Eastern Rituals Search Engine (ERSE)
Retrieving Top-K Videos Based on Motion Similarity

Ayman Aboueloula, David Cian, Javiera Quiroz
Laboratory for Experimental Museology (eM+), EPFL

17 December 2020

Abstract. The Eastern Rituals Search Engine (ERSE) is such a CBIR engine that enables searching a database of human motion videos for similar videos. The search pool is composed of a small part of the large Remaking of the Confucian Rites database. The similarity calculation is based on the analysis of the position of key joints of the person identified in the video on a frame by frame basis. We build on the Query-by-Dancing retrieval method and add several extra features.

1 Introduction

Finding relevant videos similar to a target video posses a major difficulty when faced with a large body of video data, due to our limited data processing capabilities - a human needs to watch the entire body of video data to complete this task. It is desirable then to create a content-based information retrieval (CBIR) system, in other words, a domain-specific video search engine. We find a surprisingly good performance of our baseline implementation, albeit in a limited setting, obtaining an average recall of 62% on videos taken from a side view of the actors when returning the 10 best results, as well as qualitatively satisfactory results.

2 Dataset

The dataset consists of three folders of videos (*bodyvocab*, *posecuts_frontview* and *posecuts_sideview*), each containing videos in QuickTime (file extension .mov) format, along with an annotation file in CSV format. The videos are part from the Remaking of the Confucian Rites database in which elements of ancient Eastern rituals are re-enacted. The *bodyvocab* folder contains videos of ritual elements shot from both front and side views, whereas the other two contain the same videos but shot from different views, namely front and side respectively. The number of videos and memory size are displayed in the table below.

Folder name	Number of videos	Memory size
bodyvocab	72	1.26 GB
posecuts_frontview	40	242 MB
posecuts_sideview	40	245 MB

Table 1 – Size and quantity of videos in each folder of the dataset.

During the project, we concentrate on the folders containing videos from one view only to work with consistent visual features. These videos vary in length from 4 to 35 seconds, and contain only one subject in static poses (such as sitting or kneeling) as well as performing repeated movements (such as walking from one side to the other), which increases the complexity of the problem.

The annotation file in each of the folders contains information of the motions that can be seen in each video file under the column called Motion Tags, describing the motions by verbs and adverbs. These descriptions are then used to evaluate the model produced (see Section 3.3).

3 Methodology

3.1 Problem Statement

3.1.1 Workflow

Our hypothesis is that the movement of limbs correlates strongly with the motion tags, which is why we consider it valid to base our implementation of ERSE on Query-by-Dancing [4]. We summarize the main principles below for the reader.

3.1.2 Evaluation Metrics

Part of what makes evaluating a search engine results page (SERP) difficult is the subjectivity of this task. Indeed, the relevance of the SERP depends on the specific user. For this project, the main aim is retrieval as it is desired to instill a high degree of trust in our search engine in the end-user, thus more emphasis is placed on recall in the evaluation of the model.

To compute the recall of the model- as well as other metrics such as accuracy, precision and F1 score- it is necessary to construct a confusion matrix. For this, the definition of the ground truth similarity between videos was based on the motion tags and measured by a value termed as the Correct Motion Tag Ratio.

	Correct Motion Tag Ratio > 0.5	Correct Motion Tag Ratio < 0.5
Returned in SERP	True Positive	False Negative
Not returned in SERP	False Negative	True Negative

Table 2 – Confusion matrix defined given a input video.

The Correct Motion Tag Ratio is defined as the Jaccard similarity coefficient between the sets of the motion tags of each video. For videos a and b with sets A and B containing their respective motion tags, the correct motion tag ratio between them is given by:

$$J(a,b) = \frac{|A \cap B|}{|A \cup B|}$$

These quantitative evaluation metrics are useful for measuring the impact of changes to the baseline system, but they are not very representative to the reader on their own. For this reason, it is useful to get a feel for the system through its web interface, which allows the user to see the video results of a query lined up.

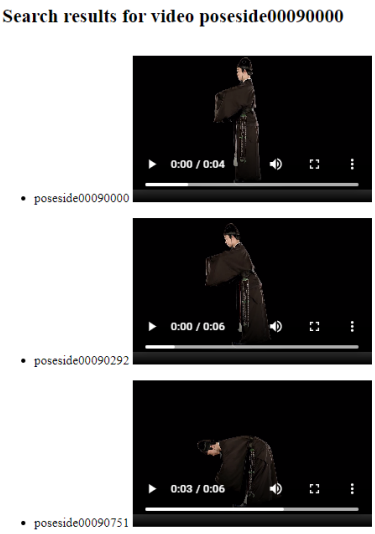


Figure 1 – Sample search engine results page (SERP)

3.2 Feature Extraction and Selection

The angular position, velocity and acceleration of key joints are extracted for each frame. The OpenPose library was used to extract the positions of keypoints in 2D from the raw videos as a JSON file with 25 body parts for each person identified in the video [1]. While not the only open-source pose estimation library, OpenPose was chosen due to its greater accuracy on keypoints detection compared to the lighter PoseNet [3].

We define a directed limb as an ordered pair of joints: 14 anatomically consistent limbs are defined in such a fashion[1]. Following this, we compute the clockwise angle of each directed limb from the vertical line, yielding 14 angles. We then decompose these angles into their projection along x and y as shown in Figure 2 by computing the sine and cosine of the angle respectively, yielding 28 features. We compute the speed as a discrete difference between angles and the acceleration as a discrete difference between speeds, yielding 84 features per frame. If a person or one of the keypoints is not identified its angle is set to 0. For videos with multiple people, only the keypoints of the first body are kept in the feature vector as usually all subjects in the same video are performing the same movements.

The advantage of working with the projected angles is that the components are between 0 and 1, thus there is no large bias due large magnitude of coordinates of given joints. Similarly, this approach ensures that the feature vector is independent of the relative position of the subject on the frame which increases the resilience of the model.

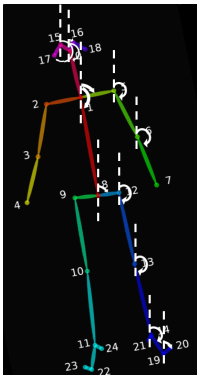


Figure 2 – Pose Output Form (BODY_25) with annotated angles [2]

3.3 Similarity Search

The similarity between videos was measured by the similarity between frames. Due to the different lengths of each video, we defined the distance between videos a and b as the average distance between all pairwise combinations of their frames, where the distance between frames refers to the the Cartesian distance between the feature vectors $f_a(n)$ and $f_b(m)$ at a given frame n and m. We consider the distance $D(a, b)$, defined by:

$$D(a,b) = \frac{1}{N_a N_b} \sum_{n=1}^{N_a} \sum_{m=1}^{N_b} d(f_a(n), f_b(m)) \tag{1}$$

to be inversely proportional to the similarity between videos.

A frame by frame comparison is not possible for our dataset as the length of the videos varies, and even though a number of close neighbours could be compared instead of all the frames, the appropriate number of neighbouring frames to compare would highly depend on the dataset, making it difficult for the search engine to work on a larger more diverse set of videos and thus this approach was not chosen.

[1]The 14 directed limbs defined are: neck, right and left shoulder, right and left elbow, spine, right and left hip, right and left knee, right and left angle, and right and left toes

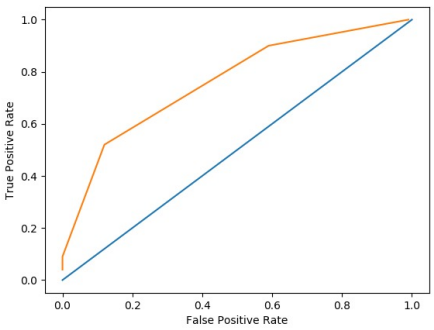
4 Results

The baseline implementation of the search algorithm using the top-k stopping criterion is evaluated separately for the front and side view search pools. We report the average values for each metric, with a k of 10.

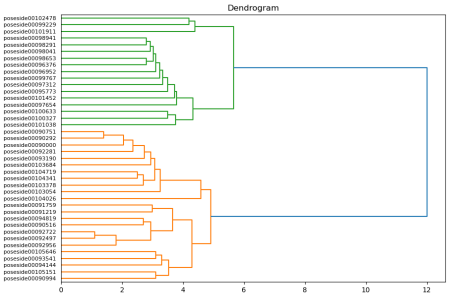
Folder name	Accuracy	Recall	Precision	F1 score
posecuts_frontview	0.770	0.619	0.317	0.390
posecuts_sideview	0.782	0.622	0.342	0.412

It seems a bit unnatural to stop after a fixed number of results, so we also introduced a more natural stopping criterion, which returns all videos similar enough to the input video (as defined by a threshold). The SERP is subject to the following tradeoff: the better the recall, the more trust the user places in the search engine, but the less relevant the results become, which diminishes its utility. Emphasis is placed on the recall, but it is useful to have a reference when setting this threshold, which the receiver operating characteristic (ROC) curve in Figure 3a provides.

Additionally, agglomerative clustering with the Ward linkage criterion was performed based on the pairwise distance, calculated as specified in [Eq. 1]. The dendrogram on Figure 3b shows two distinct groups which correspond in fact to the static and dynamic poses from the videos. This goes in fact to remark that the model was able to efficiently separate the videos by the major inconsistency between them: their termporal evolution. Clustering has many potential uses: it can show hidden structure in the data, as shown earlier, or it can be used to speed up retrieval, which is an important factor to consider when scaling the dataset.



(a) ROC curve for the natural search stopping criterion



(b) Dendrogram obtained from the clustering of videos based on their pairwise distance

5 Discussion and Conclusion

The authors of this paper would like to think of it as a proof of concept, expanding the idea behind Query-by-Dancing in several directions. First of all, as qualitative examination using the web interface shows, the approach is viable for domains other than dancing. Second, we come up with a more user-friendly approach to use such a CBIR system, using a web interface and a natural stopping criterion. Finally, we show that the pioneering work of Query-by-Dancing allows for much more than simple retrieval, as clustering methods can reveal hidden structure in the data which can further help domain experts in their analysis.

Directions for future work include automated representation learning for videos, using for instance an autoencoder. ERSE does include such a feature, but results were modest due to the small dataset size. A time and space complexity analysis of the code would be another interesting direction, as the problem is quite computationally intensive, and one of the main goals of such a search engine is to handle massive datasets. Additional structure extraction tools would also be interesting for domain experts: in our original application context, it is fascinating to think of ERSE as Sherlock's looking glass for unearthing information about ancient Eastern Asian Confucian rituals.

References

[1] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186, January 2021. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2] Gines Hidalgo. Openpose output, 2020. [Online; accessed 14 December 14, 2013].

[3] ParleyLabs. Exploration: Pose estimation with openpose and posenet, 2020. [Online; accessed 14 December 14, 2013].

[4] Shuhei Tsuchida, Satoru Fukayama, and Masataka Goto. Query-by-Dancing: A Dance Music Retrieval System Based on Body-Motion Similarity. pages 251–263. January 2019.