Action Retrieval with Relevance Feedback on YouTube Videos

Simon Jones, Ling Shao
University of Sheffield
Sheffield
UK
simon.m.jones@sheffield.ac.uk, ling.shao@sheffield.ac.uk

ABSTRACT

Content-based retrieval systems are becoming increasingly relevant for managing large multimedia databases, such as those found on the Internet. In this paper, we investigate applying content-based retrieval with relevance feedback to the popular YouTube human action dataset[8], using a variety of methods to extract and compare features, in order to determine the most accurate techniques in this setting. Among other techniques, we explore soft-assignment codebook clustering, feature pruning, motion and static features, Adaboost and ABRS-SVM for relevance feedback. We evaluate the performance of several different systems to find the best combination of techniques for human action retrieval. We demonstrate that existing relevance feedback methods do not work well for YouTube media, and that a naive algorithm consistently outperforms these.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Relevance feedback, Clustering*; I.2 [**Artificial Intelligence**]: Vision and Scene Understanding—*Motion, Video analysis*

General Terms

Experimentation, Performance

Keywords

Action Recognition, Youtube Dataset, Feature Pruning, Soft-Assignment Clustering

1. INTRODUCTION AND RELATED WORK

Internet video websites, such as that provided by YouTube, have over the past few years exploded in size, as users find them a convenient source of information and entertainment. However, as the video databases that support these sites continue to grow, finding a video relevant to a user's interest is becoming increasingly difficult. Conventional keyword searches, while very effective for retrieving text documents, rely on manual user annotation to retrieve multimedia. Such annotation is innately incomplete and prone to human error. Content-based information retrieval (CBIR) – that is, retrieval that analyses the content of images and videos to find results – has commonly been posed as the solution to this problem. The accuracy and computational cost of such techniques are still prohibitive, however – especially when applied to videos – so much research is needed before CBIR can see any level of real world use.

Research has shown promising results in applying CBIR to video retrieval as early as 1994[1]. Nevertheless, while there have been varied attempts to improve accuracy in image retrieval, until recently video retrieval has been ignored. Some attempts to perform video retrieval include Jiang et al.[5], who demonstrated how to optimise the Bag of Features model for this purpose. Moving towards practical systems, Jin and Shao[6] incorporate a single iteration of relevance feedback. Yan et al.[15] use pseudo-negative relevance feedback in order to improve results.

In this paper, we evaluate the effectiveness of a CBIR system with relevance feedback to retrieve human actions, and evaluate several different techniques for action representation in such a system. We focus on the retrieval of human actions in particular for two reasons. Firstly, the vast majority of video media contains humans as subjects of interest, making it very important to recognise their actions. Secondly, human actions can potentially have huge intraclass variability, depending on, for instance, the actor performing the action, the clothing the actor is wearing, or how quickly the action is performed. There has been a considerable level of research done on human action recognition to date, but these techniques have not yet reached usable accuracies except in very primitive videos[10, 2, 12, 11]. It is clear that human actions present a significant challenge that must be overcome in any potential real-world system. We show that despite these and other problems we can achieve a relatively high retrieval performance on the UCF YouTube Action dataset[8], utilising relevance feedback and a highly informative video representation.

2. ACTION REPRESENTATION

2.1 Feature extraction

When dealing with a large video database, it is infeasible to perform a search on the raw video due to the huge amount of data involved. Therefore, some level of pre-processing to extract compact key features of the videos is necessary. In

ICIMCS'11 August 5-7, 2011, Chengdu, Sichuan, China

a real world scenario search queries cannot be predicted in advance, and so the extracted features must maximise information retention. Additionally, for real world videos the features must be highly invariant in order to deal with the expected high intraclass variance; intraclass variance is especially problematic in the context of CBIR, as the effective training set is, before relevance feedback, just one sample.

Based on these considerations, we decided that local features would give the best representation. For our system, we utilise two types of local features, as described in Liu et al.[8]: Dollar's method[3] for extracting motion features, and SIFT[9] for static features. A modified form of the feature pruning from Liu et al.[8] is performed to remove noisy features generated by ego motion and moving backgrounds, and retain only the most discriminative features. By combining static and motion features, we increased the total discriminative information available in the representation, for greater accuracy.

2.2 Codebook generation

Using PCA and K-means clustering, we generate a feature codebook for both types of feature from the dataset. Each video sequence is then represented in the database as a histogram of codeword frequency. This allows us to use the χ^2 distance on the histograms as a simple metric to compare two videos' similarity. We also explore soft-assignment of codewords, where each feature is assigned a proportional "belonging" to each codeword by the following equation:

$$B_{f,C} = \frac{e^{\frac{d_{f,C}}{\sigma^2}}}{\sum e^{\frac{d_{f,C}}{\sigma^2}}} \quad (1)$$

where $B_{f,C}$ is a vector of the proportional belonging of feature f to each of the clusters C, $d_{f,C}$ is a vector of the Euclidean distance between the feature and every cluster centroid, and σ is a constant which determines the "hardness" of the soft assign. In the case of soft assignment, histograms are created from soft-assignment by aggregating $B_{f,C}$ over all features $f \in F$.

3. VIDEO RETRIEVAL

3.1 System overview

Here, we briefly outline the operation of our system. Initially, the video database is preprocessed; features and the codebooks are extracted according to the methods described above, and these are used to generate a histogram of codeword frequency for each video.

Then, in order to perform a search the user provides a sample video of the desired action, known as the query. Feature extraction is performed on this video in an identical matter to the video database. An initial set of the top X results from the database is returned, based on the χ^2 distance between histograms. The user marks several of the returned videos as positive or negative examples. The system generates an improved model using a relevance feedback model, and using this attempts to give an improved order of results. The user can iteratively perform relevance feedback and retrieve improved results until the results are satisfactory, or no further improvement is observed. Each search iteration takes $O(n)$ time, where n is the size of the video database.

3.2 Relevance feedback

We look at three different techniques for incorporating relevance feedback into CBIR: the Asymmetric Bagging and Random Sampling SVM (ABRS-SVM)[13], Adaboost[4], and a naive algorithm modelled purely on positive feedback.

ABRS-SVMs have been used to great effect for relevance feedback – they take into account that most systems will result in far more negative than positive feedback. Adaboost, on the other hand, is far less specific in purpose, and is commonly used as the classifier in action recognition scenarios. We also introduce a simple naive algorithm to give a baseline performance.

3.2.1 ABRS-SVM

ABRS-SVMs are designed to compensate for three problems:

- The number of feedback samples given is usually quite small, meaning an ordinary SVM will be unstable.
- There will often be more negative feedback than positive for very noisy/complex datasets, resulting in a biased hyperplane.
- The dimensionality of the feature vector is often much greater than the number of feedback samples, leading to overfitting.

They achieve this by random sampling on both the feature space and the negative sample space, resulting in several weak SVMs, which are combined into a single regressor using the Bayes Sum Rule. The input to this regressor is a histogram in the database, and the output is a single numerical value, performing as a similarity measure by which we order the results. ABRS-SVMs have been used with some success on human action datasets previously [7].

3.2.2 Adaboost

Adaboost is a general purpose algorithm for combining many weak classifiers (classifiers with low individual discriminative power) into a highly discriminative strong classifier. In this context, we treat each bin in the histogram as a weak classifier, and want to use Adaboost to select the most discriminative of these. Using this assumption, we can apply the most basic binary Adaboost algorithm, as CBIR can be treated as a special case of the binary classification task; results are either relevant or irrelevant. We order the results by the sum of weak classifiers that give a positive result.

3.2.3 Naive algorithm

Given only positive feedback, this algorithm models a distance metric as follows:

$$D_{h,pos} = min(\{\chi^2_{h,p} | p \in pos\}) \quad (2)$$

where $\chi^2_{h,p}$ is the χ^2 distance between database histogram h and positive feedback example p, and pos is the set of all positive feedback.

4. EXPERIMENTS

4.1 Setup

We use the UCF YouTube Action Dataset for our experiments; it is drawn from existing Internet media, and as such

is particularly suitable for testing our methods. It consists of 11 different action categories, most of which are sports activities, such as riding a bike, or playing tennis. Each of these categories is divided into 25 scenes, and each of these scenes is sub-divided further into several instances of the action. There are 1599 action instances in total. Figure 1 shows some examples of action instances taken from the dataset.

Figure 1: Examples of actions from the YouTube action dataset. Top left: basketball shot; top right: bike riding; bottom left: horse riding; bottom right; tennis swing

To perform CBIR on this dataset, we leave out one scene from the database, and then perform a search with each of the action instances from that scene as the query. Leaving out a scene at a time is necessary, because action instances from the same scene as the query would be too similar, skewing the results unrealistically. We perform 9 iterations of relevance feedback for each search, retrieving the top 20 results and calculating the percentage of correct returned results at each iteration.

We perform several types of feature extraction and codebook clustering in order to show their effects on the system. These are shown in Table 1. Motion features are extracted from each video at an average rate of 4 per frame, before pruning is performed. Static features are extracted from 9 keyframes taken from the video, and pruning performed, removing static features that are not near any motion features. After this process, each action instance contains between X and Y motion features, and X and Y static features. For codebook generation, we reduce the features' dimensionality using PCA to retain 95% of variance, and then use k-means clustering, with $k = 400$, for both types of feature. Features are assigned to codewords variously by hard and soft assignments. For our soft assignment clustering experiments, we set $\sigma = 1.2$. To create hybrid features, we concatenate the motion and static feature histograms, resulting in 800 bin histograms.

For the ABRS-SVM experiment, we set T_s and T_f to 5, and for Adaboost, we select a total of 50 weak classifiers on each iteration.

4.2 Results

Regarding feature extraction, we can clearly see in Figure 2 that a combination of static and motion features outperforms either individual feature; using more sources of information will generally result in better discrimination.

Table 1: Experiments Performed

Exp.	Feature type	Cluster assign	RF
1	Static	Hard	Naive
2	Motion	Hard	Naive
3	Hybrid	Hard	Naive
4	Hybrid	Soft	Naive
5	Hybrid	Hard	Adaboost
6	Hybrid	Hard	ABRS-SVM

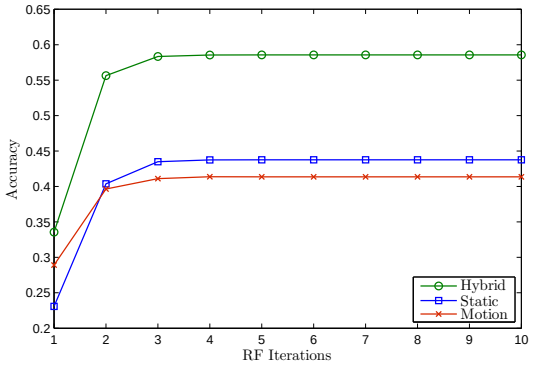

Figure 2: Comparison of feature extraction methods on accuracy over 9 iterations of relevance feedback. (Experiments 1, 2 and 3)

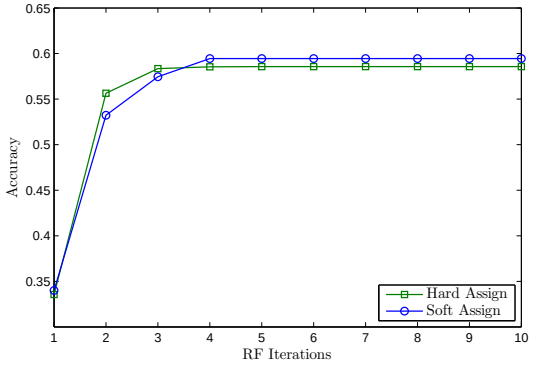

Figure 3: Comparison of hard and soft assignment clusters. (Experiments 3 and 4)

There is little difference between using soft and hard assignment clustering in Figure 3. Soft assignment might prove useful in a situation where features are extremely sparse, or when the codebook size is incorrectly chosen – in this case, soft assignment would lose less information in quantisation than hard assignment. However, here, with numerous features and an appropriate codebook size, soft assignment offers no benefit over hard assignment. Additionally, soft assignment is parametric so generally non-parametric hard assignment is favoured.

Most surprisingly, ABRS-SVM and Adaboost – both sophisticated relevance feedback tools – fail to generate promis-

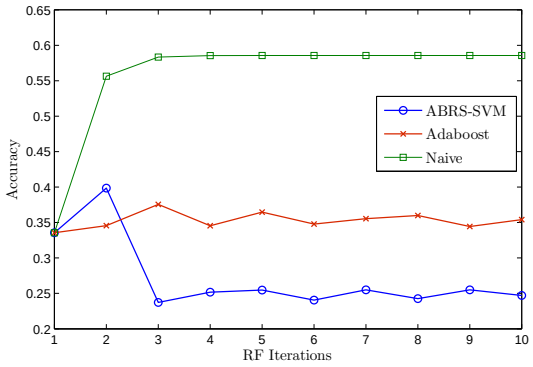

Figure 4: Comparison of relevance feedback methods. (Experiments 3, 5 and 6)

ing results, seen in Figure 4. Instead, our naive algorithm gives the best and most consistent results after multiple iterations of relevance feedback. We attribute this to the high intraclass variability of actions found in the UCF YouTube Action dataset; our naive algorithm is best suited to data with high intraclass variability, as it does not make assumptions about the correlation between the feature vectors. In contrast, both ABRS-SVMs and Adaboost assume that a class' feature vectors are all clustered together and entirely separable from other feature vectors.

On close observation, we can see a spike in performance after the first round of ABRS-SVM relevance feedback, which is quickly lost in subsequent iterations. We attribute this to the properties of the dataset, where multiple instances are extracted from one scene. In the first round of feedback, we retrieve positive feedback from a variety of different scenes; however, in subsequent rounds of relevance feedback, the positive feedback may be selected from a single scene, so the positive feedback set will lose intraclass variability, and lessen the discriminative power of the ABRS-SVM. In future work, active learning techniques, such as those described in Wang and Hua[14], could be explored as an alternative to improve results after relevance feedback.

5. CONCLUSIONS

In this paper we have shown how an effective real-world video retrieval system can be made utilising relevance feedback. We have compared several different methods of action representation and relevance feedback. In our experiments we showed that using a combination of static and motion features, along with soft cluster assignment, gives a superior representation of human action in the YouTube dataset for action retrieval. We demonstrated that state-of-the-art techniques for relevance feedback ABRS-SVM and Adaboost perform worse than a simple naive algorithm for this task. This work suggests that more research should be conducted into more effective methods of relevance feedback for Internet video.

6. REFERENCES

[1] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu. Content-based browsing of video sequences. In *Proc. of ACM International Conference on Multimedia*, pages 97–103. ACM, 1994.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. of IEEE International Conference on Computer Vision*, page 1395, 2005.

[3] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. of the Second European Conference on Computational Learning Theory*, pages 23–37, 1995.

[5] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proc. of ACM International Conference on Image and Video Retrieval*, pages 494–501, 2007.

[6] R. Jin and L. Shao. Retrieving human actions using spatio-temporal features and relevance feedback. In L. Shao, C. Shan, J. Luo, and M. Etoh, editors, *Multimedia Interaction and Intelligent User Interfaces: Principles, Methods and Applications*. Springer-Verlag, Sept. 2010.

[7] S. Jones, L. Shao, J. Zhang, and Y. Liu. Relevance feedback for real-world human action retrieval. *Pattern Recognition Letters*, In Press, Accepted Manuscript, 2011, doi:10.1016/j.patrec.2011.05.001.

[8] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, June 2009.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[10] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *Proc. of IEEE International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[11] L. Shao, L. Ji, Y. Liu, and J. Zhang. Human action segmentation and recognition via motion and shape analysis. *Pattern Recognition Letters*, In Press, Accepted Manuscript, 2011, doi:10.1016/j.patrec.2011.05.015.

[12] L. Shao and R. Mattivi. Feature detector and descriptor evaluation in human action recognition. In *Proc. of ACM International Conference on Image and Video Retrieval*, pages 477–484, 2010.

[13] 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 Transactions on Pattern Analysis and Machine Intelligence*, 28:1088–1099, 2006.

[14] M. Wang and X.-S. Hua. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol.*, 2:10:1–10:21, February 2011.

[15] R. Yan, A. G. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *Proc. of ACM International Conference on Multimedia*, pages 343–346, 2003.