ReEnact: Sketch based Choreographic Design from Archival Dance Footage

Stuart James
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
s.james@surrey.ac.uk

Manuel J. Fonseca
Dept. of Computer Science and Engineering
INESC-ID/IST/TU
Lisbon, Portugal
mjf@inesc-id.pt

John Collomosse
Centre for Vision Speech and Signal Processing
University of Surrey
Guildford, UK
j.collomosse@surrey.ac.uk

ABSTRACT

We describe a novel system for synthesising video choreography using sketched visual storyboards comprising human poses (stick men) and action labels. First, we describe an algorithm for searching archival dance footage using sketched pose. We match using an implicit representation of pose parsed from a mix of challenging low and high fidelity footage. In a training pre-process we learn a mapping between a set of exemplar sketches and corresponding pose representations parsed from the video, which are generalized at query-time to enable retrieval over previously unseen frames, and over additional unseen videos. Second, we describe how a storyboard of sketched poses, interspersed with labels indicating connecting actions, may be used to drive the synthesis of novel video choreography from the archival footage.

We demonstrate both our retrieval and synthesis algorithms over both low fidelity PAL footage from the UK Digital Dance Archives (DDA) repository of contemporary dance, circa 1970, and over higher-definition studio captured footage.

Categories and Subject Descriptors

H.4 [**Content Analysis and Indexing**]: Miscellaneous; D.2.8 [**Info. Search and Retrieval**]: Metrics—*Pose Estimation, Image Retrieval, Video Synthesis*

1. INTRODUCTION

The performing arts are increasingly turning to digital archives for dissemination. Dance in particular has launched several major online archives of historic dance footage, including the EU GAMA and UK Digital Dance Archives (DDA). Existing solutions for searching this footage rely on text, which focuses on archival metadata rather than the choreography itself.

In this paper we describe a novel system for searching dance video content directly, using free-hand sketches of human pose – the essential element of choreography. Moreover, we enable the synthesis of novel choreographic video sequences from a sequence (or 'storyboard') of such sketches. The novel choreography is composed using fragments of original archive footage seamlessly combined to match the intention expressed in the sketched storyboard.

Our system accepts a free-hand drawing of a stick-man in the desired pose as input, and searches through footage to retrieve frames containing similar poses. This is a challenging task, as archival dance footage is frequently low fidelity with poor definition and contrast bleaching due to transfers between analogue media over the years. Such footage has recently been shown [27] to frustrate the automated parsing of an explicit estimate of pose or labelling of limbs using state of the art techniques [1, 9, 32].

Our first contribution is a technique for matching such footage to sketched stick-men. To the best of our knowledge we are the first to attempt this in 2D footage, and without relying upon explicit limb pose estimation within content.

Our second contribution is a technique for synthesizing new video-realistic choreographic footage from a sequence of sketched stick-men. Synthesis is performed by seamlessly stitching together video fragments from archival footage to produce new choreographic sequences that follow the poses (and interspersed actions) specified by a user sketched storyboard. This novel interactive authoring tool enables users to create choreography along the theme of past work, directed using a novel set of pose and action transition requirements. In many cases the creation of such sequences would be impossible due to unavailability of historic costumes or the performers depicted in the footage.

2. RELATED WORK

Our work is aligned with prior research exploring the sketch based retrieval (SBR) of images and video. The early SBR systems of the nineties performed image retrieval using query sketches comprising blobs of color texture [20]. Sketched line-art depictions were later matched against edge contours extracted from clip-art vector graphics [30], or from images via edge detection. The latter were enabled by global image descriptors such as Curvature Scale Space [25] and edge orientation histograms [10] offering fast matching. Optimization approaches that fit sketched contours to edges within the image have also been explored [8], offering higher accuracy in the presence of clutter but with a time complexity that hinders scalability. Recently, SBR has been revisited using local image descriptors in a Bag of Visual Words (BoVW) framework. A modified version of the Histogram of Gradient descriptor (GF-HOG) was proposed by Hu et al. [17] to adapt the BoVW pipeline to SBR. An adaptation to the HOG sampling strategy for BoVW was proposed by Eitz et al. [11]. Wang et al. described a complementary approach to scalable SBR using an inverted index of local edgels [5]. Sketches of shape have been augmented with additional information such as motion cues to enable matching of moving

ICMR '14 Apr 01-04 2014, Glasgow, United Kingdom

shapes to video [6]. Sketches annotated with semantic labels have been proposed for hybrid text-sketch matching [24, 18].

These modern approaches offer robust solutions to the SBR problem of shape match for general objects, but lack the power to clearly discriminate between the different poses within a dance performance. Although visual search using human pose has been explored within the context of photographs [13, 27], or using natural interfaces such as Kinect [21], the topic has not been explored in image based SBR. We tackle the problem of sketch based pose search by parsing the sketch into a parameterized skeleton (stick man) representation [14], and learning the mapping between that parameterized space and the space of appearance descriptors extracted from the video.

Our concatenation approach to synthesizing video choreography falls squarely within the domain of example-based synthesis (EBS). EBS was introduced in speech to allow reproduction of natural speech from a corpus of recorded spoken audio fragments [19]. Subsequently EBS was exploited in computer graphics to reuse and modify video sequences. Bregler et al. [3] introduced *video rewrite* to create a novel video of a person speaking by retrieving and concatenating mouth images from a training via using audio cues. Similarly, [12] presented an audio-driven visual-speech animation system which also parameterizes the mouth images, enabling generalization beyond captured video frames using a morphable model. Schödl *et al.*'s *video textures* [29] extended video EBS to be driven, for the first time by visual cues. They demonstrated the synthesis of perpetual videos by copying and re-arranging frames from a single source video. The video is modelled as a Markov process with each state corresponding to a single frame and the probabilities corresponding to the likelihood of transitions from one frame to another. These likelihoods are computed as frame-to-frame image similarities over a short temporal window.

In related work within the 3D graphics domain, Kovar et al. [22] construct a directed graph on 3D skeletal Motion Capture sequences, referred to as a *Motion Graphs*, where edges correspond to segments of motion and nodes identify connections between them. Motion segments include original motions and transition motions generated by blending segments together. Distances between pairs of frames are computed in pose space to determine if a transition is possible, decided using a fixed similarity threshold. Synthesis is performed by finding an optimal graph walk that satisfies user-defined constraints.

Our work for the first time combines both video textures, and Motion Graphs, driving path optimization over the latter using key framed poses identified using our SBR pose search. Uniquely, our choice of path is also constrained by user specified actions linking the sketched poses.

3. FEATURE EXTRACTION

We first outline how representations of human pose are extracted from the sketch query (Sec. 3.1) and video frames (Sec. 3.2). The representations are matched in Sec. 4.3.

3.1 Sketch Parsing

The system accepts a sequence of free-hand sketched strokes as query, captured via a web interface (see video, Sec. 6). We use a set of heuristics to label strokes to components of a stick-man, following the approach of Fonseca et al. [14]. Briefly, candidates for the torso stroke are prioritized via a voting system that combines measures of stroke intersection, center of mass, and similarity to a straight line. The head stroke is identified by the finding the stroke most similar to an ellipse. To label arms and legs the intersection point that unifies the arms or legs is identified. We enable the user to manipulate the left-right orientation of the stick-man (e.g. whether the figure faces toward or away from the page) as this information is absent in a sketch.

Having fitted a stick-man to the sketch, the joint angles of the articulated skeleton are converted into a pose descriptor $\mathcal{S} \in \Re^{20}$ as follows. Interpreting the skeleton as a hierarchy with torso at root, the angle θ_i that the i^{th} stroke makes with its parent is converted into a vector $\mathbf{v}_i$:

$$\mathbf{v}_i = \begin{bmatrix} cos(\theta_i + \epsilon) - cos(\theta_i) \\ sin(\theta_i + \epsilon) - sin(\theta_i) \end{bmatrix} \quad (1)$$

where ϵ is a small constant. The descriptor $\mathcal{S} = \{\mathbf{v}_0^T \mathbf{v}_1^T ... \mathbf{v}_9^T\}$ is formed from the nine joints in the skeleton ($i = \{1..9\}$), with $i = 0$ indicating the angle that the torso makes with the vertical. This construction removes the disjoint at $\theta_i = 0, 2\pi$.

3.2 Video Descriptor Extraction

We extract a pose descriptor for each video frame in an offline pre-process. Each descriptor is derived from a binary mask representing the shape (silhouette) of the dance performer in the frame. We opt to compute shape descriptors of this kind, rather than perform interest point detection, due to the noise and paucity of stable sparse features exhibited in our low fidelity archive footage.

3.2.1 Silhouette Extraction

Although performers appear distinct in low resolution footage, the presence of soft edges and changing intensity gradients from stage illumination precludes the use of simple heuristics to produce the silhouette (such as background subtraction) that may succeed on higher resolution footage. Thus our first step is to learn, for each video, an appearance model for texture that is likely to represent the performer. We apply a bank of Gaussian filters (Textons) to all frames in the video, and apply the standard Bag of Words model using a codebook size $k = 100$. Each pixel is thus assigned to one of k codewords, and a texton descriptor can be computed as the normalized histogram of codeword occurrence within a given window. We train a support vector machine (SVM) with positive and negative examples of performer texture using textons computed within 10×10 windows.

For training examples applying an adaptive threshold to an Itti and Koch image saliency field [15] is often sufficient to extract a good silhouette of a performer in simple, uncluttered conditions. We use this process to bootstrap our more robust texton silhouette extraction method; providing positive and negative exemplars.

Given the trained SVM, we extract the silhouette from a frame by predicting the probability of each pixel being foreground (performer) or background using its texton descriptor. We enhance the spatial coherence of this probability map by using a binary graph-cut [2], with the probabilities forming the unary (data) term and a standard edge preserving pair-wise term commonly used in image segmentation algorithms e. g. GrabCut [28]. Texture alone can be insufficient to discriminate the performer from background in overexposed footage. We therefore extend the unary term to incorporate the probability of the object being moving foreground, obtained by differencing neighbouring frames.

3.2.2 Descriptor Formation

Having computed the binary masks (silhouettes) of the performer, we form a pose descriptor using a simplified version of the Histogram of Oriented Gradients (HOG) descriptor [7] computed over the bounding box of the mask. A 2×2 grid (Fig 1) is centred upon the bounding box and a histogram computed independently within each cell, using 8 angular bins per cell and resulting in a 32 dimensional shape descriptor. Hereafter we refer to this space as $\mathcal{D} \in \Re^{32}$. The

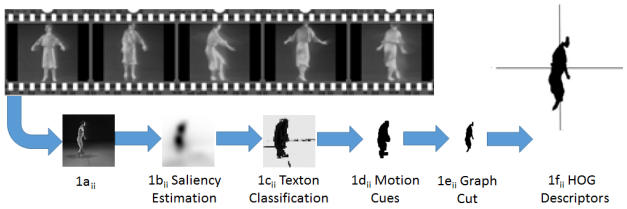

Figure 1: Video descriptor extraction. Source Image 1a$_i$, 1b$_{ii}$ Saliency Estimation, 1c$_{ii}$ Texton FG/BG Classification, 1d$_{ii}$ Incorporating motion cues, 1e$_{ii}$ Graph cut yielding silhouette. Resulting in a HOG descriptor computed per cell of a 2×2 grid

processing steps of subsection 3.2 are illustrated in Fig 1.

4. CHOREOGRAPHY BY SKETCH

We learn a non-parametric mapping between the query space ($\mathcal{S}$) and pose descriptor space ($\mathcal{D}$), using a set of around 230 manually marked up *training poses*. Valid poses lie upon manifolds in both spaces, each of which is sampled by the training process (Sec. 4.1). A graph strategy is used to compute similarity between a query and candidate video frame (pose) by approximating geodesic distance across these manifolds (Sec 4.2). This similarity score is used to rank each video frame in the database for relevance to a given query, underpinning both our pose retrieval (Sec 4.3) and video synthesis (Sec 4.4) applications.

4.1 Video manifold construction

The Quality Thresholding (QT) clustering algorithm[16] is used to identify the set of *training poses* from training footage by clustering data points (frames) within our descriptor space (Sec. 3.2.3). QT recursively prunes data points from the space exhibiting the greatest number of neighbours within a threshold distance, and suggesting these as cluster centres. In our experiments the result is a set of around 230 diverse poses (from $\sim$4500 frames) sampled from the performance 'Blueprint' ($c.1978$)[1].

The training pose descriptors lie upon a non-linear manifold of valid poses within $\mathcal{D} \in \Re^{32}$, which we model in piecewise linear fashion by building a graph ($\mathcal{G}$) in which training poses are nodes (denoted n_s) such that $\mathcal{G} = \{n_s\}$. Connectivity is defined via undirected edges, connecting each node to up to the N other closest nodes in the Euclidean neighbourhood (and falling within an upper distance threshold T). In practice we use $N = 10$. The weight between two training nodes $\omega(n_s \mapsto n_t)$ on each edge are proportional to the Euclidean distance between the nodes connected.

$$\omega(n_s \mapsto n_t) = \begin{cases} 1 - exp(|n_s - n_t|^2) & \text{if } |n_s - n_t| < T; \\ 0 & \text{else.} \end{cases} \quad (2)$$

where $|.|$ yields the Euclidean distance between training pose descriptors. Assessing the similarity of two video poses on the manifold is now a matter of computing shortest path between two nodes (see Sec 4.2).

In addition to building $\mathcal{G}$ with training pose, it is necessary to include all frames in all videos within our database to produce a useful retrieval system. Due to the noisy nature of the footage, invalid silhouettes may give rise to invalid pose descriptors off the manifold. It is undesirable to permit such data points to make larges changes in the topography of $\mathcal{G}$. We categorize frames as being either "confident" (n_c) or "unconfident" (n_u) by checking the covariance of their pose descriptors within a temporally local window in the video. Limited determinant of the covariance indicates a stable set of descriptors over time, which we assume implies a frame is n_c otherwise n_u. We expand the graph to $\mathcal{G} = \{n_s, n_c, n_u\}$ via the process outlined above, but limit N to 1 when admitting n_u to $\mathcal{G}$ as illustrated in Figure 2a. The geodesic distance across the manifold for any two poses in the dataset is now approximated by a shortest path computation over $\mathcal{G}$.

4.2 Learning Domain Transfer

We learn a mapping between $\mathcal{S}$ and $\mathcal{D}$ as a one-off process using the set of training poses identified in Sec 4.1. We manually annotate each pose with a sketch, from which the joint angles are obtained via Section 3.1. This yields a mapping $s \mapsto d \in \{S, D\}$ for each training pose. From this sparse mapping we are able to make a number of inferences at query-time facilitating sketch based pose retrieval.

First, for any provided query sketch ($q \in S$) we can compute the similarity between that sketch and any of the training sketches. The closest training sketch to q (denoted hereafter s) is identified, and the probability of similarity in space S modelled using a Gaussian distance function:

$$p(s|q) \propto exp - \frac{|q - s|^2}{2\sigma}. \quad (3)$$

Second, for any training sketch (e.g. s) we know $s \mapsto d$ and so know the corresponding node in $\mathcal{G}$ (denoted n_d). We may thus compute the shortest path across $\mathcal{G}$ to any other node i.e. video frame in our database. The product of the weights along the path between nodes is normalized similar to eqn 3, but where the distance between s and d is the geodesic distance. So the normalized probability of frame n_d and an arbitrary frame n_x being similar are:

$$p(n_x|n_d) = \prod_{\{a,b\} \in \mathcal{G}} 1 - p(n_a|n_b). \quad (4)$$

as a product where n_a and n_b are pairs of adjacent nodes on the shortest path. In practice the product of weights along the shortest path can be obtained using Dijkstra's algorithm over a set of log-weighted edges.

Combining equations 3 and 4 we compute the joint probability of any video frame in our database (n_x) being similar to our query (q) as:

$$p(n_x|q) = p(s|q)p(n_d|n_x). \quad (5)$$

where $p(n_d|n_x) \propto p(d)p(n_x|n_d)$ by Bayes, and we assume a uniform prior $p(d)$ over all training frames. This can be efficiently computed at query time as the shortest path calculations may be pre-computed across $\mathcal{G}$ offline. The process for computing $p(n_x|q)$ is illustrated in Fig. 2b.

4.3 Sketch based interaction

The proposed retrieval algorithm is integrated into a web based UI. Users are able to draw and then manipulate the articulated skeleton (q) to query the system. The results are displayed as in Fig. 3, as a grid of clustered results. Ranking all frames n_x in the database by $p(n_x|q)$ provides the user with the results. An enhanced results view enables the clustering of temporally local results (as adjacent frames exhibit similar scores), and so the user may readily identify

[1]Three Dances (Spink) ref ED/2010/10/5, Blueprint (Alston) ref ED/2010/4/6 from the Extemporary Dance Theatre Archive held at the National Resource Centre for Dance, University of Surrey ©

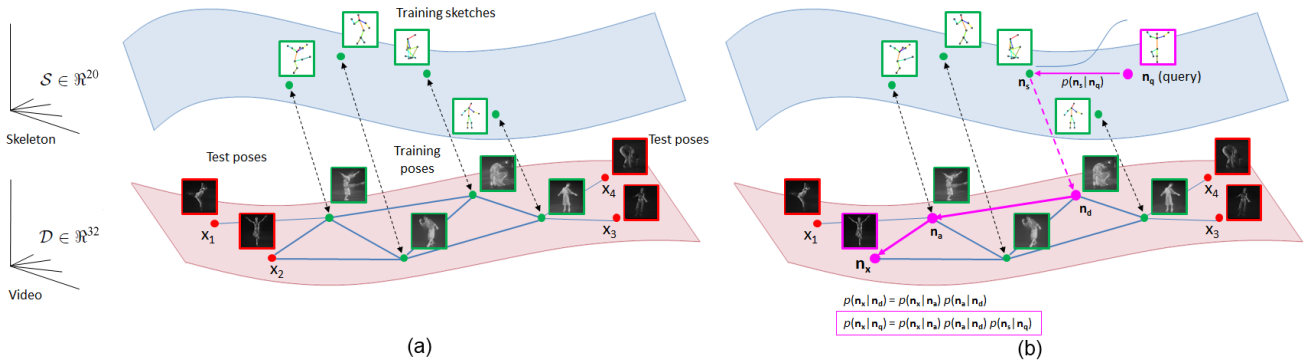

Figure 2: Manifold mapping underpinning our retrieval system. (a) Manifold construction. Training poses (green) are manually marked up creating sparse correspondence between $\mathcal{S}$ and $\mathcal{D}$. The search graph is constructed across training points in $\mathcal{D}$ to approximate the manifold. Additional points (red) are added to $\mathcal{D}$ from each video frame in dataset. 'Confident' frames (section 4.1) connect up to N training nodes (e.g. x_2), 'unconfident' frames (e.g. $x_{1,3,4}$) connect to the nearest training node. (b) Query processing. A query n_q is initially matched to find the closest training sketch n_s. The corresponding training pose n_d is used to compute geodesic distance (magenta) to each item in the dataset x_i.

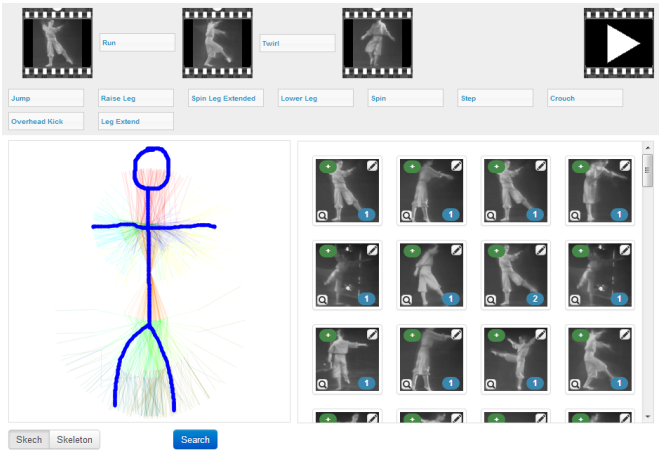

Figure 3: Pose retrieval and choreography creation interface. See accompanying video URL (Sec. 6) for demo and results.

temporally disjoint segments of the video that closely match the query pose of interest. With several poses defined in this manner, users are able to specify inter-pose actions, multiple desirable actions between poses may be combined (our representation for each connecting action is a probability distribution over all action classes, see Sec. 5).

Interestingly, the ability to map bidirectional between $\mathcal{S}$ and $\mathcal{D}$ enables us to transfer not only from skeletal pose to image (for retrieval) but also from image to skeletal pose. The ability to convert images to a set of joint angles allows query suggestion, where "shadows" of stick man poses within the database may be visualized beneath the users sketch as outlines to assist in the retrieval process. This produces an interactive interface reminiscent of the ShadowDraw clipart retrieval system[23].

5. SKETCH BASED CHOREOGRAPHY SYNTHESIS

The novel choreographic video sequence by sketching a sequence of pose keyframes, interspersed by desired actions (e.g. twirl, run). The novel sequence is generated by seamlessly concatenating fragments of an existing video, extending the Video Textures concept of Schödl et al [29].

5.1 Video Motion Graph generation

We construct a *motion graph* [22] by identifying transition frames; points in the video where temporally disjoint frame sequences may be seamlessly concatenated for playback. These transitions form nodes in the motion graph, with edges indicating frame sequences between transitions (Fig. 4). Random walks across the graph could generate novel sequences in perpetuity (as with [29]); however our system plans paths across the graph to produce user-guided output.

Transition points are identified by exhaustively comparing pose descriptors via eq. 4 from all pairs of frames of the video, and retaining those above a similarity threshold as transition candidates. This comparison is computed in a matrix, and smoothed using an isotropic filter to penalize local temporal incoherence in pose. As visual dissimilarity may be observed in video even in the presence of similar poses, optical flow[4] is used to calculate the visual dissimilarity of frames at candidate transitions by summing motion vectors between the frames and discarding candidates that exceed a threshold.

Frame-frame edge weights in the motion graph are computed via eq. 4, and sketch-frame weights via eq. 5. In addition we require frames to be labelled to indicate the likelihood of eleven different activities taking place local to that instant. Our activity set is: *twirl, spin, walk, run, leg raise, leg lower, leg extend, spin with leg extended, crouch, step* and *overhead kick*. Action labelling can be performed by any regular activity recognition algorithm (e.g. [31]) using our pose descriptors ($\mathcal{D}$) as a basis for activity classification — such labelling is a black-box process beyond the scope of this paper.

5.2 Video Path Optimisation

With a sketched storyboard defined by the user we identify a path across the motion graph. For each pair of keyed poses a 'virtual' source node is added into the motion graph for the start pose, with edge links to all nodes weighted by the similarity between the sketch and that transition frame (via eqn. 5). The successive (end) key pose is added as virtual sink node. This node also becomes the source node on a second copy of the motion graph, which serves this start pose and the subsequent end pose (Fig. 4 illustrates). Thus for k keyposes, $k-1$ copies of the motion graph and chained together by virtual source and sink nodes. A shortest path optimization is used to find the optimal route passing from

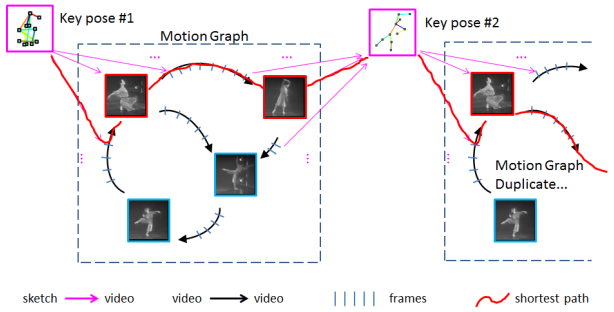

Figure 4: Video synthesis via motion graph. A directed graph is constructed from video fragments comprising sequential blocks of frames (blue marks on edges) linked seamlessly at transition frames (blue nodes). Sketched key poses (magenta nodes) are added as virtual nodes linking copies of the motion graph. The path with lowest cost (red), by eq. 6, from first to last key pose yields the new choreographic sequence.

the first to last virtual node (key pose), using Dijkstra's algorithm.

Path cost is evaluated as a function of pose similarity (C_{Target}), action constraints along the path (C_{Action}), and duration of the sequence (C_{Time}):

$$C = w_p C_{Target} + w_a C_{Action} + w_t C_{Time}. \quad (6)$$

with user-specified weightings $\{w_p, w_a, w_t\}$ for pose, action and time respectively, which are set as specified in Sec. 6.2 for results presented in this paper are configurable via the UI.

Pose similarity encoded by term C_{Target} is analoguous to edge weights accumulated in the classical shortest path algorithm. In the case of frame-frame moves within the motion graph (black edges in Fig. 4) similarity is determined by geodesic distance between the two frames across the manifold $\mathcal{D}$ (eq. 4). In the case of edges between the virtual nodes (start/end poses) and a frame in the motion graph (magenta edges in Fig. 4) the cost is determined by the joint probability (i.e. overall similarity function) of our pose retrieval algorithm (eq. 5).

To incorporate action constraints, each edge in the motion graph is augmented with a probability vector across action classes expressing the activity (run, twirl, etc.) detected local to that pair of frames. The proposed shortest path is segmented into $k-1$ linked stages; each being the portion of the path passing through a copy of the motion graph linked by the virtual source/sink nodes (i.e. between the $k-1$ pairs of key poses). The probability vectors for the set of frames in each linked stage $\{l_1 ... l_{k-1}\}$ are averaged independently yielding action probability vector $A(l_i)$. An ideal path would result in minimal total difference between $A(l_i)$ and the action distribution specified by the user for that linked stage $A(q_i)$ i.e. between the respective pair of key poses. The cost C_{Action} is therefore given by:

$$C_{Action} = \frac{1}{k-1} \sum_{l=1}^{k-1} |A(l_i) - A(q_i)|. \quad (7)$$

The temporal cost C_{Time} is derived from a count of the number of frames on the proposed path. The absolute difference betwen this and a target sequences length L (here we use 5 seconds per keypose pair) encourages appropriate transition times. Conceivably this parameter could be incorporated in the UI in future.

$$C_{Time} = S(\sum_{l=1}^{k-1} ||l_i| - L|). \quad (8)$$

where $S(x)$ is sigmoid function is used to normalise this final term.

$$\mathcal{S}(x) = x^2(3 - 2x). \quad (9)$$

5.3 Video Synthesis

The optimised path yields a frame sequence through the original video comprising the novel choreography. Although pose-coherent, any variability in the performer's appearance (e.g. illumination) and in stage location can complicate visually seamless stitching. Although spatial location might be incorporated as a constraint into the optimisation, in practice requiring adherence to original stages location places too many constraints on the original footage to permit novel choreographic sequences to be realised. We therefore opt for an 'infinite' stage, scrolling the stage against the motion of the performer's bounding box. This scrolling is smoothed using a low pass filter to avoid visual discontinuities. The performer is composited onto the stage utilising the previously obtained silhouette, and utilising Poisson blending [26] for gradient-aware compositing. Additional simple cross-fading is applied at the points of transition between video fragments to mitigate any remaining visual discontinuity in playback.

6. RESULTS AND DISCUSSION

We evaluate ReEnact using both low and high fidelity footage from the UK Digital Dance Archives (DDA) repository. For low fidelty footage we use archived performances of "Blueprint" and "ThreeD" from the 1970s Extemporary Dance collection[1], both digitized from cinefilm at 25fps, of duration 5:18 and 2:49 minutes respectively. Challenging features of this footage are its grainy, low contrast nature and heavy motion blur. We also demonstrate higher fidelity footage, "Autumn" from the NRCD [2] and a studio captured expressive contemporary dance performance "Expressive" similarly at 25fps, of duration 2:05 and 6:40 respectively. In the latter case footage is HD (1920×1080), otherwise footage is PAL (720×576). All videos were interlaced.

The pose retrieval, and the synthesis component of ReEnact are evaluated independently. Retrieval is evaluated using the learned manifold for Blueprint, generalised onto unseen videos (ThreeD, Autumn, Expressive) with performance quantified using Average Precision (AP). In addition, we demonstrate our inference process for retrieval can be run 'backward' yielding explicit pose estimation for any given frame. Choreography synthesis is evaluated qualitatively with examples of sketched storyboards and representative output (see also the accompanying video[3])

6.1 Sketch based Pose Retrieval

We evaluate the ability of the pose retrieval system to generalize to all (non-training) frames in Blueprint as well as to three unseen videos "ThreeD", "Autumn" and "Expressive". For each video six queries were drawn and AP computed over these for the top 1-80 results (Figure 6). Note that when evaluating on "Blueprint" we use only $\sim 97\%$ of the available frames (as 230 frames were manually marked up and used to train the system). We determine a result to be incorrect if, on visual inspection, more than one limb is

[2]Reconstruction of Autumn created for National Resource Centre for Dance, University of Surrey ©

[3]Supplementary video at
`http://www.dance-archives.ac.uk/media/12905`

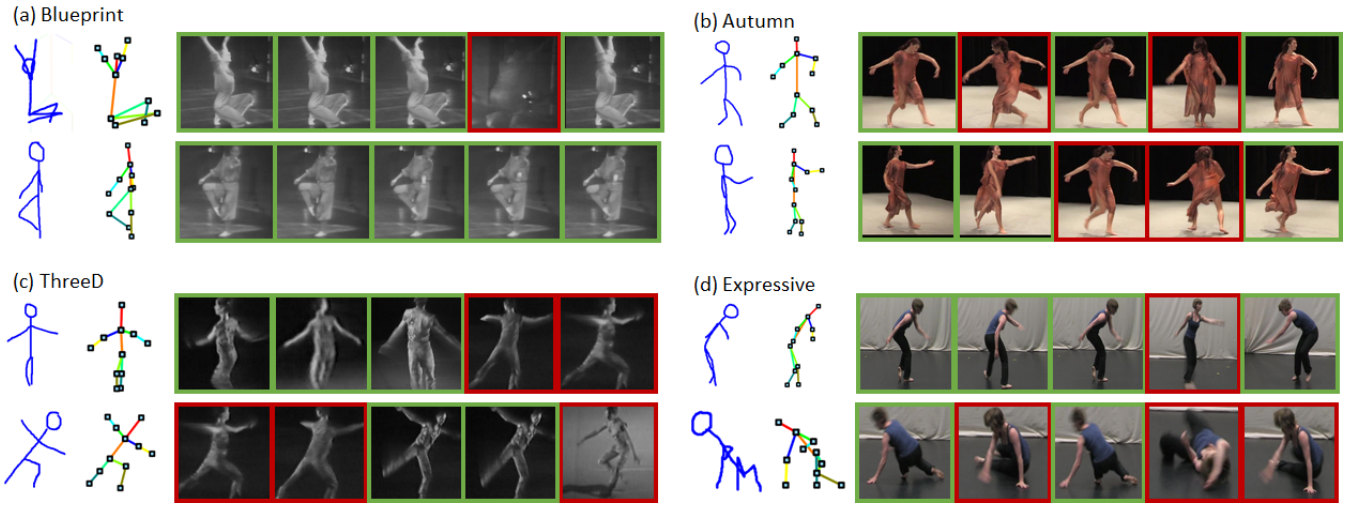

Figure 5: Top 5 results of the sketch(left most) / skeleton query(second left) for: (a) Blueprint(Train) video, (b) Autumn(test) video, (c) ThreeD(test) video, (d) Expressive(test) video

judged to be out of place by a few degrees. Although not explicitly handled by HOG descriptor we make this comparison insensitive to the left-right orientation of the figure.

Figure 5 illustrates a representative subset of queries for each video and their corresponding results. In all cases poses returned closely mirror those of the sketched query, and as expected the results from the (unseen frames) of training video Blueprint appear qualitatively superior to those from entirely unseen clips as test data is closer to the domain of the training data. Nevertheless the system has correctly transferred learning over Blueprint poses to enable correct retrieval of unseen poses from the three other video sources (including those with significantly different visual quality). The sensitivity to left-right orientation is seen in 5c second query where shape is very similar but are evaluated as incorrect.

A quantitative comparison of performance over the four clips is given in Figure 6. The MAP scores for Blueprint (60.0%), ThreeD(50.6%), Autumn(47.0%), Expressive (44.4%) indicate the system was able to generalize well from minimal training, without significant performance drop on content that differed from the training exemplars. The runtime performance of the system is real-time with queries taking ~10ms for several thousand frames using an unoptimised C++ implementation on a quad core 3Ghz PC.

Retrieval precision is influenced by the quality of mask extracted from the video; despite an elaborate silhouette extraction process being performed, limbs are susceptible to being removed by the algorithm especially in the more challenging lower fidelity footage that exhibits heavy blur and contrast bleaching. In cases where the algorithm failed to retrieve available poses, or returned unexpected results, visual checks identified that the silhouette masks were being incorrectly generated. We conclude that there is sufficient robustness and generality offered by our main contribution (manifold mapping over a HOG based descriptor), but that the initial performer extraction pre-processing could be robustified or potentially tailored to match individual content types.

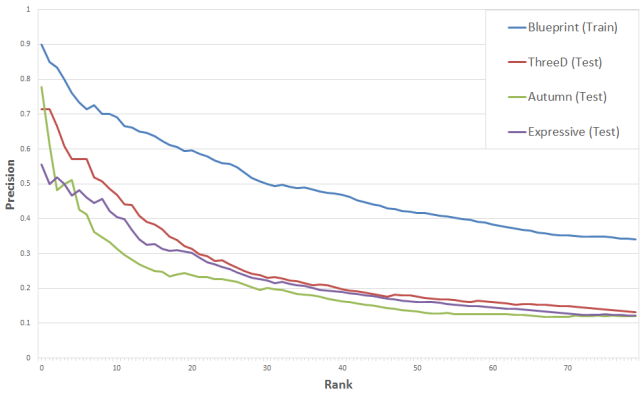

Figure 6: Plotting precision for Blueprint (training), ThreeD (test) Autumn (test), Expressive (Test) over the top 80 ranked results

6.2 Choreography Synthesis

We demonstrate the ability of the system to create new choreographic sequences from sketched keyframes, over both the "Blueprint" and "Expressive" videos. In Figure 8 we demonstrate a sample video created, from its raw sketch automatically converted to skeleton to the synthetic frames. We demonstrate both the full frame results of the key frames as well as actions as performer cut out images demonstrating the action betweeen. For the purpose of visualization, the distribution of action between keyframes in the new synthetic video is shown between keyframes. The example in Figure 8 and further example videos are included as supplementary video material (Sec. 6).

Choreographic generation through Poisson blending on the stage quality can suffer badly from a bad mask, therefore in these cases we avoid entering sections that has automatically been identified as low quality. This can complicate the synthesis, in both requiring an alternative source/target being defined that is handle implicitly by our virtual node approach to motion graph generation; also in the path to be completed may be forced to go off a more desirable route.

Although weights for video duration are user defined, generally video lengths for a story board composed of three sketches with interspersed actions are of length between 15-90s. In the case of the sample video in Figure 8 the new video is of 67s. The system is reactive but not particular sensitive to the weightings in the cost functions; for all the

results reported here we used the same weightings. Specifically, we up-weighted the action and pose requirements, to 0.6, 0.9 respectively and set the time weight low at 0.3 to avoid a short video being generated.

New sequences are typically generated in under 10 minutes, though the majority this runtime is spent on blending frames (e.g. Poisson blending) without which a sequence may be generated in under one minute.

6.3 Inference of the Skeleton

An interesting property of our inference framework is that it may be run *in reverse* to infer the joint angles (skeleton) of the performer given a video frame. Given an indexed video frame corresponding to any node n_x in $\mathcal{G}$, the similarity $p(n_i|n_x)$ to n_i, the i^{th} training frame in set n_t, is available via eq. 4. Given the marked-up pose $s(n_i) \in \mathcal{S}$ corresponding to n_i we can infer a skeleton i.e. vector of joint angles approximating n_x as:

$$s(n_x) = \frac{1}{|n_t|} \sum_{\forall n_i \in n_t} s(n_i) \mathcal{N}\left(1 - p(n_i|n_x), \sigma\right). \qquad (10)$$

where $\mathcal{N}$ is a Gaussian distribution with emperically set standard deviation σ. Fig. 7 (top) illustrates sample output for Blueprint.

Although explicit human pose estimation was not originally intended as contribution of this work, it is nevertheless interesting to observe that reasonable skeletons can be obtained from low fidelity footage where state of the art methods [13, 32] currently fail. We qualitatively demonstrate this property through visual comparison on video frames that fail under these algorithms. Fig. 7 contrasts the skeleton obtained from a single frame using the public implementations made by Ferrari et al. [13] and Yang et al [32] on their respective project web pages. As our approach doesn't directly infer the lengths of limbs we use a skeleton of user-specified size and set the joints as per the angles inferred by our process. Although this results in some alignment error, the pose generated is comparable and in some cases more closely mirrors the video content under our approach. Explicit pose estimation is not a goal of our work, and future work will more robustly investigate these initial observations on this additional property of our algorithm.

Figure 7 shows that we are able to infer resonable approximations of poses not only on our training video but on test videos too. We additionally demonstrate this through a small clip in the supplementary material in contrast to the more successful approach of Yang, over the footage.

7. CONCLUSION

We have presented two contributions: 1) a system for searching archival dance video for free-hand sketched poses, 2) a system for generating new dance sequences from the archive, keyframed by retrieved poses and intermediate actions.

Our retrieval system operates by learning a mapping between a query space (skeletal joint angle) and a visual descriptor space, using around a two hundred hand-labelled examples. Once learned this mapping is shown to generalize to unseen video with a averaged(over test videos) MAP of 0.46. This mapping is also demonstrated to perform skeleton inference comparative or better in more challenging cases than state-of-the-art techniques. Interestingly, performance on the higher fidelity test footage types was slightly lower than the more challenging footage ('ThreeD') showing that similarity of test footage to the training domain (230 frames from 'BluePrint') is more important than resolution or signal quality. This indicates that transfer learning approaches could be applied in future work when adding nodes from unseen (test) content to our graph search structure.

We demonstrated that the concatenative synthesis approach of Video Textures [29] can be extended to appearance and action-aware synthesis of new video choreography, via our novel path optimization approach. Further work to optimize silhouette extraction would improve both search and synthesis, as merging limbs in the mask is the main cause of ambiguity in the image descriptor. Additional cosmetic improvements could be made in the video blending used at transitions also. However we believe the most promising directions of the work are in the synthesis and search algorithms rather than the video pre- or post-processing steps.

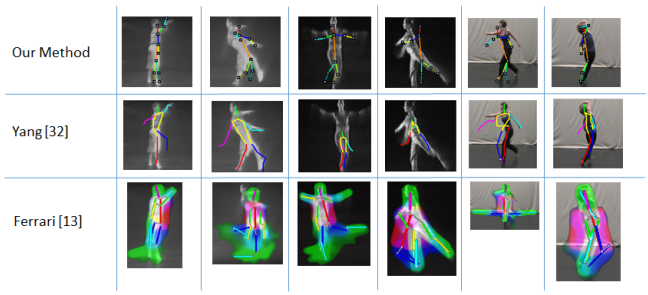

Figure 7: Comparison of Articulated skeleton estimation between our method, Yang [32] and Ferrari [13] over Blueprint two left most, ThreeD middle two and Expressive right two.

8. REFERENCES

[1] M. Andriluka, S. Roth, and B. Schiele. Pictoral structures revisited: People detection and articulated pose estimation. In *Proc. Computer Vision and Pattern Recognition*, 2009.

[2] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105 –112 vol.1, 2001.

[3] C. Bregler and M. Covell. Video rewrite: Driving visual speech with audio. *SIGGRAPH*, pages 353–360, 1997.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. pages 25–36. Springer, 2004.

[5] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang. Mindfinder: interactive sketch-based image search on millions of images. In *ACM Multimedia*, pages 1605–1608, 2010.

[6] J. Collomosse, G. Mcneill, and Y. Qian. Storyboard sketches for content based video retrieval. In *ICCV*, 2009.

[7] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *CVPR 2005*, 1:886–893, 2005.

[8] A. del Bimbo and P. Pala. Visual image retrieval by elastic matching of user sketches. 19(2):121–132, February 1997.

[9] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proc. British Machine Vision Conf. (BMVC)*, 2009.

[10] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *SBIM*, pages 29–36, 2009.

[11] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. In *IEEE TVCG*, volume 99, 2010.

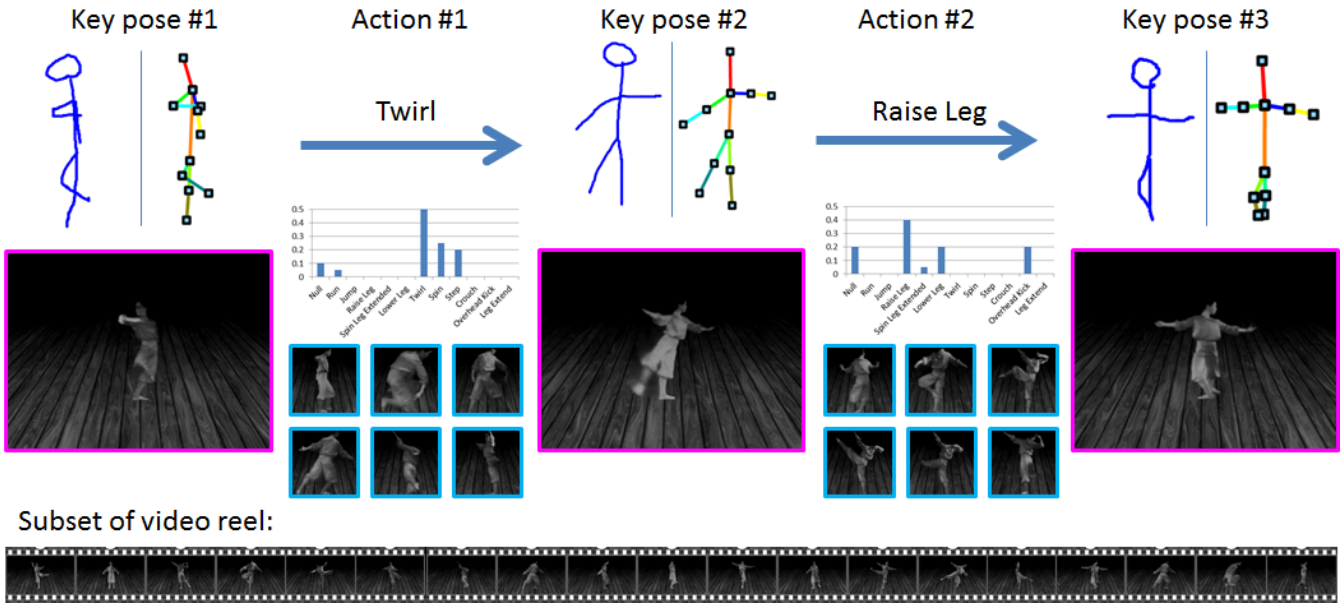

Figure 8: Top, Sample query storyboard composed of three key poses and two actions connecting. Bottom, Full frames selected from the provided key poses synthesised onto stage background, as well as a subset of frames between to demonstrate the related action. The overall distribution between keyposes is included as bar chart, with dominant bars being the target.

[12] T. Ezzat and G. Geiger. Trainable videorealistic speech animation. *ACM Transactions on Graphics (TOG)*, 21(3):388–398, July 2002.

[13] V. Ferrari, M. J. Marín-Jiménez, and A. Zisserman. Pose search: Retrieving people using their pose. In *CVPR*, 2009.

[14] M. J. Fonseca, S. James, and J. P. Collomosse. Skeletons from sketches of dancing poses. In *VL/HCC*, pages 247–248, 2012.

[15] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems 19*, pages 545–552. MIT Press, 2007.

[16] L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, 9:1106–15, 1999.

[17] R. Hu and J. Collomosse. Motion-sketch based video retrieval using a trellis levenshtein distance. In *Intl. Conf. on Pattern Recognition (ICPR)*, August 2010.

[18] R. Hu, S. James, and J. Collomosse. Annotated free-hand sketches for video retrieval using object semantics and motion. In *Proc. Multimedia Models*, 2012.

[19] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96)*, 1:373–376, 1996.

[20] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multi-resolution image querying. In *Proc. ACM SIGGRAPH*, pages 277–286, Aug. 1995.

[21] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar. Video retrieval by mimicking poses. In *ICMR*, page 34, 2012.

[22] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, July 2002.

[23] Y. J. Lee, C. L. Zitnick, and M. F. Cohen. Shadowdraw: real-time user guidance for freehand drawing. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 27:1–27:10, 2011.

[24] C. Liu, D. Wang, X. Liu, C. Wang, L. Zhang, and B. Zhang. Robust semantic sketch based specific image retrieval. In *Proc. Intl. Conf. and Multimedia Expo*, 2010.

[25] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:789–805, August 1992.

[26] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 313–318, New York, NY, USA, 2003. ACM.

[27] R. Ren and J. Collomosse. Visual sentences for pose retrieval over low-resolution cross-media dance collections. *IEEE Transactions on Multimedia*, Accepted 2012.

[28] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA, 2004. ACM.

[29] A. Schödl, R. Szeliski, D. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498, New York, New York, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[30] P. M. A. Sousa and M. J. Fonseca. Geometric matching for clip-art drawing retrieval. *J. Visual Communication and Image Representation*, 20(2):71–83, 2009.

[31] R. Vezanni, D. Baltieri, and R. Cucchiara. HMM based action recognition with projection histogram features. In *Proc. Intl. Conf. Pattern Recognition (ICPR)*, volume 6388, pages 290–297, 2010.

[32] Y. Yang and D. Ramanan. Articulated pose estimation using flexible mixtures of parts. In *Proc. Computer Vision and Pattern Recognition*, 2011.