Efficient Search and Localization of Human Actions in Video Databases

Ling Shao, *Senior Member, IEEE,* Simon Jones, and Xuelong Li, *Fellow, IEEE*

***Abstract*—As digital video databases grow, so grows the problem of effectively navigating through them. In this paper we propose a novel content-based video retrieval approach to searching such video databases, specifically those involving human actions, incorporating spatio-temporal localization. We outline a novel, highly efficient localization model that first performs temporal localization based on histograms of evenly spaced time-slices, then spatial localization based on histograms of a 2-D spatial grid. We further argue that our retrieval model, based on the aforementioned localization, followed by relevance ranking, results in a highly discriminative system, while remaining an order of magnitude faster than the current state-of-the-art method. We also show how relevance feedback can be applied to our localization and ranking algorithms. As a result, the presented system is more directly applicable to real-world problems than any prior content-based video retrieval system.**

***Index Terms*—Human actions, relevance feedback, spatio-temporal localization, video retrieval.**

I. Introduction

WITH THE increased availability of digital video recording technology, more videos are being created than ever before, with these videos coming from diverse domains such as surveillance, amateur film-making, and home recording. These videos contribute to the growth of video media databases, such as those available online to consumers (e.g., YouTube), or CCTV footage collections. From this exponential growth rises a new problem: how can these vast collections of media be accessed in the most effective way, so that users can find what they are looking for?

Manuscript received February 6, 2013; revised May 18, 2013 and June 22, 2013; accepted July 9, 2013. Date of publication August 6, 2013; date of current version March 4, 2014. This work was supported in part by the EPSRC; in part by the University of Sheffield; in part by the National Basic Research Program of China, 973 Program, under Grant 2012CB316400; in part by the National Natural Science Foundation of China under Grants 61125106 and 61072093; and in part by the Shaanxi Key Innovation Team of Science and Technology under Grant 2012KCT-04. This paper was recommended by Associate Editor P. L. Correia.

L. Shao is with the College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China, and also with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield, S1 3JD, U.K. (e-mail: ling.shao@sheffield.ac.uk).

S. Jones is with the Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: simon.m.jones@sheffield.ac.uk).

X. Li is with the Center for Optical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, Shaanxi, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2276700

Currently, video databases such as YouTube employ a text-based search, where videos are returned based on a set of keywords provided by a user. Such a system, however, is flawed; text searches can search the textual metadata associated with a video (e.g., title, description, keyword tags), but not search the videos directly. The textual metadata is rarely an accurate representation of the video's content, for two reasons: firstly, the textual information is provided by the video's uploader, whose assessment of the video may be flawed/incomplete; secondly, the amount of information in a video cannot be represented in a few keywords without necessarily losing much potentially salient information.

In this paper, we look at an alternative approach to this problem—content-based video retrieval (CBVR), which is an extension of content-based image retrieval (CBIR) to the video domain. Given an example video—a query—of what the user is searching for, CBVR directly searches the database's contents, meaning it can potentially return far more accurate results than existing text query systems, as it avoids the above problems associated with poor quality metadata. While CBVR has been well researched through a focus on keyframes and 2-D features, only a few previous works have looked at this problem using temporal information, such as [1] and [2]. We focus on searching human actions, which allows us to utilize the vast amount of prior human action recognition research. Human action recognition, a distinct task from retrieval, focuses on using trained models of human actions for classification. Because supervised recognition algorithms require prior knowledge of all the classes that are to be classified, they are unsuitable for direct use in retrieval tasks, but many of the unsupervised action representation techniques developed for recognition are also applicable in retrieval.

The system presented here distinguishes itself from the majority of previous works by not only retrieving relevant human action videos, but also localizing the exact relevant part of these videos. Such localization is particularly useful, as in a real-world database a video might be quite long, but only a very short section of it relevant to the user's query. This paper is an extension of our paper presented in CAIP 2013 [3]. There are also a few other works that focus on this problem, which we address in Section II. Among other differences, this paper distinguishes itself in two facets.

1) We design an extremely efficient algorithm for spatio-temporally localizing human actions within a dataset using only a single query of the sought-after action—this algorithm is considerably computationally simpler than

comparable works for action retrieval with localization. It could also be extended into a hierarchical model for better-than-linear performance.

2) We consider how to use relevance feedback in the context of localization, and demonstrate its efficacy in this application, also considering the effect of imperfectly localized relevance feedback.

II. Related Work

We first consider human action representation methods. Representation techniques can be largely split into two categories: global and local representations. Global representations typically attempt to capture the full structure of an action or motion with a single vector. Some better known methods based on global features include those by Yamato *et al*. [4] who introduced hidden Markov models to action recognition, and Davis and Bobick [5], who created the popular motion energy image and motion history image. While providing good reliability on clean datasets, global representations have typically been sensitive to noise, making them insufficient for realistic datasets. Local representations, on the other hand, have shown considerable promise on noisier datasets. Laptev [6] has showed that by extracting many spatio-temporal interest points (STIPs) from an action video, and using a statistical representation to model the distribution of these points within a dimensionally-reduced feature space, it is possible to distinguish between actions in a robust manner. As local features do not rely on techniques such as body segmentation or background subtraction, they are a lot more robust to noise and variables such as rotation and viewpoint, making them more suitable for tasks in realistic settings [6], [7], though they are marginally worse in very clean datasets, such as the Weizmann [8]. As we wish to prove our system in a practical scenario, we have based our algorithm on local features.

There are two stages to local feature extraction: detection and representation. We first consider detection. The earliest detectors, such as the Harris detector [9] and SIFT [10] originated in image recognition, and work only on individual frames, discarding temporal information. These detectors were later extended to the spatio-temporal domain, such as the 3-D Harris-Laplace detector [6] and the 3-D SIFT detector [11]. Each of these functions looks for a specific structural feature in the image or video; for instance, the Harris detector finds corners, and the SIFT detector finds minima/maxima after applying a difference of Gaussian (DoG) function. The most popular spatio-temporal feature detector currently comes from [7], which uses a pair of 1-D Gabor filters in quadrature on the temporal dimension, along with spatial Gaussian smoothing. As a result, this detector finds highly discriminative STIPs at points of complex motion.

After detection is feature representation. Raw pixel values are generally ineffective for this task. Dollar *et al*. [7] presented a simple descriptor that takes the brightness gradients along all three dimensions in the STIP volume, and concatenates them. Laptev *et al*. [12] used the combined histogram of oriented gradients and histogram of optical flow (HOG-HOF) descriptors. The SIFT descriptor [10], as with its detector, has been extended to 3-D by Scovanner *et al*. [11]. Histogram of oriented gradients (HOG), which is similar to the SIFT descriptor, has also recently been extended to 3-D by Kläser *et al*. [13], who used a series of implementation innovations that made its calculations efficient enough to be practical. 3-D-HOG has proven itself to be a particularly effective descriptor, achieving state-of-the-art results compared to all other existing local descriptors.

In most works on human actions, after the action has successfully been represented, the goal is to train a classification model on the represented action, such as in recent works, [14] and [15]. Our goal, however, is to perform content-based action retrieval, where we attempt to rank videos in a dataset based on their similarity to a query video—this task has its origins in CBIR [16], [17]. Several works have considered content-based action retrieval before, such as [18] and [19], taking features typically used for classification and using them in a retrieval context. Other works have looked at video annotation [20], [21] and video summarization [22] in order to facilitate keyword-based action retrieval.

Our goal is to perform action retrieval with spatio-temporal localization. Trained recognition and localization models for human actions are relatively common in recent years, such as Yuan *et al*. [23], who devised a spatio-temporal extension of the branch-and-bound method (STBB), using local features. In [24], spatial body tracking and a temporal sliding window are used to perform action localization in the noisy Hollywood localization dataset. Local features have also been used for localization. In [25], local features are used to perform localization in several datasets, by attaching a relevancy weight to each local feature (which measures how relevant that local feature is to the classified action) and then creating a spatio-temporal bounding box around all the local features which pass a relevancy threshold. Ryoo and Aggarwal [26] demonstrated the effectiveness of local feature voting in their activity recognition system—where each feature casts a vote for the spatio-temporal boundaries of the action. Oikonomopoulos *et al*. [27] also used feature voting in combination with mean shift. Kliper-Gross *et al*. [28] performed action classification in unconstrained videos but ignore the localization task, instead assuming that each unconstrained video has exactly one action somewhere within it. Liu *et al*. [29] performed video copy location detection, where the goal is to find copies of a clip within a video repository with certain transformations applied to it, such as blurring.

Action retrieval and spatio-temporal localization—sometimes referred to less precisely as an spatio-temporal action search—is an uncommon common task, though it has been attempted by several works, which we address here [2], [29]–[32]. In [31], the authors perform spatial localization by person tracking, and temporal localization using shot boundaries. This method of temporal localization assumes that each shot will only contain one action; however, Ning *et al*. [32] use biological features to represent motion videos, and perform a hierarchical 3-D sliding window search to find action candidates. This, however, is only demonstrated on simple, single-subject works, and 3-D sliding windows are costly to implement in practice. On the other hand, Ke *et al*. [30] represent a scene using oversegmented spatio-temporal

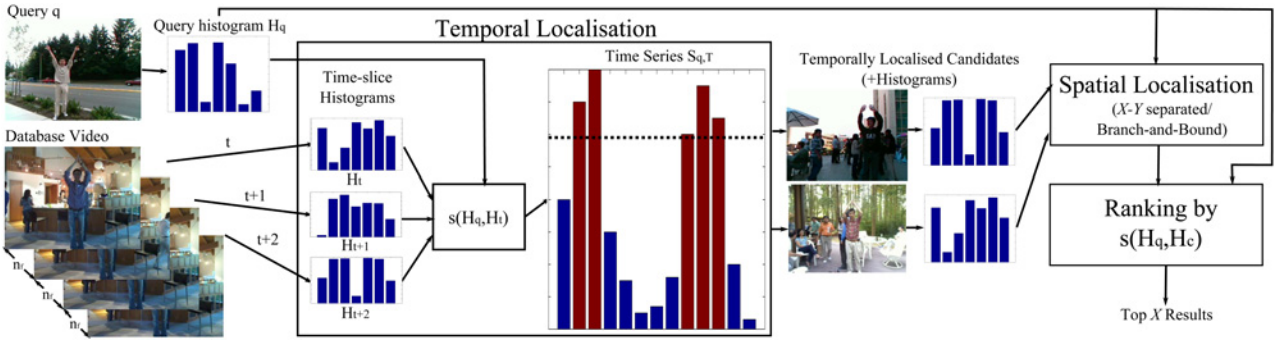


Fig. 1. Overview of the spatio-temporal localization and ranking aspects of our algorithm. Note that the spatial localization part of our algorithm has been simplified here for convenience.

volumes to find actions in crowded videos. Their work, however, requires interactive use of a graph-cut tool to generate an appropriate query, and is highly scale-dependent both spatially and temporally. In their work, they only consider a single scale. Extending this paper to find actions at multiple 3-D scales would be computationally costly. Finally, and most similarly to our own work, Yu *et al*. [2] use Random Forests to find features to represent the action videos, and then perform STBB to find ideal action candidates within the action videos. Their method, however, has a very computationally expensive query algorithm—an hour of video will take over 20 seconds to search through. After offline indexing, our method takes less than a second to perform the same query.

Even without localization, the challenge in content-based retrieval is extracting sufficient salient information about the action from the single query to perform accurate ranking; a single sample neither contains information about intraclass variation, nor provides enough data to easily distinguish between genuine action features and noise. To deal with this problem, researchers have introduced relevance feedback (RF) [33]–[35], where the retrieval system uses user feedback about its results to iteratively improve the query. While RF has to date been primarily applied to image retrieval [34]–[39], the concept has been proven in action retrieval as well [1], [18], [19], [40]. In this paper, we look at using relevance feedback to improve both the retrieval and localization results.

III. Methodology

A. System Overview

We define the goal of our system as follows: given a query video containing a prelocalized human action, the system searches a video database for all instances of this human action. It spatio-temporally localizes and ranks these actions according to relevance, before returning them to the user. At this point, the user may mark results for relevance feedback and run the query again iteratively, until he/she is satisfied with the results.

We focus on human actions for two reasons. Firstly, videos of humans constitute the majority of existing video media, and are therefore highly likely to be the target of a user's query. Secondly, the majority of existing video datasets for recognition and retrieval research are also focused on human actions.

Fig. 1 gives an overview of the system we have designed. The database of videos is preprocessed in batch—local features are extracted and clustered into codewords. When a user provides a query video, several steps are performed in sequence. First, the same feature extraction is applied to this video. Using these features, the system performs temporal localization to find a large number of candidate results in the database. We refine the candidates by performing a further round of localization—this time spatially—and then rank the candidates according to a bag-of-words model. The top X results are returned to the user. At this point, the user can choose to provide relevance feedback if necessary, to improve results. Several of the ranked videos are marked as relevant/irrelevant to the query, and this relevance feedback is used in a further search, to improve both the localization and ranking steps.

Below, we discuss in detail how our system operates. We have treated efficiency as the utmost priority, while attempting to maintain practical accuracy; therefore we justify the design in these terms.

B. Video Representation

As described above, the video database must be preprocessed with feature extraction before a search can be performed. The system achieves this using an existing feature detector and descriptor (e.g., Dollar's [7], SIFT [10] and HOG3-D [13]). We extract features at a roughly consistent rate with respect to time. Having extracted features in such a manner across the whole dataset, we then apply a combination of PCA (capturing 95% variance) and k-means clustering to the descriptors, retrieving k visual codewords. The choice of k is important, as generally higher k will give better accuracy, but will also slow down retrieval, and too high k will lead to sparsity issues in the localization algorithm.

Then, each video can be efficiently represented by the set of its features. Each feature can be represented as a tuple, $t = (x, y, t, c)$, where x, y and t represent the spatio-temporal location of the feature within its video, and c is its codeword. The process of feature extraction can often be quite costly—however, in a retrieval model this impact is minimized, as feature extraction is performed just once on the database—for subsequent searches, feature extraction is only performed on the query video.

C. Localization

We separate temporal and spatial localization into linear time algorithms to decrease the search time of our algorithms. Simultaneous spatio-temporal localization using branch and bound, such as in [2], has very high computational complexity, especially for lengthy video sequences. We believe such localization is unnecessarily complex—we argue that, using local features, temporal localization can be performed accurately, independent of spatial localization. It can be observed that, in the majority of video sequences, a simple human action will occupy only a small proportion of the temporal domain, but a relatively large proportion of the spatial domain. Therefore, to perform an efficient search, we first perform temporal localization to identify a relatively small number of candidate regions with respect to the size of the dataset, before performing a more complex spatial localization operation on only these candidate regions.

1) *Temporal Localization:* We first perform an additional step of preprocessing on the database in order to facilitate fast temporal localization. We divide the temporal space into slices of f frames, and for each time-slice, we generate a normalized bag-of-words histogram, $H_t \in H_T$. The appropriate choice of f is made empirically, and this choice is important—it should be small enough to account for short actions (approximately half the length of the shortest action in the database), but large enough so that the histograms are not overly sparse. The choice of k and f also presents a trade-off between the time efficiency and accuracy of temporal localization during a search. In our experiments, we set f to 8, which is approximately half the length of the shortest action in our test datasets.

During a search, features are extracted from the query video and the query is represented by a single normalised bag-of-words histogram, H_q. It is important at this point to note that H_q and H_t are not directly equivalent; H_q represents the complete action, whereas each H_t will at most represent a temporal fraction of the overall action. By using the correct comparison metric, however, it is still possible to compare H_t and H_q to generate a useful value. For this we can use the histogram intersection

$$s(H_q, H_t) = \sum_{i=1}^{k} min(H_q^i, H_t^i). \tag{1}$$

For retrieval tasks, however, a modification of the histogram intersection might be more appropriate

$$s(H_q, H_t) = \sum_{i=1}^{k} \frac{min(H_q^i, H_t^i)}{H_n^i} \tag{2}$$

where H_n is the normalized histogram of the features across the entire database. Using H_n in this way, rarer features in the dataset—those with a presumably higher information value—have a greater weight. A major assumption of our model here is that a high value for $s(H_q, H_t)$ is predictive that time-slice t contains part of action q. This is subtly different from the standard procedure, where two full-action histograms are compared. We have experimentally determined that our comparison is indeed suitable for this purpose.

Having calculated $s(H_q, H_t)$ for all $t \in T$, the system then looks for temporally adjacent regions with high values for s, which indicate potential candidate regions within each database video. A threshold value is determined individually for each scene[1] in the database, above which a time-slice is considered to be a match for the query. This threshold is set as a standard deviation above 0, and from this we generate a set of candidate regions.

In order to reduce the noisiness of these regions, we then perform two additional operations. Firstly, we join adjacent regions: if time-slice t is a match, and $t + 2$ is a match, then t+1 will also be considered a match. Then we remove singleton regions: if t and $t + 2$ do not match the query, then $t + 1$ is not considered a match either. Similar in concept to region growing and shrinking in 2-D image segmentation, these operations significantly reduce the effect of noise (e.g., partial occlusions) but are relatively cheap to perform. When all these operations have been performed, the final set of candidates, termed C, is passed through to the spatial localization step.

The complexity of our temporal search over a single video is $O(k\frac{n}{f})$ where n is the number of frames in the video, k is the number of codewords, and f is the size of the time-slice. We postulate that the efficiency of the presented technique may be improved to a logarithmic time function by performing a coarse-to-fine hierarchical search on large-to-small time-slices, but such considerations are beyond the scope of this paper.

2) *Spatial Localization (SL):* Once temporal localization has identified a set of candidates, we linearly apply SL to the temporal candidates—at this point our candidate set has already been pruned to a comparatively small subset of the total database, so spatial localization need not be as computationally complex as temporal localization. If maximum efficiency is desired, however, spatial localization can be performed after ranking, only on the top X results—this results in a SL step which is constant with respect to the size of the database. For reasons of storage complexity, we only perform minimal preprocessing of the database to prepare for SL. The feature tuples retrieved in the feature extraction preprocessing step are stored in temporal order, so that once the temporal bounds of each candidate are known, the features belonging to each candidate can be swiftly retrieved.

In our experiments we investigate two methods of spatial localization, to evaluate their accuracy and performance trade-offs. We also look at the effects of performing spatial localization before, and after the ranking step. The former is expected to result in greater accuracy, as ranking can be more precise; however, if spatial localization is performed after ranking, spatial localization only needs to be performed on the top X results that are to be returned to the user. This may result in a considerable efficiency improvement.

a) *$X - Y$ Separated SL:* Our first approach linearly separates localization along both spatial dimensions X and Y to minimize computational complexity, and to mitigate local feature sparsity in two dimensions. In the absence of simultaneous identical actions, we hypothesize that this technique will achieve reasonable accuracy. We split the temporally localized

[1]A scene refers here to a single contiguous camera shot

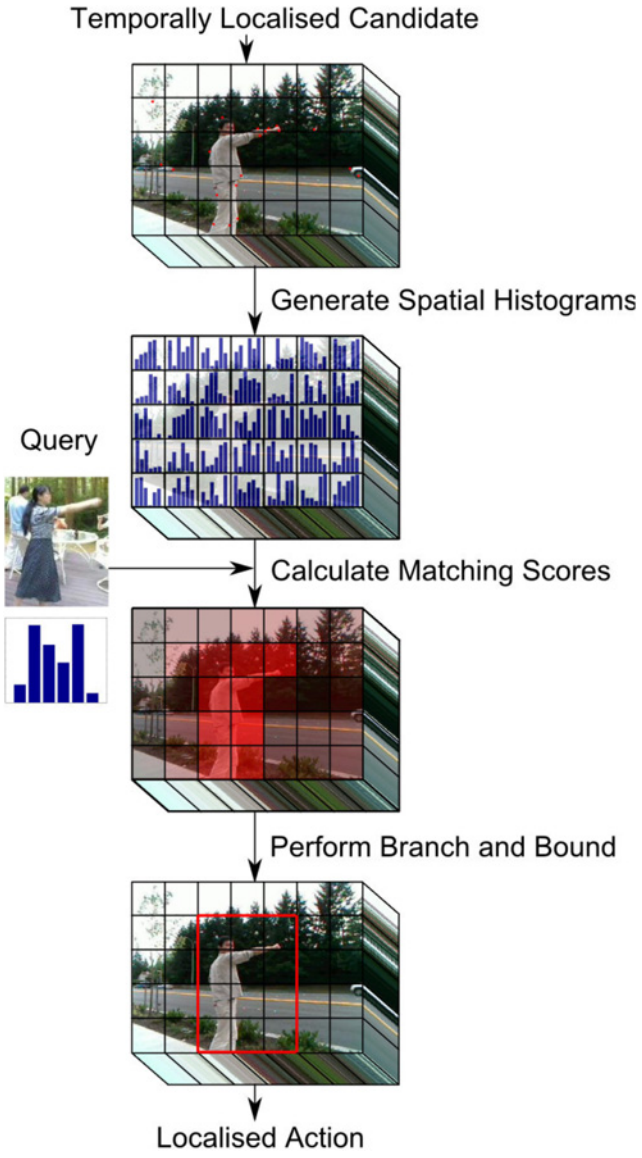


Fig. 2. Overview of how the branch and bound algorithm is applied for spatial localization in our work.

block into equal slices along each dimension in turn, and generate a histogram H_s for each slice, similar to the procedure used in temporal localization. Then, we use (2) to generate a set of scores r over that dimension.

At this point, the method diverges from temporal localization, as in spatial localization we are only looking for a single region, whereas in temporal localization we attempt to find multiple instances of the action. To find the optimal subregion, we first subtract a threshold $d = l \cdot max(r)$ from each of the scores in r, with l set to 0.25. Then, we perform a maximal sub-array search on r, using Kadane's algorithm [41] to perform this in $O(|r|)$.

Having determined the extent of the action along each dimension separately, we simply combine this information to determine the final bounding box.

b) *Branch-and-Bound Simultaneous SL:* For potentially greater accuracy, but at the cost of higher computational complexity, it is possible to perform localization along both spatial dimensions simultaneously. Here, the spatial extent of the candidate is divided into a number of equally sized 2-D windows. Using the features, ad hoc histograms for each of these windows are generated for each temporal candidate. Using (2), the system establishes for each 2-D window whether it matches the query action; the result of this entire operation is a single low-dimensional relevance image, r.

Using r_i, we perform a 2-D branch-and-bound operation, based on the 2-D object localization algorithm described in [42], to find the optimal subwindow containing the action. Branch-and-bound, similar to the sliding window approach, is guaranteed to converge to the optimal subwindow, but its average running time is $O(xy)$ rather than $O((xy)^2)$. Similar to $X-Y$ separated localization, we need to choose a decision threshold d to subtract from r—branch-and-bound assumes the relevance decision threshold of each window is at 0, but r consists only of positive values. We pass $r-d$ into the branch-and-bound algorithm, and we set the upper bound function to the following:

$$\hat{f}(Y) \equiv f^{+}(y_{\cup}) + f^{-}(y_{\cap}) \tag{3}$$

where $y_{\cup}$ and $y_{\cap}$ are the maximally and minimally sized rectangles within candidate set Y, respectively; $f^{+}(y)$ and $f^{-}(y)$ are the sum of all positive points and sum of all negative points in rectangle y respectively. This is the same upper bound described in the linear classifiers section of [42]. As this paper details, we can use positive and negative integral images to calculate (3) in constant time.

A diagram of our branch-and-bound spatial localization method is shown in Fig. 2.

D. Ranking

Having performed localization, we have a set of candidates c_i and their bounding boxes B_{c_i}. A single feature histogram H_{c_i} is generated for each c_i, collating the features that fall within B_{c_i}; H_c for all $c_i \in C$ can then be matched against H_q using (2). The set of scores generated by this operation provide a simple basis for ranking the localized candidates; those with higher scores are ranked first. The top X ranked candidates are returned to the user as results. The generation of H_{c_i} can be made computationally simpler using integral histograms, as seen in [32], though this speed-up is relatively insignificant compared to the extra storage required.

In previous systems, ranking and localization occur simultaneously, such as in top X branch-and-bound localization. We have set apart our system. Because of the inclusion of this discriminative ranking step, the localization can be extremely permissive, or, in other words, insensitive to false positives. The localization generates a large number of candidates, not all of which will be matches for the query, with the expectation that most false candidates will be pruned at the ranking stage. Furthermore, good ranking should generally favor better-localized candidates, meaning that to an extent ranking can compensate for a weaker localization step. This has allowed us to keep the localization step computationally very simple, while still returning strong results to the user.

E. Relevance Feedback

While the process described above delivers useful results, they can be further enhanced through relevance feedback. If a user's initial search has completed, but that user is still unsatisfied with the results of the query, he/she can provide feedback about the relevance of each result to his/her search. Using a model to integrate this additional information with the original query, a more discriminative second search can be performed—this model is usually an online learning technique such as an SVM or AdaBoost.

As the goal of our system is efficiency, we have evaluated various relevance feedback methods through their relative time-cost effectiveness, and settled on two effective techniques. For determining the relevance of time-slices during temporal localization, we apply an SVM with a histogram intersection kernel (which satisfies Mercer's condition). To update the ranking of candidates, we use simple query expansion from positive feedback only; this method is described further in [40].

We also considered applying SVM relevance feedback to the spatial localization step of our algorithm; however, due to complexity-constraints and minimal performance improvements in our preliminary experiments, we do not report on this in the results section below.

Finally, as our system performs localization, we have to make a novel consideration related to relevance feedback. In prior retrieval systems with RF, it is straightforward for a user to mark the results for feedback, as each returned document will have a binary relevance value—in other words, it is either relevant or not. However, with localization many of the returned results will be partially relevant, because many of the ostensibly relevant results will be imperfectly localized. To deal with this, we first consider that a user will only view a result as relevant if its bounding box overlaps sufficiently with the desired action. Once a user has decided that a result is relevant, he/she can then return it as feedback in one of two ways.

1) Adjusted: the result is modified by the user to overlap perfectly with the action, and is then returned to the system as feedback.
2) Unchanged: the result is returned as feedback with no modification to the bounding box.

If a user returns adjusted feedback, results on the next iteration should be more accurate than if he/she provides unchanged feedback. However, correcting the bounding box for adjusted feedback places a considerable onus on the user, relative to simply marking results as relevant or not. In our experiments below, we consider both methods of returning feedback to evaluate which is more practical.

IV. Experiments

In this section we discuss the experiments we performed on two datasets to validate our ideas. We detail the exact experimental setup, and give an evaluation of the results obtained from these experiments.

A. Datasets

We used the MSR2 [23] and UT-interaction [43] datasets to show our results. These datasets are specifically designed for spatio-temporal localization experiments, which make them well-suited for our purposes. Each dataset is composed of multiple videos of around 1 minute long each, and in each video multiple actions/interactions take place—these actions mostly occur temporally sequentially, though a certain few actions/interactions do occur in parallel.

The MSR2 dataset has 54 videos, comprising a total of 46 minutes of raw footage, shot in various locations on a university campus. In many of the videos, there is a lot of background motion unrelated to the action, generated by passers-by, making this dataset a challenge for traditional methods based on background subtraction. There are three classes of cyclical action—handwaving, handclapping, and boxing—performed by a variety of actors, with a total of 203 actions overall. Examples of these actions can be seen in Fig. 3. All of the actions are shot from the same orthogonal perspective, and actions within each class are performed in roughly the same way (only small variations in execution). The distance from the camera, the length of the action, and the actor performing the action are all varied.

The UT-interaction dataset, alternatively, is divided into only 20 videos with a total of 23 minutes of video, shot in two scenes from a single, aerial viewpoint. Rather than simple human actions, this dataset contains six classes of human-human interactions, namely: hand shaking, hugging, kicking, pointing, punching and pushing, which can be seen in Fig. 4. There are 120 interactions in total, performed by varying combinations of actors. The difficulty of localising within the dataset is increased, as the actors engage in several unlabeled actions which look similar to the labeled actions. Unrelated persons also walk into and out of the shot during the video sequences.

B. Setup

We first prepared each dataset for the retrieval experiments. The datasets were scaled uniformly to 240 pixels in height (maintaining aspect ratio) and 15 frames per second, so the feature extraction procedure was identical for both. We extracted features from each dataset at an average rate of 180 features per second, detecting features with multiscale Dollar [7] and describing them with HOG3-D [13]. The resulting features were clustered into 1000 codewords after PCA was performed to capture 95% of the features' variance. Time-slice histograms were generated over the whole dataset in batch before the main retrieval experiments; as these preprocessing steps can be performed before a retrieval search is performed, they are not included in the performance statistics. Each time slice was 10 frames in length, and the 2-D spatial grid was divided into 10 by 10 pixel blocks. These parameters were chosen based on observations of the minimum length and size of the actions within the dataset.

We performed leave-one-out cross validation retrieval experiments on each dataset in order to provide the most reliable results. We treated each action $a_i \in D$ as the query in turn, where D is the entire dataset. The search for action a_i was performed on a subset of D, $D - v_i$, where v_i is the discrete video sequence from which a_i was extracted. $D - v_i$ is used rather than $D - a_i$, as results may be skewed in favor of other

Fig. 3. Actions of the MSR2 [23] dataset.

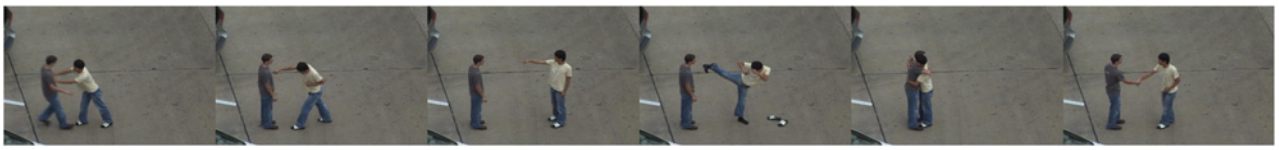


Fig. 4. Actions of the UT-interaction [43] dataset.

actions in the same video sequence. We averaged the results over each individual query to get the final results.

Relevance feedback, rather than being given by a real user, was simulated for experimental consistency and convenience. We assessed whether a result would be deemed as relevant by our virtual user using the following metric:

$$L(E, G) = \frac{volume(E \cap G)}{volume(E \cup G)} \tag{4}$$

where E is the spatio-temporal bounding box of the estimated action, and G is the bounding box of the closest relevant action (taken from the ground truth). A result was deemed to be relevant when $L(E, G) > thres$. $thres$ was chosen to allow for an average overlap of 0.5 per localized dimension—0.5 for temporal-only localization, and 0.5^3 for spatio-temporal localization—following the example set by previous works such as [2].

Typically, the performance of a retrieval algorithm can be assessed through precision/recall and top X results. However, the formulation of these metrics assume that each result has a binary relevance to the query. In our model, an imperfectly localized result may have partial relevance to the query, as measured by its overlap with a relevant action. We used (4) to calculate relevance in our results.

During relevance feedback, a maximum of five positive and five negative feedback samples were given at each iteration. We ran five iterations of relevance feedback, after which we saw no further significant improvement in any of the experiments.

C. Results

Below we show our results and describe their implications.

It is clear that there is a significant difference between performance on the UT and MSR2 datasets. We ascribe the significantly poorer performance on the UT dataset primarily to: a greater number of action classes, increasing the chance of false positives and fewer examples per action class, resulting in a lower percentage of true positives. It is worth noting that relevance feedback makes a considerable impact on the performance here too, suggesting that there may be greater intraclass variability in the UT dataset compared to the MSR2 dataset.

We can see the contribution of various methods of relevance feedback in Figs. 5(a) and (b). On the MSR2 dataset, by

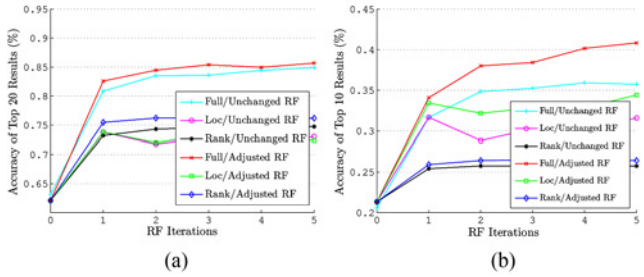


Fig. 5. Comparison of the contribution of various relevance feedback methods. (a) MSR2. (b) UT.

the fifth iteration, it matters little to the results whether a user returns adjusted or unchanged feedback—however on the UT dataset, adjusted feedback is considerably better than unchanged feedback. This would confirm the natural intuition that more difficult search queries require higher quality feedback samples. The results when feedback is applied to only: 1) the temporal localization step, and 2) the ranking step of our algorithm are also shown in these two figures. Feedback improves the accuracy of the localization step more, but both steps work synergistically to achieve the highest performance.

Fig. 6 shows the performance of our modified histogram intersection against the ordinary histogram intersection implemented in the kernel SVM, on the MSR2 dataset. After relevance feedback, our modified intersection performs consistently better by around 1%.

Fig. 7(a) and (b) gives precision/recall curves for various levels of relevance feedback. For both datasets and all iterations, at low recall, precision is relatively high, but at higher levels of recall—beyond 10%—performance rapidly tails off. This indicates the difficulty of learning a human action from a single example. However, relevance feedback considerably improves the situation, and the tables furthermore show that only one or two iterations of relevance feedback are required to reach optimal performance. Improvements after this are negligible.

Fig. 8(a) and (b) shows the effect on accuracy of using branch-and-bound localization, $X - Y$ separated localization, and temporal-only localization—for the spatial localization methods, this is also broken down by whether spatial localization was performed before or after ranking. Several results are clear from this. Firstly, spatial localization appears to be

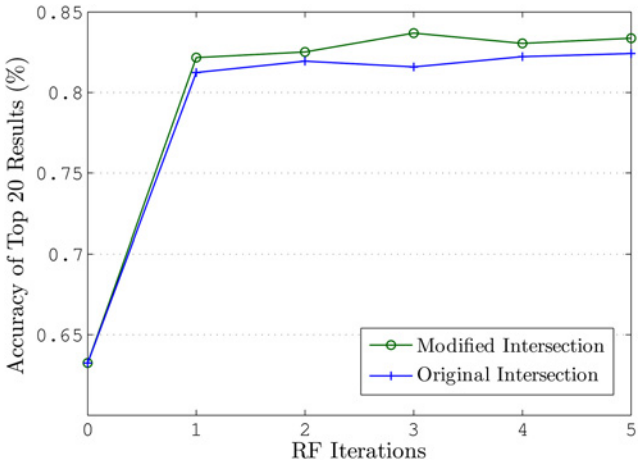


Fig. 6. MSR2. Modified histogram intersection against the original.

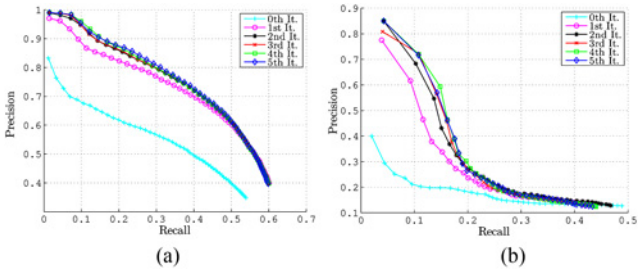


Fig. 7. Precision/recall after different levels of relevance feedback. (a) MSR2. (b) UT.

a relatively trivial task—there is no significant difference in accuracy between the temporal-only and spatial localization methods. Secondly, branch-and-bound performs considerably better than $X-Y$ separated localization, as expected. Finally, performing spatial localization after ranking has a small but significant impact on accuracy. Note that for the MSR2 dataset we look at the top 20 results, whereas for the UT dataset we only look at the top ten results—retrieval results can be distorted by size of the dataset and the number of action classes, so we partially compensated for this.

In Tables I and II we can see the running times of an individual query, both before and after relevance feedback, for various methods of localization. These are an order of magnitude better than the next best attempt to perform a search on the MSR2 dataset [2], which takes 26.7 seconds for a single query. This highlights the advantages of separating ranking, temporal localization and spatial localization into discrete steps. We compared the branch-and-bound spatial localization method to $X-Y$ separated localization, showing that the former, as expected, is much slower than the latter. This impact on runtime can be mitigated, however, by performing spatial localization only after the ranking step, on the top X results—in Tables I and II run-times are shown both for spatial localization performed before and after ranking, denoted in the Loc. Order column.

V. Discussion

Efficient content-based search systems, such as the model presented here, are becoming increasingly relevant in today's

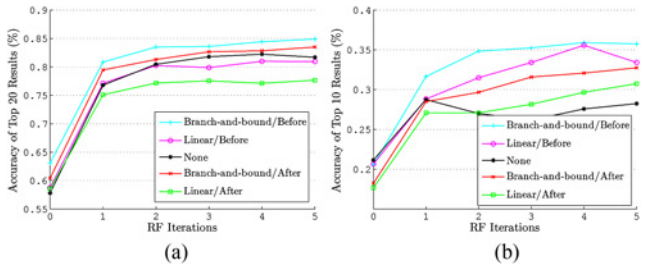


Fig. 8. Effect on the accuracy of various spatial localization methods, as well as temporal localization alone. (a) MSR2. (b) UT.

TABLE I
MSR2 Query Time Costs

Loc. Met.	Loc Order	Time (s) (1st it)	Time (s) (RF)
Temp. Only	N/A	0.1774	1.283
Linear	Before	0.502	1.392
Linear	After	0.223	1.363
B & B	Before	0.725	1.446
B & B	After	0.247	1.304

TABLE II
UT Query Time Costs

Loc. Met.	Loc. Order	Time (s) (1st it)	Time (s) (RF)
Temp. Only	N/A	0.099	0.635
Linear	Before	0.281	0.810
Linear	After	0.195	0.750
B & B	Before	0.906	1.303
B & B	After	0.204	0.693

world, as sophisticated searches are increasingly necessary to navigate the huge amounts of data. Through theoretical discussion and experimental results, we have demonstrated basic practical applicability of our system to this task of real-world video search. In designing our algorithm, we have taken an efficiency-first approach—this has resulted in the creation of a fast permissive temporal-then-spatial localization technique, followed by a more orthodox histogram ranking step, both of which can be assisted by relevance feedback.

Despite the moderate success of the model shown here, our work is only an initial example of how content-based searches can be tackled from an efficiency perspective. We believe that general principles of our system, such as batch preprocessing, spatio-temporal feature binning, and dimensionally-sequential localization can be combined with a wide variety of existing human action recognition/localization techniques, and that concentrated efforts in investigating these techniques may yield further improved performance. Additionally, while our algorithm is fast enough for use on databases of even thousands of hours in length, it would not work well with online databases of millions or billions of hours—future research should concentrate on how to represent visual features so that videos of length t can be searched in better than linear time. One potential method for this includes extending our work into a temporally hierarchical model, using decreasing values of f to perform a coarse-to-fine search through the dataset, and performing indexing on the histograms at the coarser levels, potentially giving a logarithmic time complexity. Our future work will consider this problem.

REFERENCES

[1] S. Jones, L. Shao, J. Zhang, and Y. Liu, "Relevance feedback for real-world human action retrieval," *Pattern Recognit. Lett.*, vol. 33, no. 4, pp. 446–452, Mar. 2012.

[2] G. Yu, J. Yuan, and Z. Liu, "Unsupervised random forest indexing for fast action search," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 865–872.

[3] S. Jones and L. Shao, "Rapid localization and retrieval of human actions with relevance feedback," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2013, pp. 20–27.

[4] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1992, pp. 379–385.

[5] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1997, p. 928.

[6] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, nos. 2–3, pp. 107–123, 2005.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Workshop Visual Surveill. Perform. Evaluat. Track. Surveill.*, 2005, pp. 65–72.

[8] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[9] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. Eur. Conf. Comput. Vision*, 2002, pp. 128–142.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. ACM Multimedia*, 2007, pp. 357–360.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2008, pp. 1–8.

[13] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3-D-gradients," in *Proc. Brit. Mach. Vision Conf.*, 2008, pp. 995–1004.

[14] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 288–303, Feb. 2010.

[15] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. ICCV*, 2011, pp. 1419–1426.

[16] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu, "Content-based browsing of video sequences," in *Proc. ACM Multimedia*, 1994, pp. 97–103.

[17] H. J. Zhang, J. Wu, D. Zhong, and S. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, no. 4, pp. 643–658, 1997.

[18] R. Yan, A. Hauptmann, and R. Jin, "Negative pseudo-relevance feedback in content-based video retrieval," in *Proc. ACM Multimedia*, 2003, pp. 343–346.

[19] R. Jin and L. Shao, "Retrieving human actions using spatio-temporal features and relevance feedback," in *Multimedia Interaction and Intelligent User Interfaces*, L. Shao, C. Shan, J. Luo, and M. Etoh, Eds. Berlin, Germany: Springer-Verlag, 2010.

[20] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song, "Unified video annotation via multigraph learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 733–746, May 2009.

[21] M. Wang, X.-S. Hua, J. Tang, and R. Hong, "Beyond distance measurement: Constructing neighborhood similarity for video annotation," *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 465–476, Apr. 2009.

[22] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, and T.-S. Chua, "Event driven web video summarization by tag localization and key-shot identification," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 975–985, Aug. 2012.

[23] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, Sep. 2011.

[24] A. Kläser, M. Marszalek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *Proc. Int. Workshop Sign, Gesture, Activity*, 2010.

[25] T. H. Thi, J. Zhang, L. Cheng, L. Wang, and S. Satoh, "Human action recognition and localization in video using structured learning of local space-time features," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, 2010, pp. 204–211.

[26] M. Ryoo and J. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 1593–1600.

[27] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal localization and categorization of human actions in unsegmented image sequences," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1126–1140, Apr. 2011.

[28] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 256–269.

[29] B. Liu, Z. Li, Y. Yang, M. Wang, and X. Tian, "Real-time video copy-location detection in large-scale repositories," in *Proc. ACM Multimedia*, vol. 18, no. 3. 2011, pp. 22–31.

[30] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Proc. IEEE Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[31] R. Ji, H. Yao, and X. Sun, "Actor-independent action search using spatiotemporal vocabulary with appearance hashing," *Pattern Recognit.*, vol. 44, no. 3, pp. 624–638, Mar. 2011.

[32] H. Ning, T. X. Han, D. B. Walther, M. Liu, and T. S. Huang, "Hierarchical space-time model enabling efficient search for human actions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 808–820, Jun. 2009.

[33] G. Salton, *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1971.

[34] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proc. ACM Multimedia*, 2001, pp. 107–118.

[35] P. Hong, Q. Tian, and T. S. Huang, "Incorporate support vector machines to content-based image retrieval with relevance feedback," in *Proc. IEEE Int. Conf. Image Process.*, 2000, pp. 750–753.

[36] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[37] W. Bian and D. Tao, "Biased discriminant Euclidean embedding for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 545–554, Feb. 2010.

[38] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.

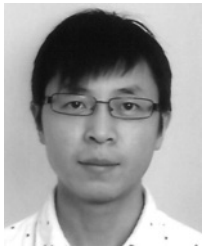
[39] D. Tao, X. Li, and S. J. Maybank, "Negative samples analysis in relevance feedback," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 568–580, Apr. 2007.

[40] S. Jones and L. Shao, "Content-based retrieval of human actions from realistic video databases," *Inform. Sci.*, vol. 236, pp. 56–65, Jul. 2013.

[41] J. Bentley, "Programming pearls: Algorithm design techniques," *Commun. ACM*, vol. 27, no. 9, pp. 865–873, Sep. 1984.

[42] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2129–2142, Dec. 2009.

[43] M. S. Ryoo and J. K. Aggarwal. "Ut-interaction dataset, ICPR contest on semantic description of human activities (SDHA)" [Online]. Available: http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html



Ling Shao (M'09–SM'10) is a Senior Lecturer with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K., and a Guest Professor with College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China. Prior to this, he was a Senior Scientist with Philips Research, The Netherlands. His research interests include computer vision, pattern recognition, and video processing.

Dr. Shao is an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS and several other journals. He is also a fellow of the British Computer Society.

Simon Jones received the B.Eng. degree in software engineering and artificial intelligence from the School of Informatics at Edinburgh University, Edinburgh, U.K., in 2008. He is currently pursuing the Ph.D. degree from the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K.

His research interests include multimedia retrieval, human motion clustering, and activity analysis.

Xuelong Li (M'02–SM'07–F'12) is a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Chinese Academy of Sciences, Shaanxi, China.