \( N_p \) will first be analyzed to select highly discriminating text tokens for retrieval. We employ the 0.5 formula [32] as shown in Eq. (3) to select the top \( k \) terms which are most salient.

\[
FS(\text{term}_l) = \log \frac{(r + 0.5)(N - n - R + r + 0.5)}{(n - r + 0.5)(R - r + 0.5)}
\]  

where \( N \) is the number of shots in the collection, \( R \) is the number of shots found to be relevant to the query, \( n \) is the number of shots containing term \( l \), and \( r \) is number of relevant shots containing term \( l \).

Step 2: The HLF scores for each labeled shot is used to estimate the new \( Q_{HLF}^{(1)} \), which is the new relevance of HLFs to the query. This is done by averaging the detection confidence of each HLF in the \( N_p \) relevant shots as shown in Eq. (4).

\[
Q_{HLF}[1..50] = \frac{1}{N_p} \sum_{i=1}^{N_p} S_i^{HLF}[1..50]
\]

where \( S_i^{HLF}[\cdot] \) is the detection confidence of various HLFs in shot \( i \). This detection confidence of HLFs can be taken as appearance likelihood. If many positive shots have high scores for certain HLF, it implies that the query is closely related to that HLF.

Step 3: After obtaining the text and HLF scores, we compute the new scores for each individual shot by using Eq. (5).

\[
\text{Score}(S_i) = \lambda \cdot \frac{1}{k} \sum_{i=1}^{k} (FS(\text{term}_l)|\text{term}_l \in S_i) + (1 - \lambda) \cdot \frac{1}{50} \sum_{j=1}^{50} (Q_{HLF}^{(1)}|j \cdot S_i^{HLF}[j])
\]

where \( \lambda = [0..1] \) is set according to the importance of text or HLF for a particular query. In our experiments, \( \lambda \) is empirically set as 0.7 and adjusted accordingly by calculating the standard deviation \( SD \) of \( Q_{HLF}^{(1)} \). A high \( SD \) will signify that the query shows reliance to certain HLFs. Alternatively, if the \( SD \) is low, it means that there is low correlation between HLF and the query.

5.2.2. Precision-driven active learning

To complement the high-recall feedback, active learning based on multimodal features is also carried out on a subset of retrieved shots (set to 3000 in our experiments). The aim of active learning is to improve the performance of SVM classifier and hence the precision of retrieval. We choose to use Support Vector Machine (SVM) for active learning since our problem can be simplified into a simple probabilistic binary classification problem (either positive or negative), and SVM is able to perform this task with high efficiency. The selected visual features consist of the 50 HLFs used in Section 2, color moment features (1st, 2nd, 3rd) obtained at a 3 \times 3 block, eight directional motion features and one global motion feature. The overall dimension of the feature space is 86.

Most active learning algorithms sample instances close to classification boundaries to judge as this will enable classifiers to converge faster. However, in normal retrieval circumstances, searchers are more interested in finding correct shots in the shortest time. This implies that shots that are deemed more relevant should be presented rather than irrelevant shots. This is in particular reasonable for most retrieval tasks that have a strict time constraint. To handle this problem, we propose an adaptive sampling strategy based on the classification output quality. The system will first sample instances that are far from the boundaries (those deem to be most relevant) instead of instances near the boundaries, and if many of these instances are indeed marked positively by the user, the system will continue to sample in that particular region. On the other hand, if these samples are mostly negative, the system will then switch to sample more instances near the classification boundaries. An adaptive sampling strategy based on this rationale is developed to allow users to have quick access to highly probable relevant shots. In addition, it provides fast convergence to deliver more accurate classification. The outline of the strategy is shown in Fig. 6.

5.2.3. Temporal locality-driven feedback

An intuitive method which has been shown to be efficient and effective for video retrieval is exploring temporally related shots [4]. The main idea is, video stories usually span across several shots and there is high semantic coherence among the temporally neighboring shots. For example, in a video story of “George Bush on Iraq War”, George Bush may appear in the first shot, followed by some irrelevant shots and back to him again. Fig. 7 shows an example of such an occurrence.

In long video story case, it is common that the consecutive shot series are relevant if one of the shots is relevant. For example, Fig. 8 illustrates that there are many shots containing snow in a news video story on snowstorm.

Temporal locality-based feedback is the process of automatically learning in which nearby shots are returned when a particular shot is judged as relevant. Since video story is a segment of video with coherent focus, we use the story boundaries obtained in [20] to constraint the feedback process. As shown in Fig. 9, the story constraints forbids the return of any shot,
such as Shot L2 in Fig. 9 which is outside the story boundary of the video that contains the marked relevant shot. The intuition here is that if a shot is relevant, only other nearby shots that belong to the same video story are likely to be relevant. The only exception occurs when neighboring video stories are about the same topic. As this is a rare exception in our corpus, it shall only be investigated in our future work.

Given the story boundaries and restrictions, for each marked relevant shot, we attempt to select temporally adjacent shots left and temporally adjacent shots right. This process is similar to the usage of storyboards in some existing systems where a fixed number of shots from the left and right are returned [4]. Manually setting L and R is particularly useful when the user is an expert in video retrieval. However, for general public usage, this may not be the most effective way. It is often very hard for normal users to decide L and R when they first use the system. Furthermore, the importance of left neighboring shots and right neighboring shots may not be the same. Setting a number too high may result in returning too many shots to be judged and waste the time which can be spent in judging shots using other techniques. On the other hand, setting a too low number, such as 1, may not be sufficient as video stories may often span across more than three shots. In our system, we determine L and R for normal users based on the mean average precision (MAP) on 24 questions in the TRECVID 2006 corpus (the corpus will be detailed in Section 6). Three expert users are asked to use our system based only on locality-driven feedback for the interactive retrieval task using various values of L and R. When evaluating the effectiveness for various values of L, we set R to be zero and vice-versa. From experiments, we found that L and R work best when they are set to 2 and 3, respectively. Using these values of L and R, our system returns all the neighboring shots of the k marked relevant shots in each iteration. Note that these values of L and R are applicable to the news video corpus only.

Based on these multiple feedback strategies, the user will be able to choose the type of feedback that is more suitable based on his/her intuition or experience to maximize the performance.

5.3. Motion-icon

Current systems generally let the user judge the static keyframes for video shots as relevant or irrelevant to the performance of feedback mechanism. However, there is no motion information contained in static keyframes, which makes it impossible for these systems to handle action or motion-related questions. What is more, even if the query is only "static" query unrelated to motion, the mere assessment of candidate shots based only on a static keyframe may also suffer from the problem that the single image may be wrongly chosen as the keyframe, which cannot well present the overall visual content of this shot.

To handle these two scenarios, it is therefore necessary to provide some information on motion in the icon to facilitate the annotation process. Specifically, instead of displaying an icon with static keyframe for each video shot, we construct a
summarized clip comprising a sequence of progressive keyframes which can show moving picture information. We call such icon the motion-icon. To construct a motion-icon for a shot, we first extract motion features in this shot including local and global features as introduced in Section 2. Those motion features are then analyzed to get the distribution of motion density within a shot, which can help to determine where the motion changes lie. A high motion density value means there is an obvious motion change. We select top five temporal locations of high motion density to extract the corresponding images and construct a summarized clip (motion-icon) using the five images across a timeline.

Through the use of motion-icon, the user could gain a clearer idea of what motion information is in the shot and can identify relevant shots more quickly with better confidence. For example, for retrieval topic 0197 in 2007 on “Find shots of one or more people walking up stairs”, if users were to be presented only with a single frame bounded in red box as shown in Fig. 10, this shot would have been considered irrelevant. However, through the use of motion-icon, we can determine straight away that this is a relevant shot.

The use of motion-icon not only helps in detecting the correct motion in a shot, but also helps in situations when improper keyframes are chosen for the shot. For example, for a non-motion oriented query “Find shots of a canal, river, or stream with some of both banks visible”. The keyframe shown in Fig. 11 will be deemed as irrelevant. However, through the use of
motion-icon, we can assess that the shot is relevant to the query. The tradeoff in using motion-icon is that the display speed becomes relatively slower.

6. Experiments

To evaluate the effectiveness of the proposed VisionGo system, we employ the TRECVID 2006 dataset, which consist of 160 h of English, Chinese and Arabic news videos, and follow the same evaluation methodology as in the interactive retrieval task. This TRECVID dataset is one of the largest and most widely used dataset for news video retrieval performance testing. In TRECVID 2006 interactive retrieval task, 24 queries were designed and users were given a 15-min limit for each query to evaluate the retrieval performance. In this paper, we divide the experiment into three tasks to evaluate the effectiveness of different retrieval functionalities. The first test (Test 1) evaluates the labeling speed with the use of our interface followed by a general study of user evaluation experience. The second test (Test 2) evaluates the performance of various feedback techniques as described in Section 5.2. Finally, Test 3 further experiments with the use of motion-icons to enhance retrieval accuracy.

6.1. On labeling speed and user perception

Test 1 evaluates the human annotation efforts with our designed UI. Three expert users, which are familiar to our UI, evaluate the labeling speed on 24 topics without any feedback techniques. The experiment was performed twice by each user; one is based on static keyframes while the other is based on motion-icons. The average results from three users are shown in Fig. 12, from which we can see that the UI enables the user to annotate about 3000 shots based on motion icons or 4500 shots based on static icons for most queries within 15 min. It clearly shows that the UI does help the user to maximize the annotation efforts which are faster than the reported labeling record from CMU [9]. We also engaged two novice users of our system to perform the annotation evaluation. Thirty minutes were given to the user for training before the evaluation. We found that they are able to annotate about 3300 shots when using static icons and 2300 shots with motion icons, respectively. This performance is not very far from expert users and re-affirms the feasibility of our UI.
On further analysis, we find that the number of reviewed shots varies observably on different topics, especially for static keyframes. For example, over 6000 shots are annotated in 15 min on the topic “Find shots of something burning with flames visible”. However, some topics take more time to review than others, such as the topic “find shots with one or more emergency vehicles in motion”. We conjecture that this is because users take more time to evaluate the relevance of shots in this query because of inability to judge the occurrence of motions. This further reinforces the need for motion icons for such motion oriented queries. According to the distribution of labeling shots, 24 topics were divided into four query classes (Scene, General, Person, and Action) as shown in Table 1. From Fig. 12, we can summarize that the users labeled the most shots on the topics belonging to “Scene”. The topics belonging to “Person” and “Action” are more time-consuming.

The next experiment highlights the benefits of interactivity made possible with the use of VisionGo system. The system was evaluated by 20 students from the faculty. These students were all first time users of VisionGo and were allowed to issue any one from these 24 queries to test the system. In addition, each student was asked to evaluate above three aspects: labeling speed, multiple feedback strategies and motion-icons. At the end of the experiments, the students were required to fill the assessment form containing the five questions that are shown in Table 2. Each question requires a numerical answer based on the following scale (1 – Strongly Disagree, 2 – Disagree, 3 – Neutral, 4 – Agree, 5 – Strongly Agree). The results were tabulated in Table 2. Based on the assessment of the users from Questions (1) and (2), we can see that the well-designed interface is very intuitive and popular by users. It provides a maximal interaction between the user and the system. The user agreement test based on Questions (3) and (4) shows that multiple feedback functions and motion-icons are effective in improving the quality of the retrieval results, and offering enhanced interactivity experience to the users. In addition, we see that at least 85% of the first time users to VisionGo report more than satisfactory searching experience.

### 6.2. On multiple feedback strategies

Assessing the speed of labeling is not sufficient to deduce the effectiveness of the system. We need to determine the actual precision or recall so as to understand the improvements brought about by the use of the multiple feedback strategies. In this section, we use the standard evaluation protocol set by TRECVID. The system will return a maximum of a 1000 shots for each query. The result is then automatically evaluated using the ground truth provided. For the performance measurement, we follow the use of mean average precision (MAP), which is suitable for information retrieval evaluation over large corpuses where recall rate is hard to determine. Six runs are designed to evaluate the performance of each strategy as well as their combinations. This experiment is performed with only the static keyframes.

- **S1**: Auto retrieval (baseline)
- **S2**: S1 + user’s annotation only
- **S3**: S1 + Precision-driven feedback
- **S4**: S1 + Recall-driven feedback
- **S5**: S1 + Locality-driven feedback
- **S6**: S1 + Multiple feedback strategies

### Table 1
Several query classes for 24 topics.

<table>
<thead>
<tr>
<th>Scene</th>
<th>General</th>
<th>Person</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tall buildings</td>
<td>Escort prisoner</td>
<td>Dick Cheney</td>
<td>Emergency</td>
</tr>
<tr>
<td>Water + boats</td>
<td>Day protest</td>
<td>Saddam Hussein</td>
<td>Enter/exit vehicle</td>
</tr>
<tr>
<td>Nature</td>
<td>Uniform + formation</td>
<td>Condoleezza Rice</td>
<td>Bush walking</td>
</tr>
<tr>
<td>Flames</td>
<td>Soldiers weapons</td>
<td></td>
<td>Helicopter flying</td>
</tr>
<tr>
<td>Smoke stack</td>
<td>People + computer</td>
<td></td>
<td>Cheek kiss</td>
</tr>
<tr>
<td>Soccer goalpost</td>
<td>People + newspaper</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Snow</td>
<td>Suits + flags</td>
<td>Person + books</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Adult + child</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2
Survey results by 20 students on VisionGo.

<table>
<thead>
<tr>
<th>Assessment type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) The UI is impressive and quite easy for me to use. I have no problem understanding the retrieval interface</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>(2) The UI can help me maximize the annotation efforts to enhance my searching</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>(3) The provided multiple feedback functions are quite effective and obviously improve the retrieval results</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>(4) The quality of the retrieved results using motion-icons is better than that of results using static keyframes</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>(5) The VisionGo system exceeds my video retrieval expectation</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>
The brief descriptions of the runs are as follows. S1 is the result of the automated retrieval, which is done by collating the top 1000 returned shot. This will establish the baseline performance for automated retrieval. For S2, we focus on only leveraging user efforts. In this case the user will attempt to label as many shots as possible without leveraging any form of feedback. The labeled relevant shots are re-ranked to the top of the result list for MAP computation. S2 can be viewed as a baseline run for interactive retrieval. From S3 to S5, we allow the user to use different forms of feedback strategies. Finally, S6 allows the user to flexibly choose between strategies. The results are tabulated in Table 3. From the results, we can clearly see the distinction in results especially between S1 and the rest of the interactive runs. In comparison, the runs which employ a human annotator perform 3–4 times better than the automated results. This is because the positive shots labeled by the user are being returned as the top shots, causing a jump in the precision of the results. In comparison to S2 which does not use any form of feedback, Runs S3–S5 show better performance. S3 performs slightly better than S2 while S4 and S5 yield significant improvement. This implies that the use of feedback strategies is important. In particular, S6 which uses all three feedbacks can achieve significantly better results than its counterparts. The MAP performance of S6 is also statistically better than the best reported interactive retrieval run in TRECVID 2006 (with a MAP of 0.308).

In addition, we observe that as we improve the search techniques from S1 to S6, the MAP increases while the number of judged samples decreases. This happens naturally as more positive shots are presented to the user for judgment and users tend to take more time to confirm a positive shot. To further study the performance of various runs against interaction time, we plot the MAP performance of runs S2–S6 by using the rank list generated at fixed time interval as shown in Fig. 13. From Fig. 13, we can see that S6 is the best run in term of efficiency. It only requires 8 min of user interaction time to attain the same MAP performance as S4 which has a recorded MAP of 0.276 after 15 min of user interaction. We also observe the MAP performance of S5 is higher than S4 during the first few minutes and subsequently become almost standstill, which signifies the depletion of positive instances through the use of neighboring shots alone.

6.3. On the effectiveness of motion-icon

While we see that improvements brought about by the use of multiple feedback strategies, this section aims to evaluate how the motion-icons could further enhance annotation performance. We re-create the same five runs as in the previous section, by using the motion-icons instead of the static keyframe for user annotation. The results are tabulated in Table 4.

**Table 3**
Performance of baselines and runs based on different feedback strategies.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total shots judged</td>
<td>Auto</td>
<td>4001</td>
<td>3822</td>
<td>3512</td>
<td>3555</td>
<td>3127</td>
</tr>
<tr>
<td>Positive shots found</td>
<td>63</td>
<td>116</td>
<td>131</td>
<td>147</td>
<td>143</td>
<td>158</td>
</tr>
<tr>
<td>MAP</td>
<td>0.079</td>
<td>0.218</td>
<td>0.223</td>
<td>0.276</td>
<td>0.265</td>
<td>0.328</td>
</tr>
</tbody>
</table>

**Fig. 13.** MAP performance variations with respect to interaction time of different methods.

The brief descriptions of the runs are as follows. S1 is the result of the automated retrieval, which is done by collating the top 1000 returned shot. This will establish the baseline performance for automated retrieval. For S2, we focus on only leveraging user efforts. In this case the user will attempt to label as many shots as possible without leveraging any form of feedback. The labeled relevant shots are re-ranked to the top of the result list for MAP computation. S2 can be viewed as a baseline run for interactive retrieval. From S3 to S5, we allow the user to use different forms of feedback strategies. Finally, S6 allows the user to flexibly choose between strategies. The results are tabulated in Table 3. From the results, we can clearly see the distinction in results especially between S1 and the rest of the interactive runs. In comparison, the runs which employ a human annotator perform 3–4 times better than the automated results. This is because the positive shots labeled by the user are being returned as the top shots, causing a jump in the precision of the results. In comparison to S2 which does not use any form of feedback, Runs S3–S5 show better performance. S3 performs slightly better than S2 while S4 and S5 yield significant improvement. This implies that the use of feedback strategies is important. In particular, S6 which uses all three feedbacks can achieve significantly better results than its counterparts. The MAP performance of S6 is also statistically better than the best reported interactive retrieval run in TRECVID 2006 (with a MAP of 0.308).

In addition, we observe that as we improve the search techniques from S1 to S6, the MAP increases while the number of judged samples decreases. This happens naturally as more positive shots are presented to the user for judgment and users tend to take more time to confirm a positive shot. To further study the performance of various runs against interaction time, we plot the MAP performance of runs S2–S6 by using the rank list generated at fixed time interval as shown in Fig. 13. From Fig. 13, we can see that S6 is the best run in term of efficiency. It only requires 8 min of user interaction time to attain the same MAP performance as S4 which has a recorded MAP of 0.276 after 15 min of user interaction. We also observe the MAP performance of S5 is higher than S4 during the first few minutes and subsequently become almost standstill, which signifies the depletion of positive instances through the use of neighboring shots alone.

6.3. On the effectiveness of motion-icon

While we see that improvements brought about by the use of multiple feedback strategies, this section aims to evaluate how the motion-icons could further enhance annotation performance. We re-create the same five runs as in the previous section, by using the motion-icons instead of the static keyframe for user annotation. The results are tabulated in Table 4.

**Table 4**

<table>
<thead>
<tr>
<th></th>
<th>M2: S2 with Motion-icons</th>
<th>M3: S3 with Motion-icons</th>
<th>M4: S4 with Motion-icons</th>
<th>M5: S5 with Motion-icons</th>
<th>M6: S6 with Motion-icons</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2: S2 with Motion-icons</td>
<td>S2 with Motion-icons</td>
<td>S3 with Motion-icons</td>
<td>S4 with Motion-icons</td>
<td>S5 with Motion-icons</td>
<td>S6 with Motion-icons</td>
</tr>
</tbody>
</table>
From Table 4, it can be seen that the effects of motion-icons is not consistent. M2 and M3 experience a drop in precision performance as compared to the use of static icon while M4, M5 and M6 observe a rise. One of the main reasons for the degradation in performance for M2 and M3 is the slower display speed of using motion icons, which causes a drop in the average total number of judged shots per query (reduce by about 25%). This effect is especially prominent in M2 where we observe a performance drop of more than -9%. However, M4–M6 on the hand benefited from the motion-icons. This is because of the better assessment quality brought about by the use of motion-icons. Unlike M2 and M3, M4–M6 utilize more information from feedbacks to deliver most probable shots to the user assessment. In this scenario, the iterative process of the interactive retrieval improves the probability of the returned shots being correct. In particular, we see that our M6 which uses motion-icon and the multiple feedback strategies outperforms its static icon counter part (S6 0.328) by over 4.6%.

To further understand which queries perform better when motion-icons are used, we further split them into the four subsets that are discussed in Section 6.1. The respective results are shown in Table 5. It can be seen that the motion-icons works best for queries that are action-related which validates our earlier hypothesis.

To leverage on this property, we create the last set of tests, the M0 runs, which further allows the expert users to switch between different kinds of assessment (static keyframe or motion-icon). The results are tabulated in Table 6. As predicted, we see that all runs observe improvement with the human annotator having the flexibility to switch between the assessment modes.

### 7. Conclusions

In this paper, we have proposed a high throughput, multi-function and effective interactive video retrieval system called VisionGo. It emphasizes on three approaches to support interactive video retrieval: (a) an intuitive UI that maximizes users’ annotation efforts; (b) multiple feedback techniques for different situations; and (c) motion-icons to enhance dynamic visual semantic understanding. Combining these three features, VisionGo achieves highly effective and efficient retrieval by jointly exploring human and computer. For future work, we will look into providing an intelligent retrieval system to cater to a large number of novice users. This can be done by learning the expert users’ retrieval behaviors so as to provide recommendations on the appropriate feedback strategy to be used, leading to a more precise and personalized retrieval for the user.

### References


### Table 4

Performance of baselines and runs based on different feedback strategies (% in bracket indicates improvement over its static icon counterpart).

<table>
<thead>
<tr>
<th></th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shot judged</td>
<td>2976</td>
<td>2721</td>
<td>2700</td>
<td>2666</td>
<td>2399</td>
</tr>
<tr>
<td>Positive shot found</td>
<td>104</td>
<td>130</td>
<td>151</td>
<td>150</td>
<td>163</td>
</tr>
<tr>
<td>MAP</td>
<td>0.198 (−9.1%)</td>
<td>0.221 (−1.0%)</td>
<td>0.298 (8%)</td>
<td>0.285 (7.5%)</td>
<td>0.343 (4.6%)</td>
</tr>
</tbody>
</table>

### Table 5

Respective results for four query classes.

<table>
<thead>
<tr>
<th></th>
<th>Scene</th>
<th>General</th>
<th>Person</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>S6 run MAP</td>
<td>0.40</td>
<td>0.15</td>
<td>0.76</td>
<td>0.25</td>
</tr>
<tr>
<td>M6 run MAP</td>
<td>0.39</td>
<td>0.14</td>
<td>0.78</td>
<td>0.35</td>
</tr>
</tbody>
</table>

### Table 6

Performance comparison of baselines and different feedback strategies.

<table>
<thead>
<tr>
<th></th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shot judged</td>
<td>3323</td>
<td>3032</td>
<td>2998</td>
<td>2878</td>
<td>2565</td>
</tr>
<tr>
<td>Positive shot found</td>
<td>129</td>
<td>137</td>
<td>152</td>
<td>154</td>
<td>165</td>
</tr>
<tr>
<td>MAP</td>
<td>0.241</td>
<td>0.252</td>
<td>0.308</td>
<td>0.305</td>
<td>0.363</td>
</tr>
</tbody>
</table>


