Rapid Localisation and Retrieval of Human Actions with Relevance Feedback

Simon Jones and Ling Shao

The University of Sheffield
{simon.m.jones,ling.shao}@shef.ac.uk

Abstract. As increasing levels of multimedia data online require more sophisticated methods to organise this data, we present a practical system for performing rapid localisation and retrieval of human actions from large video databases. We first temporally segment the database and calculate a histogram-match score for each segment against the query. High-scoring, adjacent segments are joined into candidate localised regions using a noise-robust localisation algorithm, and each candidate region is then ranked against the query. Experiments show that this method surpasses the efficiency of previous attempts to perform similar action searches with localisation. We demonstrate how results can be enhanced using relevance feedback, considering how relevance feedback can be effectively applied in the context of localisation.

1 Introduction

In recent years search engines – such as Google – that operate on textual information have become both mature and commonplace. Efficient and accurate search of multimedia data, however, is still an open research question, and this is becoming an increasingly relevant problem with the growth in use of Internet multimedia data. In order to perform searches on multimedia databases, current technology relies on textual metadata associated with each video, such as keyword tags or the video's description – unfortunately such metadata are often incomplete or inaccurate. Furthermore, even if a textual search engine can locate the correct video, it cannot search within that video to localise specific sub-sequences that the user is interested in.

Compared to this, content-based retrieval systems present a better alternative. Such systems directly search through the content of multimedia objects, avoiding the problems associated with metadata searches. Content-Based Image Retrieval (CBIR) is the primary focus of many researchers. Video retrieval (CBVR) has also been studied [1], but to a far lesser degree. Retrieval of human actions in particular has received relatively little attention in comparison to action recognition, with some notable exceptions in [2,3]. This is perhaps because human actions are particularly difficult to retrieve because only a single query example is provided to search on, but this single query cannot capture the vast intraclass variability of even the simplest of human actions. Additionally, if the

R. Wilson et al. (Eds.): CAIP 2013, Part I, LNCS 8047, pp. 20–27, 2013.
© Springer-Verlag Berlin Heidelberg 2013

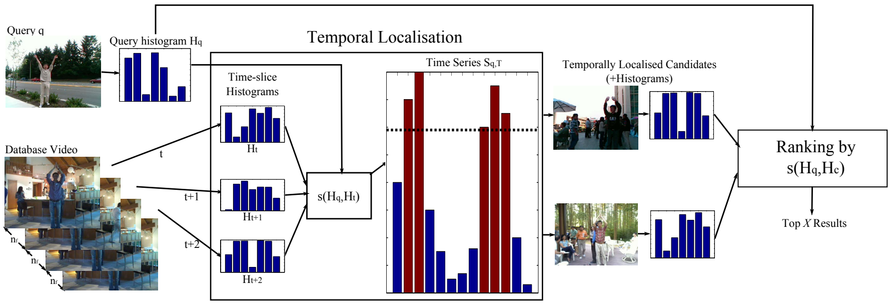


Fig. 1. An overview of the localisation and ranking aspects of our algorithm. Relevance feedback has been omitted for clarity.

query itself is noisy it can be difficult to isolate the relevant features of the action. One method researchers use to overcome this issue is relevance feedback, such as presented in [2].

Finding relevant videos alone is not enough for a practical video retrieval system. It is also necessary to *localise* the relevant segments within longer videos, as in the real world actions of interest are rarely neatly segmented. In the image domain, Rahmani et al. [4] and Zhang et al. [5] have combined retrieval with spatial localisation of objects. In videos, most localisation to date has been performed in a recognition context, such as in [6]. However, more recently Yu et al. [3] have performed human action retrieval combined with localisation.

Our goal is to introduce a time-efficient system for performing human action retrieval, showing how localisation and retrieval can be integrated while maintaining accuracy. We argue that, compared to previous works such as Yu et al.[3] our method is an order of magnitude more efficient in time and space, making it far more practical for real-world searches, while still maintaining practical accuracy. Furthermore, we experiment with the addition of relevance feedback in various forms, demonstrating that even imperfectly localised feedback can be used to significantly improve results. We believe ours is the first work to consider the effect of noisy relevance feedback samples in our experimentation, detailed further in section 3.

2 Localisation and Retrieval

Our foremost consideration in performing video localisation and retrieval is efficiency, as videos are data-intensive and yet searches need to be fast to be practical. In this section, we detail a localisation algorithm. This algorithm has linear complexity with respect to the size of the database, the potential to be further optimised, yet makes little sacrifice in accuracy. We additionally reduce the query time through batch pre-processing of the database to a compact representation. As it is based on local features, our algorithm is scale-invariant, robust against noise and partially viewpoint invariant.

2.1 Pre-processing

In the pre-processing stage it is helpful to consider previous work on human action recognition. Approaches to human action recognition are broken down into two categories based on the feature extraction method: global feature-based methods and local feature-based methods [7]. Global feature based methods, such as [8], consider the whole human shape or scene through time. Local feature-based methods, such as [9,10], discard more potentially salient information, such as the structural information between features, so are generally not as accurate on clean datasets. However, they are typically more robust against noise. Some methods, however, including the spatio-temporal shape context [11] and spatio-temporal pyramid representations [12], are local feature-based but partially retain structural information between features. The localisation technique presented in this work is similar to these structure-retaining representations.

The first step in our approach is to reduce the video database to a compact representation. As we want our algorithm to operate on realistic datasets, we use local features. Features are detected using Dollar's method [10] at a loosely constant rate with respect to time, at multiple spatial and temporal scales. At each detected point, we extract a spatio-temporal cuboid and apply the HOG3D [13] descriptor. We base our choice of detector on a human action classification evaluation study [14], and the descriptor on the experimental results shown in [13]. Next we assign each of the features one of k distinct codewords/clusters, as in the Bag-of-Words method. To achieve this, we first reduce the feature descriptors' dimensionality using principal components analysis. We then perform k-means clustering on the reduced descriptors, and each feature is assigned to one cluster. Each feature is then represented by a single value – its cluster membership.

We then aggregate these features in a way suitable for rapid localisation. While Yu et al.'s fast method [3] for action localisation can often localise the optimal 3D sub-volume, generating a score for each STIP using Random Forests is too expensive for real-world retrieval. Feature voting[6] is another potential scheme, but we have experimentally determined that such methods are only stable when applied to clean datasets. We instead propose to use a BoW-derived approach to video representation, visualised in part of Figure 1. Each database video is divided into time-slices $t \in T$, of n_f frames, and we create a code-word frequency histogram H_t for all the features within each t. Each histogram is normalised, and n_f is chosen to be approximately half the size of the smallest query that can be searched on. The time-slices do not overlap, as preliminary experiments have shown this does not improve accuracy. While this representation is simple, we show through experiments that it captures sufficient information to localise a human action. All of the aforementioned steps can be processed once on the database in batch – this improves the time efficiency of later user searches.

2.2 Search

Previous work [15,16] on human action localisation typically utilise a trained model – this requires several examples of the target action and the accompanying ground truths. This is not possible in a retrieval context, where only a single query example is provided. Some researchers have made attempts to perform image retrieval with spatial localisation [4,5], and one work focuses on spatio-temporal retrieval and localisation of videos [3]. However, all of the aforementioned techniques are computationally complex, making them unsuitable for real-world retrieval. We present a more efficient system below.

To search, the user provides a video example of the human action they want to find. The system performs feature extraction on this query in the manner described in section 2.1, but a single normalised histogram is generated for the entire length of the query, rather than for time-slices. To search for an action within a single video taken from the database, we first use a simple metric to calculate the similarity between each time-slice histogram and the query histogram H_q. This metric is the histogram intersection:

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

If n_f is chosen appropriately, each time-slice t can only, at best, represent a small fraction of the action being searched for, thus H_q and H_t will not be fully correlated. However, we show in our experiments below that the histogram intersection still generates a stronger response generally for relevant time-slices than irrelevant ones. Aggregating s over all $t \in T$ gives a time-series $S_{q,T}$ representing the similarity s of each $t \in T$ to q.

Analysing $S_{q,T}$, it is possible to find candidate regions for the localised action. One possible approach involves finding local peaks in this series. However, such a method proves too sensitive to noise. Our best method applies thresholding and then candidate segmentation. First, any t where $S_{q,t}$ is below a threshold is discarded. This threshold is one standard deviation above the mean over all $S_{q,T}$. Next, we identify false negative time-slices that occur during an action: if time slice t_i and t_{i+2} are candidates, then t_{i+1} is also considered a candidate. False-negative time-slices are often caused by brief interference with the action, such as a person walking in front of the actor as the action is performed. (The assumption is made that even the shortest action will span several time-slices, making the choice of n_f important.) Finally, remaining candidate time slices without neighbours are also discarded, as candidate regions are unlikely to be only n_f frames in length. (N.B. these last two steps are somewhat analogous to the region growing and shrinking methods found in image segmentation.) After this, any temporally contiguous set of time slices remaining are considered to be a single candidate for the action. The computational complexity of the entire localisation process is $O(|T|)$.

Performing these steps on all videos in the database, the system identifies a large set of candidate regions. A single feature frequency histogram H_c is generated over each candidate region, and $s(H_q, H_c)$ is used to rank the candidates by their relevance to the query. The top X of these are returned to the user. The entire process is shown in Figure 1.

3 Relevance Feedback

We can use relevance feedback (RF) to iteratively improve both the ranking and localisation aspects of our algorithm. After an initial search, RF can improve results by combining the original query with user feedback about the quality of the initial results, to generate a more discriminative query. Usually this second, more discriminative query will return better results than the original query. To date, RF has been used mostly in the image retrieval domain [17,18], but has also been applied to human action retrieval in more recent years [2].

In this work, relevance feedback occurs after the localisation and retrieval have been performed once as described above to give an initial ranking of videos. The user provides binary feedback on the relevance of several highly-ranked results, and the histograms associated with these results are used to train new localisation and retrieval algorithms. To improve localisation, we use the feedback histograms and the original query histogram to train an SVM, with the histogram intersection shown in equation 1 as the SVM's kernel. Then, to calculate the relevance of each time slice t, we measure the distance from the SVM's hyperplane to H_t. The rest of the localisation algorithm proceeds as described in §2.2. To improve our ranking with relevance feedback, we replace the histogram intersection shown in equation 1 with a simple query expansion metric that only utilises positive feedback *pos*. This query expansion takes the following form:

$$D_{t,pos} = min(s(H_p, H_t)|p \in pos\}) \tag{2}$$

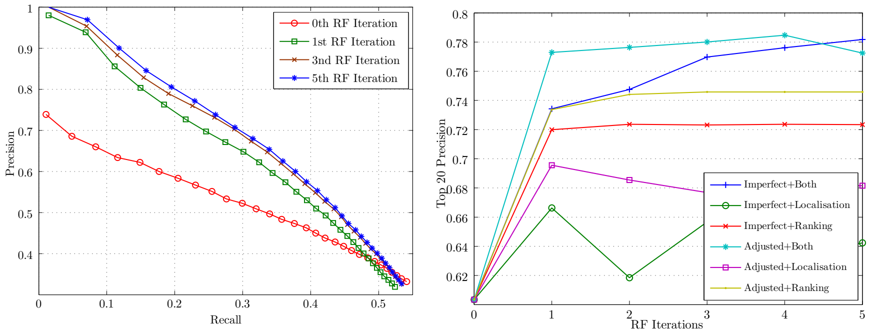
Applying relevance feedback to a system with localisation results in an unusual issue. Results returned to the user are often neither completely irrelevant nor completely relevant – a result may be mostly relevant, but imperfectly localised. In light of this problem, does the user have to manually re-localise the feedback both spatially and temporally before rerunning the query? Two methods of providing feedback are considered in our experiments.

4 Experiments

4.1 Setup

In this section, we describe experiments to demonstrate our algorithm. We use the MSR II human action dataset [19] based on its popular use in other human action localisation works. This dataset consists of 54 videos, totalling approximately 46 minutes of footage, containing 203 total examples of actions. The three classes of action are: handwaving, handclapping and boxing. These actions are performed orthogonally to the camera in a very similar fashion to one other, but the localisation is made more difficult due to various issues such movement of action-unrelated actors in the background and spatially/temporally overlapping actions.

During feature extraction, we extract, on average, 3 features per frame, at 4 different spatio-temporal scales. Because boxing can be performed to either the



(a) (b)

Fig. 2. (a) Precision-recall of localisation+retrieval after having performed relevance feedback iteratively. The improvements of successive RF iterations can be seen clearly here. (b) The precision (% of true positives) of the top 20 results after relevance feedback in different scenarios. We show both imperfect and user-adjusted relevance feedback. We also show the effects of applying relevance feedback to the localisation and ranking algorithms in isolation, to see their contributions to the overall improvement in precision.

left or the right, all features are also mirrored on the y-axis, giving an average of 24 features per frame. In the creation of the feature codebook, we use PCA to retain 95% of the total variance, and for clustering $k = 1000$.

Leave-one-out cross validation is performed, treating each of the 203 actions as the query in turn, averaging results over all runs. We use the following method to determine the accuracy of our localisation: let $L(E, G) = \frac{length(E \cap G)}{length(E \cup G)}$ where E is the temporal extent of the estimated action, and G is the temporal extent of the closest ground truth. An action is considered successfully localised when $L(E, G) \geq 0.5$. To simulate a user's relevance feedback, we use the ground truth to determine up to 5 examples of each of positive and negative feedback.

4.2 Results

Figure 2a shows a precision-recall graph using our optimal setup over the whole MSR II dataset, after various iterations of imperfect relevance feedback. Precision and recall are usually used in the context of binary relevance. To use these metrics with localised results, however, we need a way of determining whether an imperfectly localised result is still relevant. In [3], the authors determined relevance of a result differently for precision and recall. However, we contend that this method creates an unintuitive statistic, which cannot be interpreted in the same way as traditional precision-recall. We use the single, stricter criterion L, defined above for both precision and recall.

The effects of relevance feedback on the precision of the top 20 results is shown in Figure 2b. Only results that satisfy $L(E, G) \geq 0.5$ are considered for positive relevance feedback. Negative relevance feedback is taken from results where $L(E, G) = 0$. We have considered both "imperfect" feedback, unmodified from the results, and user-adjusted feedback, where the spatial and temporal extents of positive feedback are modified to exactly match the ground truth. While user-adjusted feedback performs better, imperfect feedback still shows a significant improvement after one and subsequent rounds of relevance feedback. This could have practical implications for the usability of a retrieval system with localisation. We also consider the effects of applying relevance feedback to only localisation, and only retrieval, to show their individual contribution to the overall improvement in precision.

We ran our experiments using MATLAB R2009a, on a 2.9GHz Core 2 Duo PC, with 4GB RAM, running 32-bit Windows 7. The database is 46 minutes in length, and the mean length of a query video is 5.7 seconds. The average time for a query with and without relevance feedback are 0.298 without relevance feedback, and 0.847 with relevance feedback, excluding offline computational costs. These times are at least an order of magnitude better than previous results. Our algorithm could also potentially be accelerated through programmatic optimisation, or its computational complexity reduced through a search of hierarchical time-slices according to size.

5 Discussion

We have created and demonstrated the use of an efficiency-focused video retrieval system with localisation. Our relatively simple localisation search can still give practical results, but completes in a fraction of the time of any previously reported algorithm. We have additionally looked at the application of relevance feedback in a retrieval context, and have shown that both user-adjusted and imperfect feedback can be used to improve results significantly.

Our proposed method's primary weakness, compared to existing algorithms, lies in its inability to separate spatially-distinct background noise from the results, which may cause incorrect ranking of the candidates. This has not significantly affected our results on the MSR II, but on more complex datasets, such as the HMDB[20] it may become a problem, particularly as the number of actions may decrease accuracy[21]. In future work, we will investigate ways to spatially isolate actions without the performance costs associated with branch-and-bound derived methods. Additionally, further experimentation needs to be done on more complex datasets, such as HMDB [20], to prove the algorithm's general applicability.

References

1. Zhang, H.J., Wu, J., Zhong, D., Smoliar, S.: An Integrated System for Content-based Video Retrieval and Browsing. Pattern Recognition 30(4), 643–658 (1997)
2. Jones, S., Shao, L., Zhang, J., Liu, Y.: Relevance Feedback for Real-World Human Action Retrieval. Pattern Recognition Lett. 33(4), 446–452 (2012)

3. Yu, G., Yuan, J., Liu, Z.: Unsupervised Random Forest Indexing for Fast Action Search. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, pp. 865–872 (2011)
4. Rahmani, R., Goldman, S.A., Zhang, H., Krettek, J., Fritts, J.E.: Localized Content Based Image Retrieval. In: ACM SIGMM Int. Conf. Multimedia Inform. Retrieval, pp. 227–236 (2005)
5. Zhang, D., Wang, F., Shi, Z., Zhang, C.: Interactive Localized Content Based Image Retrieval With Multiple-Instance Active Learning. Pattern Recognition 43(2), 478–484 (2010)
6. Ryoo, M., Aggarwal, J.: Spatio-temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities. In: IEEE Int. Conf. Comput. Vision, pp. 1593–1600 (2009)
7. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976–990 (2010)
8. Davis, J.W., Bobick, A.F.: The Representation and Recognition of Human Movement Using Temporal Templates. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, p. 928 (1997)
9. Laptev, I.: On Space-Time Interest Points. Int. J. Comput. Vision 64(2-3), 107–123 (2005)
10. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior Recognition via Sparse Spatio-Temporal Features. In: Proc. IEEE Workshop Visual Surveillance and Performance Evaluation Tracking and Surveillance, pp. 65–72 (2005)
11. Shao, L., Du, Y.: Spatio-temporal Shape Contexts for Human Action Retrieval. In: Proc. Int. Workshop Interactive Multimedia Consumer Electronics, pp. 43–50 (2009)
12. Choi, J., Jeon, W.J., Lee, S.-C.: Spatio-temporal pyramid matching for sports videos. In: ACM SIGMM Int. Conf. Multimedia Inform. Retrieval, pp. 291–297 (2008)
13. Kläser, A., Marszałek, M., Schmid, C.: A Spatio-Temporal Descriptor Based on 3D-Gradients. In: Proc. British Mach. Vision Conf., pp. 995–1004 (2008)
14. Shao, L., Mattivi, R.: Feature Detector and Descriptor Evaluation in Human Action Recognition. In: Proc. ACM Int. Conf. Image and Video Retrieval, pp. 477–484 (2010)
15. Kläser, A., Marszalek, M., Schmid, C., Zisserman, A.: Human Focused Action Localization in Video. In: International Workshop on Sign, Gesture, Activity (2010)
16. Sullivan, J., Carlsson, S.: Recognizing and Tracking Human Action. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 629–644. Springer, Heidelberg (2002)
17. Tong, S., Chang, E.: Support Vector Machine Active Learning for Image Retrieval. In: Proc. ACM Multimedia, pp. 107–118 (2001)
18. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 28, 1088–1099 (2006)
19. Cao, L., Liu, Z., Huang, T.: Cross-dataset Action Detection. In: Proc. IEEE Conf. Comput. Vision and Pattern Recognition, pp. 1998–2005 (2010)
20. Kuehne, H., Poggio, H.: HMDB: A Large Video Database for Human Motion Recognition. In: IEEE Int. Conf. Comput. Vision (2011)
21. Reddy, K., Shah, M.: Recognizing 50 human action categories of web videos. Mach. Vision and Applicat., 1–11 (2012)