# Visual Narratives: Free-hand Sketch for Visual Search and Navigation of Video

Stuart James

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey


Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, U.K.

March 2015

# Summary

Humans have an innate ability to communicate visually; the earliest forms of communication were cave drawings, and children can communicate visual descriptions of scenes through drawings well before they can write. Drawings and sketches offer an intuitive and efficient means for communicating visual concepts.

Today, society faces a deluge of digital visual content driven by a surge in the generation of video on social media and the online availability of video archives. Mobile devices are emerging as the dominant platform for consuming this content, with Cisco predicting that by 2018 over 80% of mobile traffic will be video. Sketch offers a familiar and expressive modality for interacting with video on the touch-screens commonly present on such devices.

This thesis contributes several new algorithms for searching and manipulating video using free-hand sketches. We propose the Visual Narrative (VN); a storyboarded sequence of one or more actions in the form of sketch that collectively describe an event. We show that VNs can be used to both efficiently search video repositories, and to synthesise video clips.

First, we describe a sketch based video retrieval (SBVR) system that fuses multiple modalities (shape, colour, semantics, and motion) in order to find relevant video clips. An efficient multi-modal video descriptor is proposed enabling the search of hundreds of videos in milliseconds. This contrasts with prior SBVR that lacks an efficient index representation, and take minutes or hours to search similar datasets. This contribution not only makes SBVR practical at interactive speeds, but also enables user-refinement of results through relevance feedback to resolve sketch ambiguity, including the relative priority of the different VN modalities.

Second, we present the first algorithm for sketch based pose retrieval. A pictographic representation (stick-men) is used to specify a desired human pose within the VN, and similar poses found within a video dataset. We use archival dance performance footage from the UK National Resource Centre for Dance (UK-NRCD), containing diverse examples of human pose. We investigate appropriate descriptors for sketch and video, and propose a novel manifold learning technique for mapping between the two descriptor spaces and so performing sketched pose retrieval. We show that domain adaptation can be applied to boost the performance of this system through a novel piece-wise feature-space warping technique.

Third, we present a graph representation for VNs comprising multiple actions. We focus on the extension of our pose retrieval system to a sequence of poses interspersed with actions (e.g. jump, twirl). We show that our graph representation can be used for multiple applications: 1) to retrieve sequences of video comprising multiple actions; 2) to navigate in pictorial form, the retrieved video sequences; 3) to synthesise new video sequences by retrieving and concatenating video fragments from archival footage.

**Key words:** Computer Vision, Information retrieval, Computer Graphics.

Email: s.james@surrey.ac.uk
WWW: http://stuartjames.bio/

# Acknowledgements

I would like to extend my thanks to the many people in my life from all over the world, who generously contributed to this thesis in time, effort or thoughts.

Firstly I would like to give a massive amount of thanks to my supervisor, John Collomosse. Throughout my PhD, John engaged with me in interesting and stimulating conversations of ideas and directions for my work. He, as well, continually encouraged collaborations across the wider academic community. Not many PhD students get the opportunities and support into the early hours, which John was willing to provide. Working with John has been a tremendous experience, which I hope will continue throughout my career. Similarly I would like to thank Krystian Mikolajczyk for being a supportive second supervisor throughout.

To the various bodies that made my PhD research possible, my thanks. Especially to the UK National Research Centre for Dance who provided the performance dance footage used widely within this thesis.

I would also like to thank Manuel Fonseca of Instituto Superior Técnico (University of Lisbon) for the collaborations during my PhD both in Surrey and Lisbon. My internships to Lisbon were a highlight in my PhD. Allowing me the opportunity to work with a fantastic group of people and learn more about life within a different research group and a different country. From INESC-ID, I would like to thank Daniel Gonalves, Joaquim Jorge and all the member of Visualization and Intelligent Multimodal Interfaces group for making me so welcome. For the stimulating conversations over coffee, after some fantastic lunches.

To all of Centre for Vision Speech and Signal Processing that have supported me continuously, my gratitude. Also a special thanks to those that I have become close collaborators with, while working with John: Rui, Tinghuai, Jongdae, Charles, Tu and Reede. The aforementioned have become not only collaborators but friends, in addition to Martin, Cemre, Evren and many more to name just a few from CVSSP. Also to my office mates over the years for putting up with my random desk tapping and other personality quirks, Zdenek, Muhammad, Aarushi, Fatemeh and Dalia.

My dearest friends collected throughout my life Adam, Luke, Mark, Matt, Roy and Will to name, but a few. Also my gratitude to more recent friends from Guildford Ibad, Marie and Yulia. All of which have been so kind to act as sounding boards as well as offering their support and encouragement.

Finally, I would like to thank my family for their continued love and support throughout my life. They have without question encouraged me in anything I wanted to do and always pushed me to strive to be as much as I can. To my dog Pixel, who I even managed to sneak into my thesis as a small tribute to him for watching me work till the early hours of the morning.

# Contents

# Glossary

**ANN** Approximate Nearest Neighbour

**AP** Average Precision

**BoVW** Bag of Visual Words

**CBIR** Content based Image Retrieval

**CBR** Content based Retrieval

**CBVR** Content based Video Retrieval

**CG** Computer Graphics

**CNN** Convilutional Neural Network

**CV** Computer Vision

**DA** Domain Adaptation

**DDA** Digital Dance Archive

**FLANN** Fast Library for Approximate Nearest Neighbour

**GF-HoG** Gradient Field - Histogram of Gradient

**GLOH** Gradient Location and Orientation Histogram

**GMM** Gaussian Mixture Model

**GPU** Graphics Processor Unit

**HD** High Definition

**HMM** Hidden Markov Model

**HoG** Histogram of Gradient

**HPE** Human Pose Estimation

**ICP** Iterative Closest Point

**IR** Information Retrieval

**KLT** Kanade Lucas Tomasi (Tracker)

**MAP** Mean Average Precision

**MCL** Multiple Classifier Learning

**MRF** Markov Random Field

**MSER** Maximally Stable Extremal Regions

**PCA** Principal Component Analysis

**PDF** Probability Distribution Field

**PHOW** Pyramid Histogram Of visual Words

**QBE** Query by Example

**QT** Quality Thresholding

**QVE** Query by Visual Example

**RANSAC** Random sample consensus

**RF** Relevance Feedback

**SBIR** Sketch based Image Retrieval

**SBR** Sketch based Retrieval

**SBVR** Sketch based Video Retrieval

**SBVS** Sketch based Video Synthesis

**SIFT** Scale Invariant Feature Transform

**SSD** Sum of Squared Difference

**STF** Semantic Texton Forests

**SVM** Support Vector Machine

**tf-idf** Term Frequency - Inverse Document Frequency

**UI** User Interface

**VLAD** Vector of Locally Aggregated Descriptors

**VN** Visual Narrative

# Chapter 1

# Introduction

The Internet is awash with digital imagery, and the rate of content generation is staggering. Every minute, 35 hours of video are uploaded to Youtube. Cisco systems predict that by 2018 video will grow to comprise 80% of Internet traffic, over two-thirds of which is user-generated and socially contributed [105]. These figures are only set to increase as consumer-publisher trends such as life-logging gain traction, and social media use penetrates developing nations.

Unfortunately technologies for the management of video collections have not kept pace with their generation. Despite the fundamentally visual nature of such content, search engines for visual media (e. g. Google, Bing, Yahoo) rely predominantly upon textual queries. Text-based search of visual media carries the disadvantage that imagery is not searched directly; text is matched against keywords present in a secondary meta-data stream e.g. user-generated keywords associated with the image. Furthermore, whilst text-based queries may efficiently convey semantic concepts (e.g. find me an image containing a car), they offer neither an intuitive nor concise vehicle for describing appearance (e.g. find me a car that looks like this, or a video containing movement like this). Pictorial queries can communicate such concepts with efficiency, and hybrid forms of query that combine the orthogonal strengths of pictorial depiction and semantic labels (e. g. encoding object type or action) offer the advantages of both; an intuitive vehicle for communicating queries in an expressive and efficient manner.

Recent years have also seen significant uptake in smart mobile and pervasive computing devices that feature touch-screen gestures rather than a keyboard as their main input modality. Such devices are now replacing the desktop as the principal way many users interact with an increasingly video-based Internet.

These parallel trends — the explosion of digital video online, and the emerging dominance of devices driven by gestural interfaces — motivate new techniques for managing and manipulating the wealth of video data available to these platforms.

## 1.1 Contributions of this Thesis

This thesis contributes novel algorithms for interactively searching and manipulating digital video using free-hand sketches, for example drawn using a tablet touch-screen.

Sketch is a rich and flexible communication medium, with an ability to concisely communicate multi-modal information that makes it well suited to the description of video. Sketches can naturally convey the structure and colour of objects. Sketched trajectories, or sequences of sketched frames, can communicate the desired dynamics of a video. Text labels and/or coded inks can be used to indicate semantic properties in the scene, such as object or action type, or can demark foreground objects from structure in the background. However with this expressive power come several drawbacks. Sketch is highly unconstrained and the casual throw-away act of producing a sketch to communicate intent (e.g. to describe a search query or to storyboard a sequence of actions) promotes the generation of inaccurate and hastily drawn sketches. The resulting inaccuracy and ambiguity, coupled in many cases with limited depictive skill on the part of the user, makes the interpretation and matching of sketches a challenging pattern recognition problem.

This thesis contributes several new algorithms to tackle this matching problem, referred to as the *Sketch based Video Retrieval (SBVR)* problem. Although Sketch based Image Retrieval (SBIR) has received considerable attention over the years, SBVR remains an under-researched arm of the broader "query by example" (QBE) or "visual search" problem. We outline our contributions to SBVR in subsec. 1.1.2.

This thesis also introduces new techniques for *Sketch based Video Synthesis (SBVS)*; synthesising a novel video sequence using sketches to specify the desired video content. Given their importance in the depiction of video events, our work focuses on people and specifically the domain of dance performance footage, which exhibits rich variation in shape (human pose) and movement. We show that sketched depictions of performers accompanied by descriptions of their movements, can be used as to direct the construction of novel choreographic sequences. We outline our contributions to sketch-driven video synthesis in subsec. 1.1.3.

Both categories of contribution in this thesis — SBVR and SBVS — are underpinned by the concept of the user specifying their intention through a sketched *visual narrative*.

### 1.1.1 Visual Narratives

Visual narratives (VNs) are pictorial representations of events, and feature extensively in our everyday lives even from an early age. Comic books are examples of VNs; sequences of salient visual snapshots that collectively describe sequences of actions. In the Creative Industries,

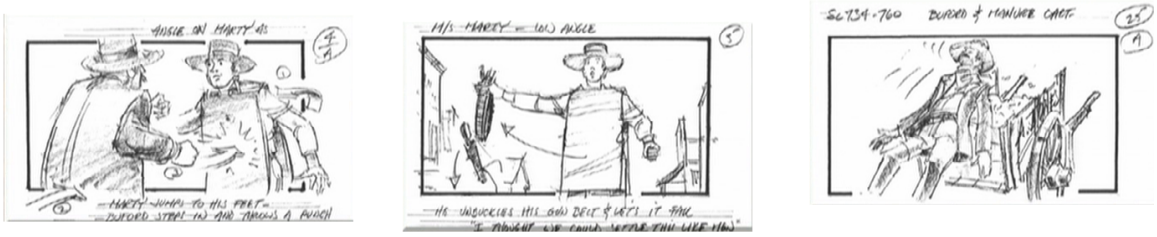


**Figure 1.1** Example of a Visual Narrative (VN) used during production storyboarding of the movie 'Back to the Future 3'. VNs intuitively and concisely describe the salient actions within an event, capturing not only semantics but also appearance and dynamics. This thesis proposes VNs as an novel interface modality for searching manipulating videos of events.

use of VNs is commonplace with sketched storyboards being used by screen writers to specify scene content and action in the earliest stages of film or animation production.

In its simplest form, a VN is a single image depicting an *action*. An action comprises one or more actors (e.g. people or objects) and their movement. More commonly a VN comprises a sequence of such images depicting a sequence of actions, which we term an *event* (Fig. 1.1).

Our work enables users to represent desired video content via sketched VNs — either for the purposes of specifying a search query (SBVR) to search for an event, or to drive video synthesis (SBVS) to construct an account of an event. To the best of our knowledge, VNs have not previously been used for SBVS and have only briefly been explored for SBVR.

### 1.1.2 Sketched Visual Narratives for Video Retrieval

This thesis contributes several new algorithms for hybrid SBVR using VNs as queries. We first present a novel high-performance algorithm for hybrid SBVR capable of searching several hundred videos in less than one second, whilst maintaining comparable accuracy to the state of the art. We then show how that accuracy can be improved beyond the state of the art using relevance feedback. A novel algorithm for VN based on sequences of sketches is then developed within the domain of Dance sequence retrieval.

These contributions serve to explore the following research hypotheses:

**[H1] Hybrid SBVR can scale to high performance large-scale video search using an efficient index representation.**

The *performance* of an information retrieval system may be interpreted in terms of *speed* i. e. time taken to retrieve results at query-time, and the *accuracy* of the retrieved results. A high performance system should exhibit both a high accuracy and a high query speed. Previous

SBVR systems have focused on accuracy over speed, often taking several minutes to search a dataset of a couple of hundred videos [53, 101, 102]. This is impractical for most use scenarios that demand interactive, i. e. sub-second retrieval times as well as high accuracy.

The poor speed of existing systems is due to their adoption of 'model-fitting' retrieval strategies. The sketch is treated as a model that is fitted to each video in the dataset via a computationally expensive optimisation procedure; the closeness of the fit is used to rank videos for relevance. The fitting is typically applied independently to each video, yielding at best $O(n)$ i. e. linear complexity of the search with dataset size. By contrast, most other visual search systems adopt an indexing strategy to retrieval. A digital fingerprint (feature vector) is distilled from each video during ingestion, and from the query at query-time. The features are then matched in a high-dimensional space using an efficient approximate nearest-neighbour (ANN) search structure such as a $k$D-Tree. Such structures can yield sub-linear e. g. $O(\log n)$ search complexity.

We hypothesise that hybrid SBVR could be achieved via an indexing, rather than model fitting, scheme without loss of accuracy through a novel multi-modal video descriptor. Moreover such an approach should be considerably faster at query-time, enabling interactive speeds. This hypothesis is tested through the development of such a descriptor and the demonstration of a high performance video retrieval system incorporating it.

#### [H2] SBVR accuracy can be improved using Relevance Feedback (RF)

Visual search systems must match in the presence of ambiguity in the query. A common technique to mitigate ambiguity is to iteratively present results to the user, asking them to indicate positive and negative examples within the returned results. These indications are fed back into subsequent iterations, forming a collaborative loop with the user to produce increasingly relevant results — a process referred to as Relevance Feedback (RF).

Although RF is commonplace in QVE and has been briefly explored for SBIR, it has never been explored for SBVR primarily due to the challenges of implementing SBVR at interactive speeds. Assuming hybrid SBVR may be implemented at interactive speeds (via **H1**) it should be possible to harness RF to enhance accuracy. We test this hypothesis through the development of RF within the novel video retrieval system developed to test **H1**.

#### [H3] Video event retrieval may be facilitated effectively using VNs

Although sketches of actions have been briefly explored for the retrieval of video events [53, 101, 102], those studies were constrained to only single sketch of an action (a so called 'storyboard sketch'). However one may more generally consider an event to be comprised

of a sequence of salient actions. We hypothesise that a visual representation of this actions sequence (the VN), coupled with a novel algorithm to match such representations to video, would provide an effective tool for video event retrieval.

We test this hypothesis through the development of the first VN based video event retrieval system using multiple sketched instants. This algorithm is developed within the constrained domain of Dance retrieval, using a sequence of sketched human poses interspersed with semantic action labels (e.g. jump, twirl). In conjunction with the contributions of subsection 1.1.3 we develop a new unified algorithm for both searching and synthesising video using such VNs.

#### [H4] Human Pose can be depicted within a VN to enable performance search

A significant proportion of video encountered 'in the wild' contains people, and often focuses upon their performances or actions. A useful SBVR system should therefore address the matching of video and sketches containing people. It has been shown elsewhere [52] that people are often depicted in a stylised pictographic form (commonly as stick figures) with their shape (pose) encoding within the limb positions of that figure. This presents a broader semantic gap versus other object types that tend to be depicted using approximations of their shape. Pose search is generally under-researched and works well only in constrained conditions [74, 109]. We test this hypothesis through development of a novel pose search algorithm operating over challenging archival dance performance footage exhibiting diverse ranges of human pose, and using sketched poses as queries. To the best of our knowledge sketch based pose retrieval has not been previously explored in SBVR.

### 1.1.3 Sketched Visual Narratives for Video Synthesis

The volume of video content available in digital archives, combined with advances in pattern matching techniques, raises new possibilities for video content re-use. *Concatenative synthesis* is the process of re-purposing existing captured data (e. g. video [181], skeletal motion capture [123] or 3D mesh capture [104]) to create new sequences of movement. Fragments of captured data are cut from the existing captured data and joined together seamlessly in a novel order to create the desired sequence. Care is taken when selecting fragments so that they appear locally seamless when joined, whilst still adhering to the globally desired pattern of movement.

#### [H5] Sketch based concatenative synthesis may be facilitated using VN

In this thesis we explore this hypothesis through the contribution of a new technique for designing video sequences using sketches. Specifically we focus on the domain of Dance

footage, and enable users to interactively choreograph new sequences using a sequence of sketches (VN) interspersed by action labels. The result is a flowing piece of novel choreography performed in the spirit of an existing piece of video footage. The basic choreographic phrases, the performer, and costume remain constant but can be manipulated to create a novel dance sequence. We refer to this system as "Re-Enact"'

We explore two complementary use modes for Re-Enact. First, a fully automatic approach that generates synthetic videos using only the VN and a set of user-weighted parameters as input. Second, a semi-automatic (interactive) approach in which a RF-like interface is used to enable the user to select appropriate dance phrases for composition given an initial sketched (VN) query.

## 1.2 Structure of the Thesis

We outline the chapter structure of the remainder of this thesis, summarising the principal contributions of each:

### Chapter 2 — Literature Review

A comprehensive literature survey of fields related to this research including: Sketch based Retrieval; Relevance Feedback; Domain Adaptation; Pose Retrieval and Estimation; and Concatenative Synthesis.

### Chapter 3 — Multi-modal Video Indexing for Sketch based Retrieval

We present a high performance hybrid SBVR algorithm based on a novel spatio-temporal descriptor encoding multi-modal video attributes. An efficient search index is constructed in this feature space enabling search with sub-linear time complexity. Furthermore we demonstrate how the interactive speeds yielded by this approach may be harnessed to incorporate relevance feedback for iterative refinement of results, exceeding the performance of the state of the art.

### Chapter 4 — Sketch based Pose Retrieval and Estimation

We present the first sketch based pose search system, capable of searching for a human pose in dance footage using a single sketch. An exemplar guided mapping is established between the query space (comprising sketched stick figure joint angles) and a HOG based video descriptor space. We explore domain adaptation techniques to scale this approach across many videos without requiring excessive training.

### Chapter 5 — ReEnact: Graph Representation for Search and Synthesis

We first demonstrate a system for retrieving a sequence from a video using a visual narrative sketch. A graph-based optimisation is performed to identify the best sequence of video frames for a given narrative. We demonstrate that the same graph optimisation framework may be used to synthesise novel sequences from a single archival video, using concatenative synthesis.

### Chapter 6 — Conclusion

We reflect on the contributions of this thesis in light of the research hypotheses (H1-4) outlined in this chapter.

# Chapter 2

# Literature Review

In this chapter we present a comprehensive overview of content based image and video retrieval (CBR) techniques. We focus on sketch based retrieval (SBR) for both forms of video, and additionally survey literature on semantic segmentation, pose estimation, transfer learning and action recognition relevant to the technical contributions of this thesis.

## 2.1 Introduction

The contributions of this thesis are within the fields of Information Retrieval (IR), Computer Vision (CV) and Computer Graphics (CG) – in particular to the sub-fields of CBR and SBR.

This literature review therefore primarily focuses upon CBR and SBR. We outline the state of the art for CBR (Sec. 2.2), focusing upon SBR to create a taxonomy of existing SBR techniques (Sec. 2.3).

Sec. 2.5 explores the CV task of Human Pose Estimation (HPE) and pose search algorithms, which are closely related to the sketch based pose retrieval study in Chapter 4. Image classification (Sec. 2.4) and transfer learning (Sec. 2.7) techniques from the CV literature are becoming increasingly common in CBR and SBR and therefore we address these topics, which are of relevance to Chapters 3 and 4. Finally we give an overview of action recognition (Sec 2.8) and concatenative synthesis approaches to animation (e. g. motion graphs, subsec. 2.6.1) that are applicable to Chapter 5.

## 2.2 Content based Retrieval Overview

The proliferation of user generated content brings with it new challenges for retrieving and presenting relevant media. Content has been traditionally indexed using semantic labels

in the form of metadata 'tags', however the creation of such tags requires extensive user annotation. Although this can be achieved through crowd-sourcing (e. g. Mechanical Turk), scaling up via such approaches introduces subjective variability in the choice of tag assigned. Automated approaches to the tagging problem are one solution, and are explored in Sec. 2.4.

By contrast, CBR approaches aim to extract meaningful information directly from visual content. Digital fingerprints that describe the content in an expressive yet concise manner are extracted and used to represent content. Most commonly such representations take the form of a high-dimensional feature or 'descriptor', due to the ability to leverage efficient data structures to create a search index from such descriptors.

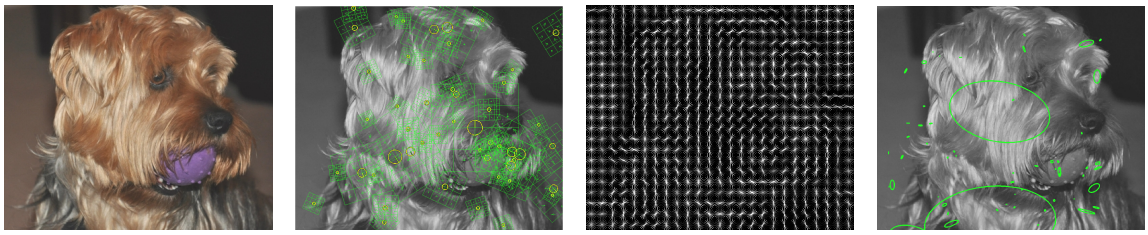
A common form of CBR is Query by Visual Example (QVE), wherein an image or video is provided as a query and the CBR system searches for visually similar content. QVE systems typically pre-process each content item in an offline step to form a database of descriptors. When presented with a similarly formed descriptor at query-time, matching is performed by comparing the descriptor of the query to those in the database. This can be performed linearly as a simple one-vs-all comparison, or efficiently in sub-linear time using data structures such as an inverse (associative) index or *kd*-tree, if the descriptor is of a form amenable to use in such structures. This is highly desirable since modern databases of visual content may contain millions of records, which may be intractable to query at interactive speeds if a straightforward linear search is used.

In this section we outline the state of the art in CBR using QVE. We build on recent survey papers [61, 197, 144, 108, 222] for image search in subsec. 2.2.1, and surveys [236, 92, 33, 156, 11] for video in subsec. 2.2.2.

### 2.2.1 Image Retrieval

Early image retrieval approaches focused on computationally efficient global representations. Colour histogram [203] is a well known early example, using quantised binning of image pixel colours to represent content; extensions included gridding [51] the image to provide a spatial representation. Additional representations include Colour Moments [207] and Colour Sets [203, 204] that overcome the quantisation problems of colour histograms.

An alternative modality is image texture – visual patterns that have properties of homogeneity that do not result from the presence of only a single colour. Texture has proven to be a fruitful research direction for image retrieval. From [89] statistics identified contrast, inverse difference moment and entropy having the greatest discriminatory effect in regards to texture. Tamura et al. [213] explored the human vision perspective of texture, identifying coarseness, contrast, directionality, line likeness, regularity and roughness as important



(a) (b) (c) (d)

**Figure 2.1** Examples of contemporary image descriptors: (a) Example image (b) Visualisation of SIFT [135] (c) Visualisation of HoG [58] (d) Visualisation of GLOH [141].

properties. Tamura et al. [213] influenced many subsequent SBR systems [40, 53, 102] in terms of which properties to represent. Extending beyond small datasets [41, 202] explored large-scale texture classification.

Another way to define a global property of an image can be to consider the shape of an object within the image. Most common global shape descriptors are based on and Moments and Fourier descriptors. With origins in Central moments, Hu et al. [98] identified seven moments that exhibit affine invariance. Moment based approaches describe the shape of a region. In contrast, Fourier descriptors – describe the contour of the shape, with good robustness to noise and also exhibit affine invariance. Shape has been considered extensively within the Sketch based Image Retrieval (SBIR) literature, and these approaches are explored in greater detail within the context of SBIR in subsec. 2.3.1.

Most global representations are sensitive to affine transformation of both the image and the objects within it. As an alternative, sparse representations can be used. Sparse representations utilise feature points to describe local information. Such points can be identified via Harris Corner Detector [91], Difference of Gaussians (DoG) [135],[17] or dense sampling [44, 229]. Detected points are then described by local pattern information. Popular descriptors include the Scale Invariant Feature Transformation (SIFT) [135], GLOH (Gradient Location and Orientation Histogram) [141] as well as [58, 34, 178] encode commonly local patch gradients to the point. The gradient approaches do not encode colour information, therefore extensions of approaches are used such as Opponent SIFT (OpSIFT, sometimes referred to as Colour SIFT) [221] or PHOW [25].

Each image is represented by a set of descriptors which can be matched via RANSAC [76] or exhaustive search. However it is more common to encode them via codebooking methods. An example of a popular method is Bag of Visual Words(BoVW) [199], adapted from IR BoW representation [239]; its simplest form follows the pipeline in fig. 2.2. In the BoVW pipeline, keypoints within the image are detected, and descriptors collected local to these points from all images in the dataset. The feature space is then quantised e. g. using K-Means or GMM,

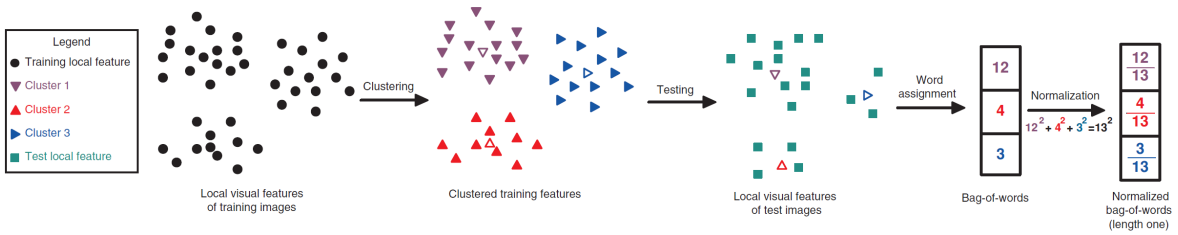



**Figure 2.2** Framework for Bag of Visual Words image encoding (Figure of Ji et al. [112]), using sparse features extracted from a pool of images to train a codebook via clustering the descriptors into K words. Each images' descriptors are then assigned to their nearest cluster, frequency counts of the number of descriptors assigned to the clusters form a image descriptor. Commonly normalised by L-1, L-2 or TF-IDF.

to form a vocabulary of codewords. Image descriptors are assigned to a codeword via nearest-neighbour (hard) assignment and a frequency histogram of codeword occurrence within each image is used as a global image descriptor. Commonly the frequency histogram is then normalised through L-1 (Manhattan) or L-2 (Euclidean) norm. As an alternative to L-norm, Term Frequency - Inverse Document Frequency (TF-IDF) [117] from IR is commonly applied to incorporate knowledge of the database encouraging more meaningful, less frequent words. TF-IDF weights the frequency of terms by their sparsity within the database.

Extending the BoVW method, Lazebnik et al. proposed Spatial Pyramid Matching [128], based on gridding the image and quantising into individual histograms per cell. Alternatively, Scale Pyramids Kernel (SPK) [26] combining spatial gridding with a multi-scaled version of the image. Much research has focused on determining the optimal configuration of methods and parameters for BoVW, most commonly for Image classification due to high-profile international benchmarks such as PASCAL [44]. Alternative feature-space quantisation methods have also been explored with variants of K-Means (Hierarchical K-Means [60] and Approximate K-Means [164]) to overcome boundary issues, where two points on either side of a boundary get assigned to different bins while being spatially similar. Philbin et al. overcomes this issue by assigning descriptors to multiple bins [165]. Hamming Encoding [110] complements BoVW by encoding the approximate location within its Voronoi cell in a binarised representation. Zhang et al. [241] refactored BoVW to use Local Sensitive Hashing (LSH). VLAD [111] produced compressed binary image representations fitting into only tens of bytes.

More recently approaches from Image Classification have been adapted for CBR, Perronnin et al. [162] adapted Fisher Vectors by compressing the typically high dimensional (yet sparse) Fisher vectors to make them more amenable to distance computations underpinning CBR. Sparse Coding [81] has also been adapted for CBR.

With recent hardware improvements (e. g. GPU) it has been made possible to learn image

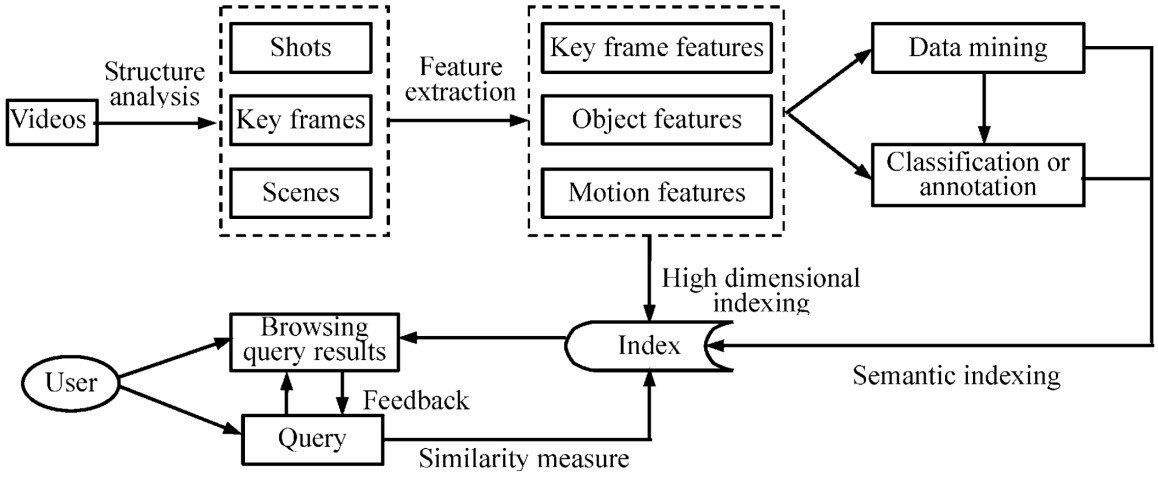



**Figure 2.3** Generic framework for Content based Video Retrieval (Figure from Huet al. [103]).

descriptors directly from datasets as opposed to prescribing them. Convolutional Neural Networks [124] learn from a large pool of training data an image encoding of convolutional operations that are discriminatory to the collection. Through a set of layers of convolutional and pooling layers within a Neural Network descriptions are learnt in addition to an a Image classifier. To use convolutional neural networks within alternative (e. g. non-classification based tasks) it is possible to extract an representative global image feature, from the fully connected layers. By doing so, it is possible to use the descriptor for retrieval as opposed to classification. Wan et al. [227] explored this application for CBR. Alternatives to convolutional neural networks include autoregressive neural networks which are an unsupervised method (i. e. do not require category labelled data for training the features).

### 2.2.2 Video Retrieval

Videos are an extension of images into the temporal domain, and this temporal aspect allows for richer representations that can be exploited for greater robustness or incorporate motion information. Yet for performance and convenience of implementation many recent CBR approaches work at a key-frame level learning a set of attributes related to set of sparsely sample video frames (i. e. an image-based problem). Spatial-temporal approaches are more commonly used for action recognition (sec. 2.8), trajectory matching, sports analysis or surveillance. A useful reference framework for generalised Video Retrieval was defined by Ji et al. [112] (Fig. 2.3). We now discuss the different components of this framework.

Initially videos are parsed into atomic units – shots, key frames or scenes. Shots are breaks in the video selected by the editor, these are commonly detected using feature extraction [96, 242, 233], similarity measures [42, 35, 55] or detection [122, 38, 46, 93, 24]. A full survey of techniques can be found in [103, 200, 237]. Key frames are frames in the video that

are representative of the shot's content [212, 19, 148]. Such frames contain as much salient information as possible while having minimal redundancy. Scene segmentation is also known as story unit segmentation [103]; scenes are higher level semantics than shots, providing a context. Extraction of such segments are more challenging often involving multiple modalities within the video such as image, text, audio [211, 47, 214].

Extracted atomic units are then described through feature extraction. Common approaches leveraged from Image Retrieval SIFT, HoG or their respective temporal extensions Spatio-Temporal SIFT, 3D HOG. Alternative to the edge approaches, motion approaches rely on optical flow, such as Histogram of Flow (HoF) [58]. Sparse interest points are then encoded through image retrieval techniques BoVW. Utilising the underlying video stream (often MPEG4/7), of $I$ (Full frames) and $P$ (partial frames, between I frames) frames, for extraction of colour or texture features. Amir et al. [4] utilised such an approach, combining these motion features with static key frames to improve overall retrieval performance.

Spatio-temporal volumes are popular in Action Recognition (Sec. 2.8), but have had little success within IR, primarily due to the problems of matching parts of the volume to the query. Basharat et al. [16] explored spatio-temporal volumes of SIFT correspondences. Correspondences are clustered to produce segments these segments are merged and split, to produce a track across the video for objects. A bipartite graph is then used to match videos, which renders the process expensive.

## 2.3 Sketch based Retrieval

Sketch based Retrieval (SBR) methods originated in the mid-nineties with a focus on image retrieval (**SBIR**) exploring how simple visual depictions could be located within image databases. Early approaches such as QBIC [12] required significant amounts of manual interaction for both the indexing process and the retrieval. These approaches were improved upon to become more robust and automated via exploring: elastic matching of contours to images [20], line orientation [39] information, and adopting from the Information Retrieval field the BoVW [99] technique, allowing for large scale retrieval.

The late nineties saw the seeds of Sketch based Video Retrieval (**SBVR**), starting with early systems such as VideoQ [40]. Methods not specifically aimed at sketch [97, 208], described the trajectories of objects without considering appearance facets. More recently there has been increase in the number of techniques, beginning with Collomosse et al. [53], proposing the concept of sketched storyboards as a query modality. These storyboards may be considered a simplistic form of visual narrative since they contain only one sketch, and with quite limited attributes (e. g. motion direction but no trajectories). This early work was followed up in

Hu et al. [101, 102] incorporating the semantic class of the sketched object as an additional facet of information. Such systems fusing both appearance and semantic information have been referred to as 'hybrid' SBR systems [101, 102].

Earlier techniques for both SBIR and SBVR relied heavily on **optimisation** as a search tool. The query is treated as a parametrised model of some form, and fitted to each database item with the closeness of fit providing a criterion for ranking that item for relevance. Alternatively **indexing** based methods distil a descriptor (or set of descriptors) from each database item. We propose a taxonomy for SBR in fig. 2.4 where we define a set of key traits common among the SBR field, outlining the different facets or methods used within different techniques. These attributes are explained more fully throughout this section, but in summary they are: *contour or line* – approaches that are based on the extraction of the outline of the object, or through edge detection; *Blob* – describing the overall shape of a blob, e. g. through statistical moments or shape factors such as aspect ratio, skewness, etc.; *interest point* as in IR using sparse points of interest can provide a robust retrieval method with key (interest) points often detected along edge or corner artefacts in an image; *grid* – spatially gridding the image and describing content within cells; *relevance feedback* – after providing the user an initial set of results, allowing them to guide the system through positive and negative examples; *semantics* – the inclusion of automatically detected classes (e. g. person, horse) within the query; *motion* – an attribute related to video, where object trajectories are considered; *colour* – describing the colour of objects as well as shape; *texture* – or patterns within the object of heterogeneous colour (or shades); *background* – more suited to SBVR, background details such as grass, within SBIR the background (or part of) is more commonly one of the objects to match against; *non-photo* – e. g. cartoon, or other sketches, loosely relevant to this thesis, but considered here in terms of RF that is approached in Chapter 3 and 5.

This section now divides into two subsections focusing on SBIR (subsec. 2.3.1) and SBVR (subsec. 2.3.2) respectively, with reference to the taxonomy within fig. 2.4. We draw upon literature survey papers for SBIR [159, 1, 158] in 2.3.1. To date there has not been a comprehensive review of SBVR, therefore we present one here.

### 2.3.1 Sketch based Image Retrieval

Sketch based Image Retrieval (SBIR) techniques may be categorised into two forms of approach: blob based and contour based. Blob based approaches commonly describe the objects within the image based on facets of information such as shape, colour, texture. This is in contrast to contour based techniques, that describe the image based in terms of the structure of a set of lines or curves. For example, elastic matching [12, 107, 201] between object outline or saliency information and features detected on the contours [99].

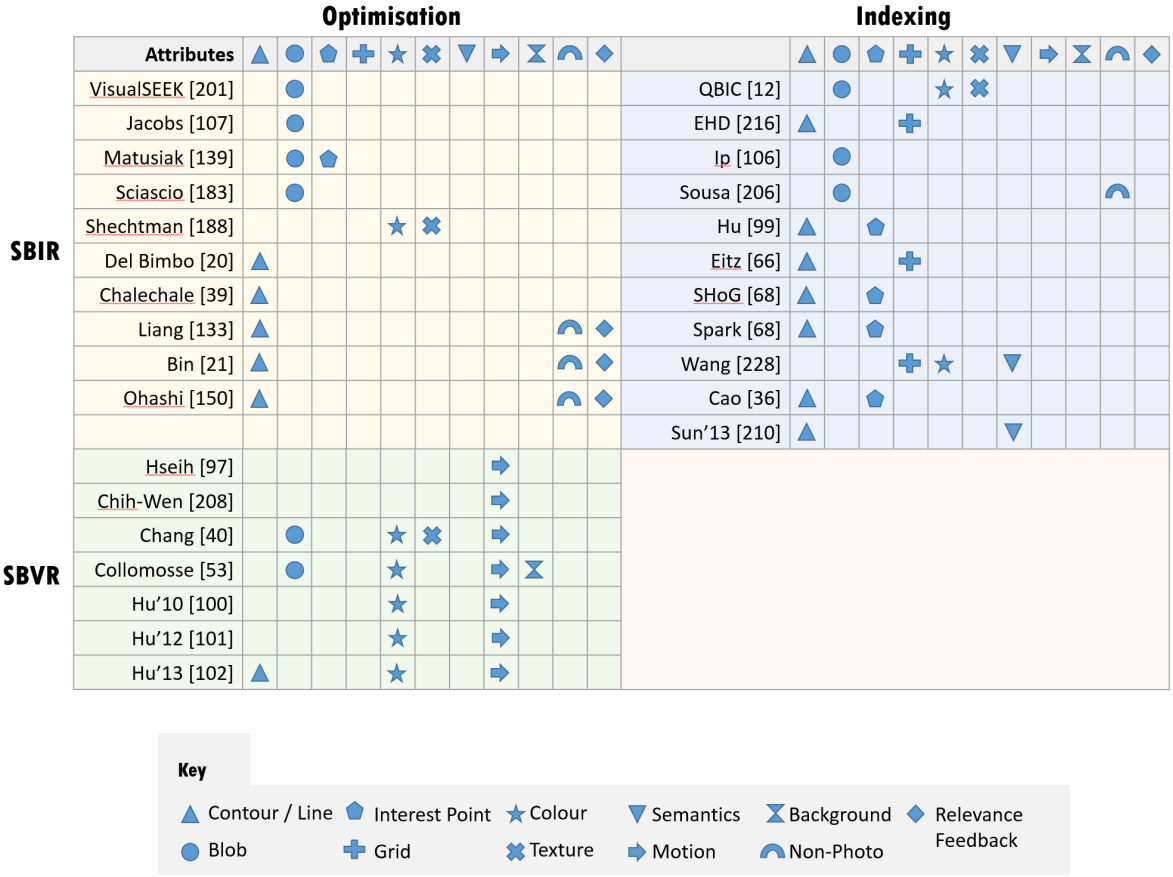



**Figure 2.4** Taxonomy for the Sketch based Image and Video retrieval literature, highlighting features of different approaches.

### Blob based SBIR Techniques

Blob based methods such as QBIC [12] describe the objects facets by a composition of colour, texture and shape independently. QBIC describes colour via colour histogram, texture by a composition of coarseness, contrast and directionality and shape by central moments. The QBIC system overcame the complexities of object segmentation by utilising a semi-automatic indexing process, allowing the user to guide the object segmentation, one of the more challenging tasks. Alternatively Jacobs et al. [107] explored a 'finger' painting style (e. g. sec. 2.9) for query specification, performing matching through the comparison of significant Haar wavelet coefficients between the image and query. This approach was later integrated into the prototype colour blob based search engine 'Retrievr' within the FlickR online photo sharing site. VisualSEEk [201] additionally explored the spatial relationship between image segments ('blobs') allowing for relative objects in an image to be retrieved. This was achieved by ranking relevant region matches then re-ranking based on the spatial relationship resulting in a joint ranking.

Topological models for sketch were explored by Sousa et al. [206] building upon Fonseca

et al. [78] work on retrieval of technical drawings where the topology is described by the eigenvalues of adjacency matrix. Fonseca developed a technique in [79] to describe sketched objects based on a moment style descriptor, describing geometric facets of the object. Sousa integrated spatial proximity into the graph, while maintaining the ability to distil a descriptor from the image, making matching efficient. Blob based techniques can also be said to include shape descriptor style approaches [98], although not always applied within the SBR context.

Many retrieval systems benefit from placing the user 'in the loop', to iteratively refine the search results through providing feedback on the relevance of results — so called 'relevance feedback' (RF). Sciascio et al. [183] provided an early RF approach for SBR utilising a simple Fourier shape and gridded colour histogram descriptor to describe the facets of the image. To perform RF they synthesise a new query merging the positive and negative example descriptors.

#### Contour based SBIR Techniques

Contour based approaches break down into key methodologies: elastic matching, grid and interest point based as defined in fig. 2.4. As opposed to blob based methods colour and texture information does not tend to be encoded within the descriptor in such systems. Although, recently, some algorithms have been proposed that encode semantic information alongside appearance information (so called 'hybrid' SBIR approaches).

Elastic Matching was originally applied by Del Bimbo et al. [20], Del Bimbo uses the elastic energy to optimise the query onto the image. The required energy to deform the $\beta$-Spline becomes the similarity distance to rank database items. This iterative process comes at a high expense. Matusiak et al. [139] defined an approach using Curvature scale space to match image contour. Matusiak's descriptor form improved retrieval efficiency in contrast to Del Bimbo. Ip [106] used high curvature control points, the angles of the vectors from the centre of mass to the control points, to formulate a descriptor for efficient matching. All these approaches make the assumption that the object is singular and dominant within the image.

Within the MPEG-7 standard, the Edge Histogram Descriptor (EHD) [216] was defined, gridding the image and computing edge information within the cells; this approach often is commonly compared to as a baseline in modern sketch literature [99, 66, 36, 210]. Eitz [66] proposed using the structure tensor within a gridded image. Each cell described the dominant tensor vector, through angular quantisation of magnitudes the descriptor was distilled, the concatenation forming a global descriptor.

Using edge points Chalechale et al. [39] proposed Angular Partitioning of an Abstract Image (APAI). Edge points are described by an angular binning method then represented using a

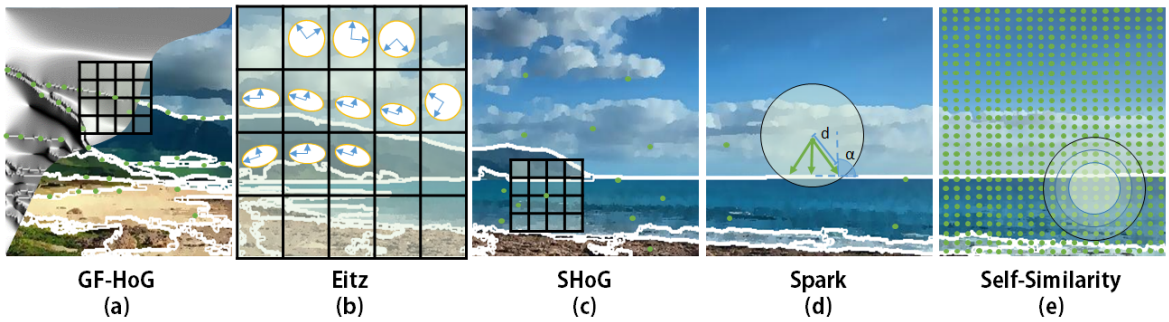



**Figure 2.5** Illustrating the extraction processes for several common SBIR descriptors: (a) GF-HoG [99], (b) Eitz [66], (c) SHoG [68], (d) Spark [68] (e) Self Similarity [68], (d) Self-Similarity [188]

Fourier descriptor. Alternatively the Self-Similarity descriptor [188], although not specifically designed for sketch, also claims to deliver depiction-invariant matching including sketch matching. The Self-Similarity descriptor is computed densely at locations across the entire image (or sketch). At each location, a correlation surface is computed by comparing a patch centred on the location to others adjacent to it. The comparison is made using the sum of squared distance (SSD) between the patches. An expensive evidence gathering process involving a voting space similar to a Hough transform is required to match objects between images. Efficient representations of Self-Similarity using BoVW were proposed by Ren et al. [174], through visual sentences for pose retrieval (discussed further in Sec. 2.5).

Applying the BoVW method [198] for the first time to SBIR, Hu et al. [99] used a dense gradient field interpolated from the Canny edge map of an image to form a descriptor. The dense fields from sketches and Canny edges maps of images appear visually similar, and are treated as a form of synthetic texture dependent on edge structure in the image from which the common HOG descriptor is sampled and fed into the standard BoVW quantisation process. Hu et al. successfully demonstrated SBIR for large scale retrieval ($> 1$ million). More recently Eitz et al. [68] surveyed the application of BoVW strategies for a similar large-scale dataset. Within [68], Eitz presented an extension to: HoG – SHoG based on using points detected from a filtered edge map, and Shape Context – Spark where Shape Context descriptors are cut from random points not belonging to edges, both approaches were designed as Sketch derivatives for BoVW codebooking. The Edgel index [36] was presented by Caoet al. allowing for Indexable Orientated chamfer matching, through an BoVW inverted index. Although the chamfer matching echoed earlier work such as Hu et al.'s gradient interpolation, the efficient indexing explored by the approach was quite novel and for the first time enabled interactive SBIR over a dataset of over 2 million images. Fig. 2.5 summarises common methods for extracting SBIR descriptors.

The incorporation of text annotation within sketched queries has become popular in recent

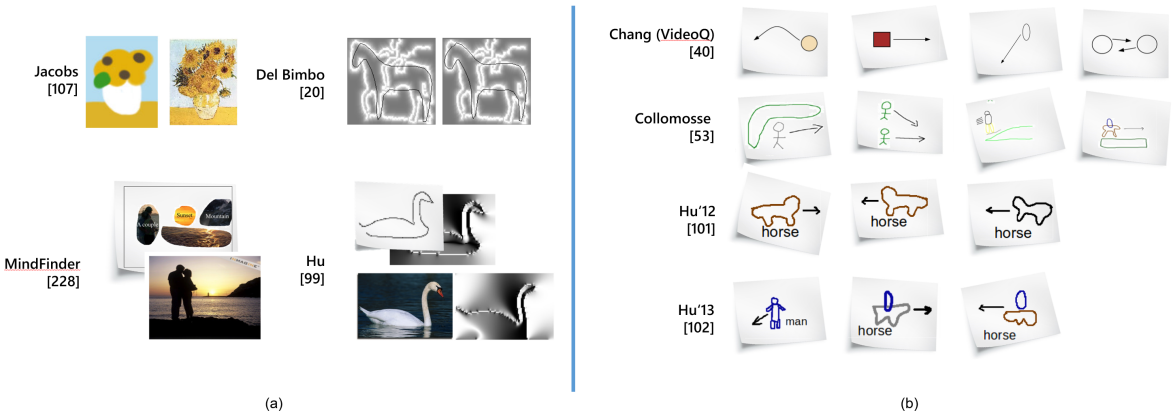



**Figure 2.6** Query styles for SBIR (a) of the techniques of [107, 20, 228, 99] images from respective publications; (b) the query styles of SBVR techniques, showing the various facets of different techniques of [40, 53, 101, 102], images from respective papers.

years. The MindFinder [228] system integrated a traditional text retrieval system within a SBIR system. The visual search approach described the spatial information via Hamming encoding and the appearance information via PCA Hashing allowing for scene appearance and layout to be encoded. A text based image search using a standard web search engine was used to bootstrap the QVE approach with a shortlist of relevant images.

Liu et al. [134] avoiding the difficulties of defining shape through sketch, using instead a query composed of rectangles labelled with semantic tags. Boxes were replaced by reference images of the desired class of object (e. g. boat) and the resulting pseudo-photographic image used as query to a standard photo-realistic QVE matching algorithm.

Recently SBIR has been demonstrated on a very large scale, using billions of images privately accessible to the researchers on the Microsoft Bing search engine. Sun et al. [210] formed a compact descriptor from Cao et al.'s Mindfinder Edgel representation [36], and combined this with a number of engineering tricks including multiple inverse-index tables to cache edge structures. This dramatically improved the efficiency of the custom orientated chamfer matching step of Cao et al. leading to massive improvements in scalability. This work also benefited from textual semantic annotation of images improving the precision of retrieval.

As with blob based SBIR techniques, relevance feedback has been fleetingly explored, with approaches focusing on the retrieval of non-photographic database items e. g. cartoon, clipart. Liang et al.'s [133] approach similar to [206] computes the Eigenvalues of an adjacency graph describing the spatial layout of blobs in an image. Replacing Sousa's sketch line primitives to describe the shape structure, Liang used a SVM over these eigenvalues to performing the relevance feedback. Bin et al. [21] applied Linear Programming for iterative refinement of results, using an updatable ellipsoid distance to measure the similarity of features. Ohashi et al. [150] have employed a similarity based re-ranking approach for relevance feedback in a

blob based SBIR system.

### 2.3.2 Sketch based Video Retrieval

Human recall of events is primarily through 'episodic memory'; we tend to remember temporally ordered accounts of a few objects and their actions or interactions [219]. Exact visual appearance is less important, as are the exact motion parameters (e. g. speed) of objects. Such information is therefore rarely depicted within query sketches.

**Figure 2.7** Set of horse depictions from Hu [99] dataset, demonstrating the difficulty of intra-class and inter-class variability. Most sketches could easily be confused with a different object (e. g. dog, fox).

SBVR techniques began to gain traction in the late 90's with approaches such as Chang's VideoQ [40]. VideoQ enables the specification of multiple appearance facets within a query – shape, texture and colour. Chang et al. compared query to each database item, with each facet being compared in a different way. Shape being compared via comparison of principal Eigenvalues additionally incorporating relative area; texture was compared by three Tamura [213] texture measures – coarseness, contrast and orientation, colour compared by LUV colour distance. For motion they compared the inter-frame compensated trajectory trails [13] by the distance between the trails. In addition to the requirement of exact appearance and motion depiction in the sketch (at odds with the nature of episodic recall) VideoQ's major drawback was its dependency on accurate video segmentation, each segmented region considered to constitute an object. The expectation of this level of video segmentation performance is generally thought to be unrealistic in CV for general footage.

With the introduction of the MPEG-7 standard focus shifted to describing motion trails in an inexact, but efficient way. Liao et al. [208] proposed a system using motion flows for video retrieval, motion vectors from the MPEG-7 specification were used to construct a trail of objects moving within the video. This method works well on complex trails since it is able to describe the detailed motion using macroblocks (as described in MPEG-7), but does not fit well with a more general sketch and episodic recall. Hsieh et al. [97] proposed a system based on similar trajectory extraction to Liao, but clustering the trails to produce a curve. Hsieh described the curve by taking control points of high curvature that characterise the trail that were then constructed into a string, applying edit distance to rank results. An alternative approach for matching trajectories from the video retrieval field was presented by

Shim [189] where k-warping was used to determine a distance measure between motion trails and query. Hu et al. [100] also proposed a motion trail based system. SIFT correspondences were clustered using GMMs then matched to queries using a Trellis-based distance measure similar to the Viterbi algorithm. In contrast to other trajectory approaches based on MPEG-7, Hu incorporated colour into the tokens when matching.

Collomosse et al. [53] aimed to overcome VideoQ's limiting assumption of perfect video segmentation, by purposefully over-segmenting the video and using query-time inference to label super-pixels to sketched objects. Collomosse performed this inference using a linear dynamical system (LDS) that modelled expected object trajectory. Unfortunately these trajectories were limited to straight lines in order to simplify the LDS model to a tractable inference problem. Even so, queries came with a high computational cost. Collomosse also proposed the 300 video clip TSF (sports) dataset that has become a benchmark dataset for SBVR and has subsequently been extended in recent SBVR publications.

Semantics have proven to become increasingly popular in SBIR, since one barrier to scalability (beyond technical considerations) is the inter-class ambiguity of objects present in typical user sketches (Fig. 2.7 illustrates this difficulty using the public SBIR dataset Flickr15k, [102]). Hu et al. [101] were among the first to explore the use of semantics within sketch queries for SBVR, extending their earlier work based on colour and motion cues alone [100]. They coined the term 'hybrid' SBVR to refer to the combination of semantics and appearance information in their annotated query sketches.

In Hu et al.'s hybrid SBVR system [101], Affinity-Propagation was used to cluster tracklets to which several attributes were attached. For example, semantic attributes extracted via a pixel-wise semantic segmentation were assigned to each tracklet enabling later matching of hybrid queries. The Semantic Texton Forests (STF) [193] technique was used to obtain the semantic labels. The approach was evaluated over a subset of the TSF dataset [53], including only two semantic foreground classes (*horse, person*) and a relatively small 140 clips.

Hu et al. [102] later extended on Collomosse et al.'s [53] approach using Markov Random Fields (MRF) optimisation. Similar to Collomosse et al.'s prior over-segmentation approach they also over-segmented video, but into spatio-temporal volumes rather than spatial super-pixels within each frame. The MRF was then solved to assign sketched objects to spatio-temporal volumes. This approach reduced the computational overhead of Collomosse et al.'s inference step; using volumes instead of per-frame super-pixels resulted in fewer entities to label. However the technique is yet another expensive optimisation to SBVR and so scales to larger dataset sizes with only linear complexity. The work was evaluated over only two semantic classes, but with a much larger 500 clip dataset.

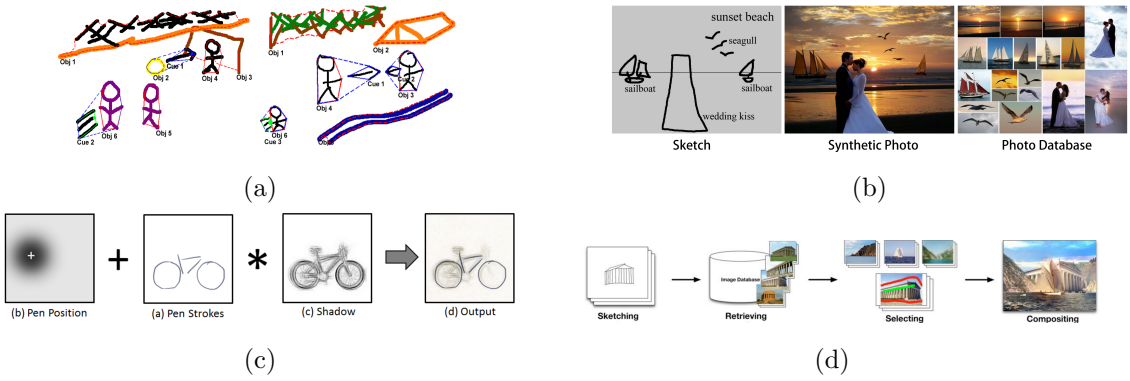



**Figure 2.8** Sample of sketch parsing and interaction interfaces. (a) SBVR query parsing [52], (b) Sketch-to-Photo of Chen et al. [49], (c) ShadowDraw of Haldankar et al. [130], (d) PhotoSketch of Eitz et al. [67]. (Images from respective publications.)

### 2.3.3 Interactive Sketch Retrieval

SBIR has also been used as a tool for assisted drawing. The *Shadow Draw* system of Haldankar et al. [130] uses a simple edge hashing retrieval technique to aid in object depiction. This approach then renders a shadow of retrieved matches to guide drawing. Similarly [10] applied an approach for sketch correction using a spring based system to match to primitive shapes. More recently, *iCanDraw* [62] aided in the drawing of human faces. Conversely Marvaniya et al. [138] proposed a deformable part based contour technique for synthesising sketches from an image. Using salient contours matched from dataset of images edge information was propagated to produce a sketch. For matching of sketches Fonseca et al. [79] presented CALI based on a composite of geometric information a compact descriptor matched simple sketches, used for quick navigating of assets.

The synthesis of images from sketches has also been a popular topic with *Sketch-to-Collage* [80] collating and blending segments of images from a database based on a query guide. *Sketch-to-Photo* [49] pulled images from the Internet to compose an image. In this case the query used text label annotations to reduce the complexity of CBR. *PhotoSketch* of Eitz et al. [67] approached the CBR challenge using their previously proposed Tensor Structure descriptor. Eitz matched a sketch to image with the user drawing over the parts of the image to be composited onto a different background.

## 2.4 Image Annotation and Semantic Segmentation

Image annotation is a popular method in the IR community, as opposed to focusing on the challenge of matching based on CBR techniques a set of image attributes (e. g. objects) are often detected to form an attribute descriptor. Therefore the task of annotating images

(heavily studied within CV) has relevance to the IR community. Image annotation is the description of the image through a set of labels, alternatively image categorisation assigns a single label to the image, we focus on the former that is most relevant to the work of Chapter 3.

Often semantic tags are generated through standard object recognition techniques, using CBR approaches like BoVW or more recently convolutional neural networks to describe the image. There are three main approaches to image annotation: keyword – images are labelled with a set of words, these can be arbitrary words or from a predefined vocabulary; keyword ontologies – A hierarchical structuring of concepts (keywords) to form relationships e. g. "is a" and "has a"; and free text – any combination of keyword or sentences, this approach is challenging for CV, therefore is commonly used in conjunction with one of the other approaches.

Early approaches were evaluated by Bernard et al. [14] since then larger datasets for Image annotation and object recognition have been presented. Datasets such as those provided by the PASCAL VOC challenge [69] provide classification benchmarks to challenge the cutting-edge, with the latest versions having 120 categories. The challenge of deciding a tag vocabulary has been explored with popular approaches [15, 244] applying WordNet [143]; an ontology of words. The most recent dataset to become widely popular is ImageNet [205]; a collection of images corresponding to a subset of WordNet labelled with 21841 keywords (only nouns are explored in ImageNet) comprising 14,197,122 images of which 1,034,908 are annotated with bounding boxes for detection. Commonly ImageNet has only one keyword per image (whereas PASCAL had no such guarantee), nevertheless the scale of the dataset has led to its widespread adoption as a benchmark especially for involving Deep Learning based techniques.

Deep Learning has become an increasing trend for classification in CV. With origins from ImageNet Krizhevsky et al. [124], proposed the structure of learning visual features from a large pool of data using convolutional neural networks. Configurations of convolutional neural networks techniques for Image Classification have been explored by Chatfield et al. [45]. The convolutional neural networks technique has been adapted to apply to object detection [83] utilising segmentation techniques. Oquab et al. [152] used midlevel features trained from an ImageNet model, to the PASCAL object detection challenge.

Alternatively segmentation can provide a deeper insight into the content within the image. Semantic segmentation is an approach of segmenting images based on classes. The concept is similar therefore a classifier is trained to be able to detect the class of a pixels or superpixels. Early methods used alternative descriptors to those used within BoVW, descriptors such as Texton [136] (filter banks); this descriptor is better able to describe the texture of objects and is therefore more robust over scenes.

Shotton et al. [194] proposed *TextonBoost* to solve the semantic segmentation problem, this

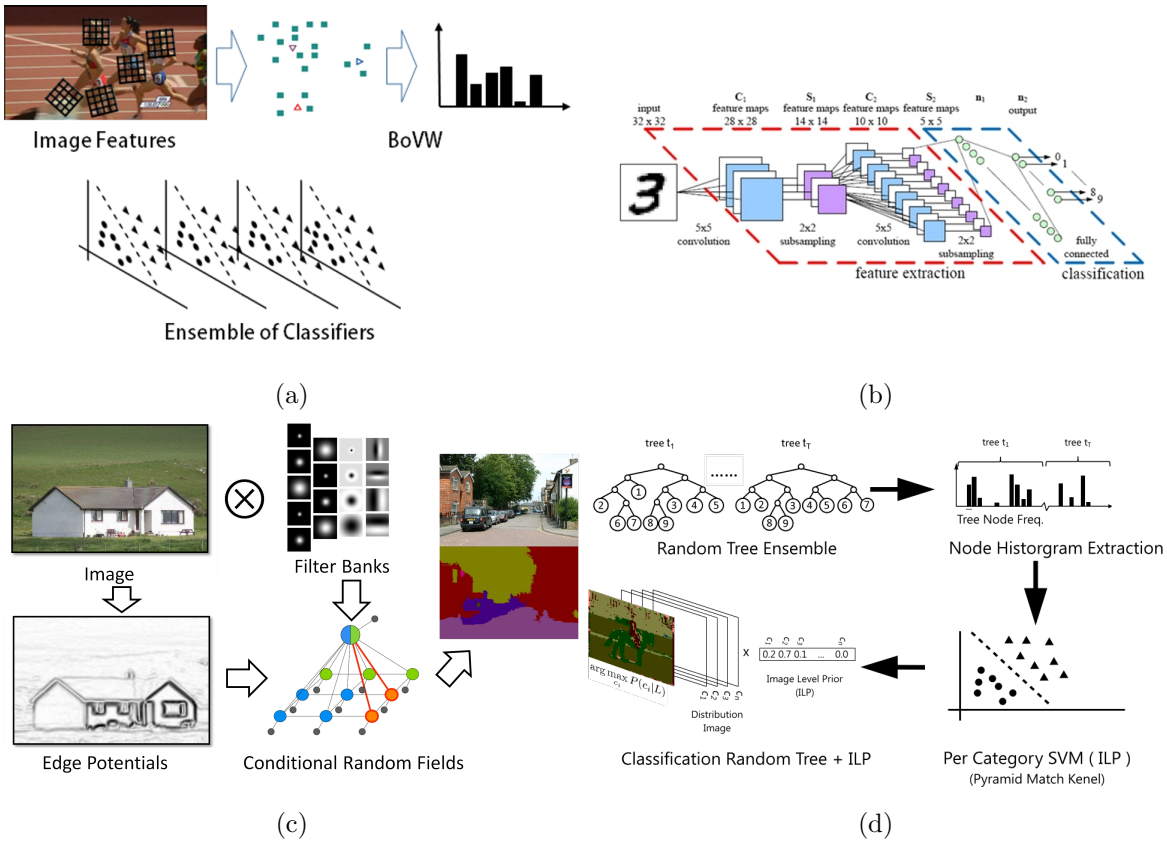



**Figure 2.9** Examples of different Image Annotation and Semantic Segmentation Techniques (a) BoVW pipeline for Image Annotation (b) The conventional structure of Convolutional Neural Networks [205] (c) Texton guided segmentation, using CRFs for spatial consistency [194] (d) Ensembles of Trees for segmentation as per [193]. Images from or derived from respective publications.

approach used texture-layout features. These are local descriptors cut from a K-Means assignment of texton descriptors within an image. This descriptor models texture information and layout was modelled within a Conditional Random Field (CRF). Shotton used a modified version of Joint Boosting algorithm for classification of pixels, and used a random sampling approach to train the system to avoid the issue of a large sample set. This approach suffered from expensive K-Means assignment (even with a low K, a 320x240 image is slow to assign) as well as challenging to solve CRF, taking greater than 10 seconds on larger images.

Moosmann et al. [146] explored the use of randomised decision forests as an alternative to K-Means for clustering. This work focused on clustering but motivated Shotton et al. [193]. Shotton expanded on this work performing semantic segmentation, Shotton showed that the implicit hierarchical structure made it faster, by not requiring the computation of textons or BoVW. Instead Shotton used CIELab pixel values differences within a window. Additionally a global descriptor is able to be extracted from Decision tree nodes, producing a image-level prior (Bag of Semantic Textons) this provided some context, but not the spatial consistency

common from CRF methods. The major benefit to [193] was in the computational performance, requiring only a couple of seconds. Adaptations to this approach have been presented including multiscale [119].

Although slower than Johnson's approach Ranganathan [173] reverted back to the use of textons and K-Means based approach but adapted the learning step to use Random Multinomial Logit (RML). Ranganathan argues that uncontrolled cluster size of Random Forests made K-Means a more reasonable approach. As an alternative to K-Means and tree based second order pooling was explored by [37] based on SIFT features, this approach was demonstrated to be computationally faster than conventional VOC challenge methods, but still more expensive than [193].

Due to the time consuming nature of manual annotation weak labelling based approaches have become more popular in recent years [226, 225]. Vezhnevets [224] presented an approach based on Active Learning modelling the problem through a CRF for finding the most informative nodes.

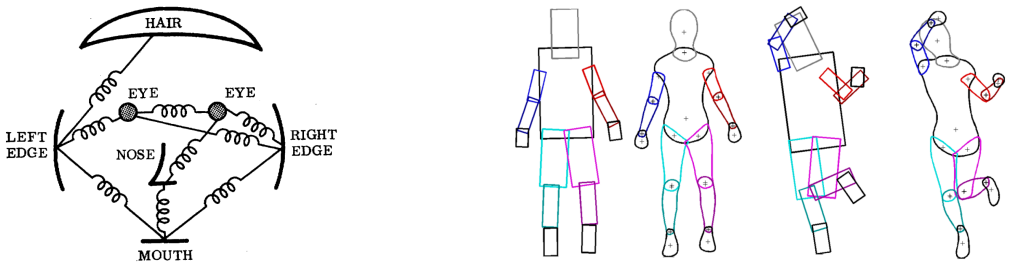
## 2.5 Human Pose Estimation and Retrieval

Pose estimation has gained increasing amounts of interest in literature, with the success of pedestrian detectors making it plausible to classify body parts. Pose methods have additionally benefited from the proliferation of depth devices e. g. Microsoft Kinect. Within 2D footage common methodologies fit into either: part based models or explicit/soft estimation(e. g. PDF) methods. For this thesis we only explore 2D (subsec. 2.5.1), but additionally we consider methods from 3D (subsec. 2.5.3) as well. Several survey papers are drawn on for this section [145, 161, 163] with [145] acting as a comprehensive overview.

### 2.5.1 Human Pose Estimation from Image

Early pose estimation focused on the use of either pictorial models of Fischler and Elschlanger [75] or through [22]. Approaches such as Felzenszwalb et al. [72, 71] applied the pictorial structure model in its original form optimising matching. Building on Felzenszwalb, Ronford [176] applied linear SVMs for discriminative body parts, alternatively Johnson et al. [116] applied a cascade detector.

Ramanan [171] matches an edge-based deformable model to learn soft estimates of the body parts, building upon this, a region-based deformable model is fitted improving estimation performance by implicitly learning the colour of relevant regions. Ferrari et al. [73] follow




(a) (b)

**Figure 2.10** (a) Original pictorial structures for faces of Fischler and Elschlanger [75] (b) Examples of pictorial structures [71] and deformable structures [245]. Where figures are from respective publications.

in a similar style learning a weak representation through a combination location scale information through an upper body detector, foreground highlighting using a modified version of GrabCut [177] and head and torso constraint. Yang et al. [235] proposed using a flexible mixture-of-parts. They learn co-occurrence, appearance and deformation models applying inference through a relational graph using dynamic programming. Yang's approach was more general than Ferrari allowing pose estimation not just Human pose. Andriluka et al. [6] explored a variety of features for pose estimation. Alternative methods that are less sensitive to occlusion [196, 113] by applying additional constraints for when two nodes of the a tree structure can be chosen, or even fully connected graphs [18].

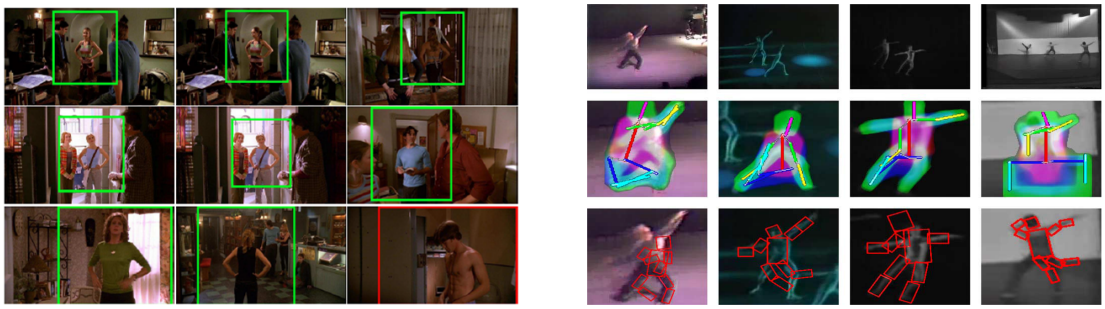
Hierarchical approaches do not specifically estimate independent body parts [209, 217]. As an alternative to the tree structure, a more holistic approach based around a NN exemplar lookup was presented by Moriet al. [147]. Derivatives such as [186] apply locality sensitive hashing or semi-global classifier for part configuration [85].

Following the Deep learning trend, DeepPose [218] learns texture style features based on optimising the convolution filter used at different stages of the Neural Network. Toschev et al. demonstrated that this approach achieved an 11% increase over current state-of-the-art.

### 2.5.2 Human Pose Retrieval of Image and Video

Ferrari et al. extended [73], for pose retrieval in [74]. They proposed 3 descriptors – part positions, part orientations including relative location and orientation, soft segmentation. In experiments they showed the composition parts and relative information performed best. Although this can be said to be the earliest of QVE pose retrieval systems, it was limited to the upper body, with upright poses (fig. 2.11a). Pose based on a blob based representation was presented by Shechtman et al. [188], using a Self-Similarity descriptor, measuring

the correlation within a patch and computing a log-likelihood descriptor at a local level. Self-Similarity matching, as proposed, is expensive requiring matching between both sets of dense descriptors. Ren et al. [174] demonstrated, relying on HPE was inadequate in more general scenarios (fig. 2.11b). Ren extended Self-Similarity applying BoVW encoding, then learning sentences running through a scale-space pyramid via topic modelling. Ren et al. work overcame the challenge of inference of a skeleton (or soft estimation) representation on unconstrained dance datasets.



(a) (b)

**Figure 2.11** (a) Using upper-body human pose estimation for retrieval [74] where the top left is the query and the top 9 results.(b) Challenges of human pose estimation making it currently unsuitable for retrieval showing: top – original frame, middle – approach of Ferrari et al. [73], bottom – approach of Yang et al. [235]. Where figures are from respective publications.

The aforementioned approaches rely on a QVE. Jammalamadaka et al. [109] explored a variety of queries, distilled into a articulate skeleton as a query. To describe images Jammalamadaka combined the pose estimation from Yang [235] and Ferrari [73], using correlating the agreement between detectors, then describing the upper body through a 12D joint descriptor composed of *sine* and *cosine* components to overcome the disjoint between 0-360 degrees. Although this approach allowed for greater variation in query style it still suffered from the requirement on HPE.

### 2.5.3 3D Human Pose Estimation from Image and Depth

Early 3D pose estimation methods from an image followed a model fitting approach, where the pose was estimated by directly inverting the kinematics [215] or numerically optimising the configurations. [215] assumed that the joint angles were known and solved the limb length through their relative depths. Agarwal [2] used shape context descriptors and regressors to infer the joints from silhouettes. Utilising 2D techniques, [220] used several frames of weakly estimated 2D poses, to then convert to 3D using a rigid structure. Using a mapping from 2D to 3D Salzmann [179] explored the use of Gaussian Processes to learn a subspace for such a mapping. Ramakrishna [170] taking a memory approach used a large corpus of Motion

Capture data with visual cues to infer the relative skeleton, similarly using 2D guide points.

Depth based pose estimation techniques have become increasingly active area of research with the introduction of the Microsoft Kinect, making depth information commercially available. Prior to this 3D pose was inferred predominantly from Multi-View or expensive depth cameras. Shotton [192] extended their Semantic Texton Forest [193] to work with depth data, essentially making the difference function from depth as opposed to colour difference. Continuing with the tree based methods, [84] applied Hough Forests. Applying a MRF Anguelov et al. [7] learnt body configuration using maximum-margin framework and performing a graph cut. Exploring a nonlinear optimisation [87] used Iterative Closest Point (ICP) to estimate with 28 degrees of freedom. Applying a heuristics based approach Zhu [243] identifying the head and torso to constrain the problem. Similarly [195] utilised specialised designed detectors to identify such body parts. [166] estimated the orientation of specially selected interest points to provide local shape information.

## 2.6 Animation and Video Synthesis

Considerable effort has been invested in the automation of animation, due to the huge amount of manual effort typically required to produce realistic content. One frequently used method to automate animation whilst maintaining realism is to capture fragments of motion (e.g. using a skeletal motion capture rig such as a Vicon system) and stitch these fragments together to create the desired movement. This approach is sometimes referred to as concatenative synthesis. Concatenative synthesis has its origins in speech synthesis, and was extended to character animation and video synthesis in the early 2000s.

### 2.6.1 Character Animation

Kovar et al. [123] introduced the *motion graph*; a graph data structure in which edges constitute segments of motion and nodes the connections (or 'transition points') between them. Walking around the graph through several transition points generates a list of frames that can be concatenated in that order to produce visually near-seamless motion. Kovar et al. focused on the domain of skeletal animation and identified nodes (or transitions) based on a fixed threshold on pose similarity between pairs of motion capture frames. This approach was later expanded by Gleicher et al. to a hierarchical representation ('move tree') which enjoyed popularity within the computer games industry.

An alternative to graph based approaches, Brand [28] applied a hidden Markov model to model a manifold of plausible poses in *Shadow Puppetry*. This allowed the inference of a

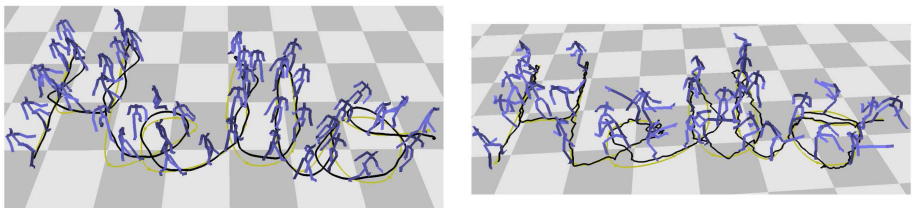


**Figure 2.12** Character animation synthesised using concatenative synthesis facilitated by a motion graph data representation. The motion graph models routes through the graph generate novel plausible motion sequences. (Figure from Kovar et al. [123])

3D skeleton from sequences of silhouette. Sequence generation is the process of finding the geodesic distance across the manifold that best supports the silhouette sequence; in essence the manifold acts as a space of plausible poses. Lee et al. [129] extended Brand's markovian approach using a two-layer system, with the second being based on statistical models through clustering.

### 2.6.2 Video Synthesis

Early video synthesis techniques focused on animating speech. Bregler et al. [31] introduced *video rewrite*, using a database of mouth images to animate a natural speaker. Animation was performed by looking up relevant segments of the database and animating the mouth and jawline based on reordered segments. Similarly [70] animated speech using a multidimensional morphable model, to parametrise the mouth of human speech. A manifold approach Brand [30] used [28] to synthesise mouth motion.

For the animation of video, *video textures* [181] applied a Markov chain approach by treating frames as states and allowing transition across, video based on frame similarity. This approach was ideal for generating looping sequences, e. g. fire animation. Video Textures also introduced video based animation as an extension allowing high-level user control. Schodl extended video textures to create character animation in *video sprites* [180] by adapting a cost function sprites of animals were animated according to a motion trajectory.

Alternatively by utilising motion capture data in combination with video footage Flagg et al. [77] implemented the animation of human video textures. Allowing a more visually appealing result than the previous methods [180], this approach still suffered from visual occlusion problems caused by single view point capture.

All of the aforementioned approaches require transitions to smoothly render a new video. Transitions, despite efforts in selection often cause visual discontinuities, therefore blending approaches are performed. Commonly simple linear blending occurs such as in [181, 180] alternatively, in order to further improve coherence PCA [151] can be performed to blend only important parts of the image for the transition.

## 2.7 Transfer Learning and Domain Adaptation

An assumption in conventional machine learning [155] is that the training data and the test data are in the same feature space and/o distribution. However in practice this is often not the case, and learning a transform from the test data to the training data can boost performance.

One way to deduce this transform is to use a few correspondences defined between the two domains, and so create a mapping between test and training domains; however the transfer learning literature also tackles with the problem of learning the mapping with no correspondences at all.

Transfer learning often breaks down into three tasks, 1) what to transfer, 2) how to transfer, and 3) when to transfer. Not all knowledge is applicable to both domains, therefore deciding "what to transfer" leads into "how to transfer" the relevant data. Survey papers such as [155] outline a taxonomy of the field, breaking it down into the three key areas: *Inductive Transfer Learning* – the target task is different to the source task independent of whether the domains are the same or not; with methods such as [169]. At minimum some labelled data in the target domain is required to induce a objective predictive model. *Transductive Transfer Learning*– source and target tasks are the same but the domains are different, commonly few or no labels are available to the target domain; with methods as [114, 238, 190] *Unsupervised Transfer Learning* – in this case the tasks between source and target are the same, but no data is available in either domain [230, 57].

Domain Adaptation (DA) is a specialised part of *Transductive Transfer Learning* where labelled data is only available in the source, but is solving the same task. DA as proposed by Arnold et al. [9] has been extensively studied for its beneficial properties of boosting performance for similar datasets, survey papers such as [114, 131, 240, 157, 137] provide a comprehensive overview of the field. Most techniques focus around learning a feature space mapping from the distribution of the test/train data. Approaches such as [23, 5, 59] applied structural correspondence learning algorithm, to learn this mapping. Alternative approaches extend co-clustering [56] or a classifier [234] to propagate labels. Dimensionality reduction using Maximum mean discrepancy embedding [153] to reduce difference between domains, with speed optimisation [154]. Specific to the visual DA approaches have been applied in face recognition [168, 95] and Object recognition [149, 86]. With limited exploration in human pose estimation [43] to the best of our knowledge it has not been explored within Human Pose Retrieval.

## 2.8 Action Recognition

Action recognition approaches commonly focused on video analysis of human events. These events are often simple concepts like walking, jogging, boxing and jumping as in the KTH dataset [182], however more complex action datasets such as Hollywood 2 have recently emerged. There are several approaches to recognising the events in videos that will be explored in this section looking at both feature methods and action modelling.

Laptev et al. [126] were first to propose the spatial-temporal extension. They implemented an expansion of Mikolajczyk's [140] Harris-Laplace detector – to perform iterative corner detection in 3D (space-time). Laptev proposed using a concatenated SIFT feature vector over the space-time cuboid, describing this as a 'behaviour descriptor'. Laptev's work was then expanded on by Schuldt [182] improving the matching approach from a greedy match of features points to a framework using SVMs to classify actions.

Dollar et al. [63] proposed using specialised 3D detectors instead of 2D extensions. Dollar used small sub-video windows to describe small events within the video such as an eye opening. Dollar's detector focused on periodic motions, but can be applied to complex motion as well. The high computational cost was solved by Ke et al. [121] using integral video to achieve real-time processing, Ke used Haar-wavelets computed on optical flow. Willems [231] proposed a dense Hessian Spatial-Temporal interest points (Hes-STIP) detector that was scale invariant to in both spatial and temporal domains and was able to densely cover video content.

BoVW was applied to action recognition modified to use spatial-temporal feature points. This required the use of the aforementioned detectors and development of descriptors able to handle the temporal dimension. Scovanner et al. [184] expanded the standard SIFT descriptor into three dimensions by adding a further dimension to the orientation histogram this encoded the temporal information. Willems [231] expanded the SURF descriptor in their experiments for Hess-STIP detector and it outperformed the common jet descriptor although was not compared with the 3D-SIFT descriptor. Chen et al. [48] explored incorporating motion into SIFT, Motion SIFT (MoSIFT). Chen replaced the HoG component with optical flow using the same technique as SIFT. By concatenating the optical descriptor with the standard SIFT for describing the appearance you get a descriptor that is able to handle appearance and motion. In Chen's scenario they omit orientation invariance since they are working with a static camera.

An alternative approach using BoVW by Mikolajczyket al. [142] used an array of existing standard detectors and descriptors. Using Harris-Laplace, Hessian-Laplace, MSER, and Pairs of Adjacent Segments to detect interest points, these were encoded using Gradient Location and Orientation Histogram (GLOH), for appearance, combined with Lucas-Kanade Tracker

(KLT) optical flow information (for motion) which is then dimensionality-reduced using PCA. Motion compensation was applied to the optical flow vectors prior to incorporation in the descriptor, using inter-frame homography estimation. Recognition of events was performed using search trees, exploiting randomised *kd*-trees with approximate nearest neighbour for classification.

Hidden Markov Models (HMM) provide a powerful way of recognising sequential events within a stochastic process. They have been used within action recognition for this reason since the late nineties. Brand [29] demonstrated that using minimisation of the joint distribution entropy, activity can be classified into meaningful states. Robertson [175] used HMMs for smoothing of action sequences by encoding rules of scenes to produce more robust action recognition. Their approach used a combined location and motion point's descriptor to model the actions within a video. More recently Jiang [115] improved on the HMM learning algorithm. By improving the parameter estimation, their HMM system avoids the problem of local optimal and therefore is able to find the global optimal solution.

Data mining techniques are commonly used for finding patterns in sparse data, although had seen limited use in CV until recently. Gilbert et al. [82] proposed using the *a priori* algorithm [3] originally designed to find correlations within supermarket shoppers' transactions. They modify the algorithm to classify events using spatial-temporal key points; unlike the approaches described above they used two dimensional points and expand over the temporal domain to create volumes denser in nature that Laptev et al.'s [127]. Key points are then mined to determine the most discriminative points for classification.

## 2.9 Summary

We have identified several gaps within the literature relating to sketch based video retrieval that will be tackled in this thesis:

**Efficient Index-based SBVR**

Although SBVR remains a comparatively recent sub-field within the sketch based interfaces literature, a number of successful approaches have been developed and shown effective over databases of a few hundred videos. Some very recent approaches have begun to explore the fusion of multiple modalities within the query sketch (including shape, colour, motion and even object semantics through text labels in the sketch). Whilst the accuracy of such approaches is good, all are very slow due to expensive optimisation or inference steps used to fit a model derived from the sketch to each item in the database. Such approaches scale linearly at best, with dataset size, and a single

comparison often takes a few seconds leading to query times of several minutes in total. We note that no prior SBVR approach using 'hybrid' i. e. multi-modal combinations of features uses an efficient feature-based search index. This is anomalous since almost all CBVR methods are index-based and most modern SBIR methods (e. g. based on BoVW [68, 102]) are too.

**Human Pose Retrieval**

Human pose retrieval has been fleetingly explored in the literature, but to-date most techniques rely on explicit estimation of human pose from photographs [74, 109]. It has been shown that such techniques are limited by the pictorial structures approach they are based on, and they struggle with more free poses such as those encountered in Performance Dance [174]. Although an implicit approach has been explored through modification of Self-Similarity [174], it has treated the problem as one of pure retrieval, without considering the highly non-linear space of possible poses. We also see that to-date human pose retrieval has only been explored through the use of photorealistic QVE rather than sketch based query.

**Visual Narratives (VNs)**

VNs that have only been explored in the context of single-frame storyboards as presented in [53]. Although it is foreseeable that approaches could be extended to incorporate a longer narrative [102], the computational expense and the difficulties of describing the different modalities are likely to limit the success of existing approaches.

There are other short-fallings in the state of the art relevant to the contributions of this thesis. Image annotation has been demonstrated to reach a satisfactory performance at the cost of computational speed. Action recognition has been a highly active area for many years, but struggles to achieve accuracy over unconstrained video. Despite the increase in popularity of applying machine learning techniques to CV problems, transfer learning has had limited influence on CV applications. To-date transfer learning has focused predominantly on the areas of image classification rather than video or sketch based retrieval.

# Chapter 3

# Multi-modal Video Indexing for Sketch based Retrieval

We present a novel spatio-temporal descriptor for representing and indexing a video repository for sketch based video retrieval (SBVR). Our descriptor encodes the semantic class, appearance (shape and colour), and the motion direction of each foreground object within the video. The appearance of the background is also captured. The resulting multi-modal video index is searchable using free-hand sketches depicting appearance and motion, accompanied by textual annotations indicating object class. The combination of sketch and text within a query, coupled with a fast video indexing scheme, results in a 'hybrid' SBVR system capable of searching $\sim 700$ videos in less than one second; several orders of magnitude faster than prior hybrid SBVR systems. This speed-up enables, for the first time, interactive refinement of SBVR results via a relevance feedback (RF) framework yielding state of the art performance for hybrid SBVR.

## 3.1 Introduction

Sketches are inherently multi-modal; they capture many aspects of appearance (shape, colour) and can be annotated with motion cues or labels to capture the motion and semantic class of the depicted objects. Despite this useful property, and the growing popularity of video on touch-screen devices amenable to gestural input, no practical approaches to sketch based video retrieval (SBVR) have yet been developed. Rather, current SBVR work falls short of practical requirements in two main ways: 1) by focussing on just a single modality e. g. shape or motion [50, 100]; 2) adopting robust but slow optimisation strategies for sketch-video matching that typically take minutes to search just a few hundred videos [53, 101].

This chapter presents a novel SBVR system that, uniquely, is both multi-modal and operates at interactive (real-time) speeds. To deliver this system two novel technical contributions are proposed.

1. **Spatio-temporal Indexing.** We propose a novel descriptor based on the encoding of foreground objects within space-time quantisation of the video clip. This descriptor is amenable to matching using a metric distance (e. g. $L^n$ norm) and so is appropriate for storage and indexing within efficient search structures, such as a $kd$-tree, leading to sub-linear complexity at query-time with respect to dataset size. By contrast existing hybrid SBVR use optimisation approaches that fit the sketch as a model to each video clip with the posterior likelihood of the fit used to rank each video in the results. Such per-record optimisation approaches scale with linear complexity at best. The descriptor itself is a 'hybrid' encoding of the foreground content, comprising object semantic category (e. g. car, person), alongside motion and appearance information (shape, colour). Although hybrid SBVR has been explored before using optimisation approaches, an indexing approach has not previously been presented.

2. **Relevance Feedback (RF).** Users may iteratively work with our system to refine the returned results by flagging a few 'good' (relevant) or 'poor' (irrelevant) matches. The RF framework accepts this interactive feedback as a further steer on the matching process, presenting new successive iterations of results for user consideration. RF has not been applied to SBVR previously, yet when dealing with large video datasets it is essential for two reasons. First, the inherent ambiguity of sketch as an under-complete description of the desired video implies the need for further user input to refine results (many false positives can be expected as dataset size increase). Second, the priority of different modalities is not defined within a sketch. Consider a user sketching a red car — is a result containing an arbitrary red object more relevant than a car of arbitrary colour? Without further steer from the user it is impossible to judge.

Our video descriptor encodes a coarse description of scene content, mirroring the approximate nature of sketch. We consider video as a space-time cuboid and quantise this into space-time sub-volumes (cells). Each cell represents local shape, colour, semantic class and motion parameters of foreground objects occurring within those sub-volumes. Additionally, we demonstrate flexibility over prior work through our ability to incorporate background appearance constraints within our search query. Previous hybrid SBVR systems have enabled search only on the foreground (moving) objects within the scene and so have ignored spatial structure in the background, e. g. landscape features such as buildings, mountains, horizon line. Fig. 3.1 presents representative query sketches accepted as input by our system. We

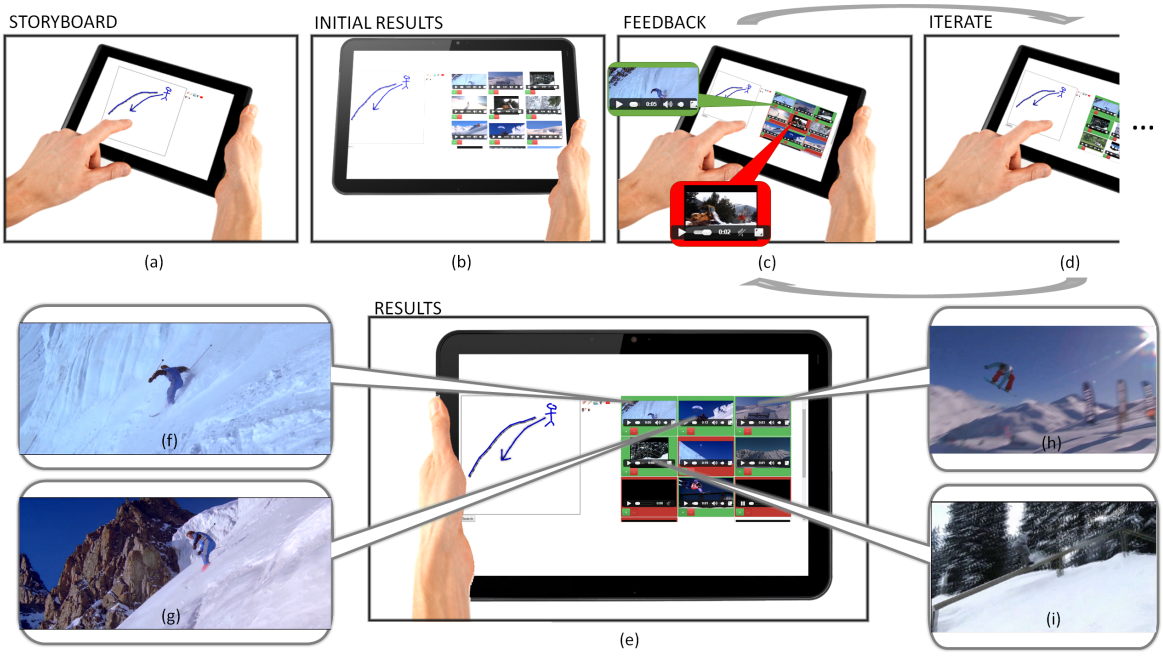



**Figure 3.1** Representative examples of query sketches (a) accepted by the user interface of our SBVR system, initial results displayed to the user (b). Results are interactive selected of positive or negative results (c) for the system to provide updated results (d), which is iterated till satisfied (e). (f-i) demonstrate the top 4 results displayed to the user after RF. Web interface visualised on a tablet

explore RF through a computationally cheap approach comprising an ensemble of classifiers – Multiple Classifier Learning (MCL) – over the multiple orthogonal facets of the descriptor. Although our framework for RF is not novel per se, the ability to perform RF for SBVR is completely new and it is interesting to conclude that RF is practical for SBVR.

## 3.2 Pre-processing and Feature Extraction

Sketched search queries are highly abstract, depicting only a few salient objects and their actions. This presents a significantly broader semantic gap [88] to classical visual search where close agreement is assumed between the photometric properties of the query and the target video. McNeill et al. explored the relationship between the episodic nature of human recall and these abstract free-hand sketches used to depict events in information retrieval tasks [52]. In addition to geometric (and often iconic) simplifications of form, sketched queries also simplify motion in the scene. Only the motion of objects relative to the scene background is depicted, and even then only approximate trajectories are indicated. The length of arrows or motion paths sketched to indicate motion often has no correlation with the duration or speed of the movement.

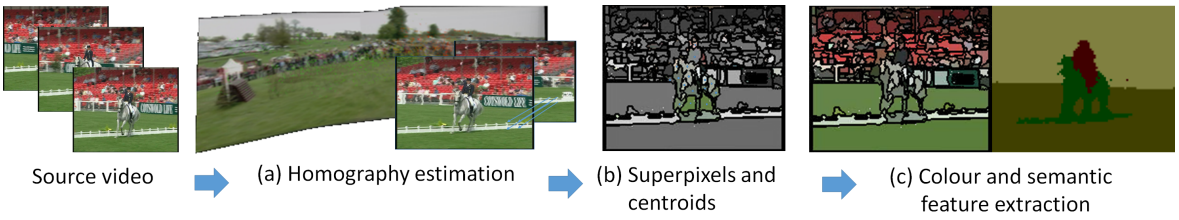



**Figure 3.2** Video ingestion process. (a) Camera ego-motion is approximated through inter-frame homography estimation. (b) Super-pixels are identified and classified as foreground/background based on an optical-flow derived estimate of their centroid motion. (c) Local histograms are extracted for each super-pixel based on colour and semantic classification of their component pixels.

An initial processing step is therefore to estimate camera ego-motion so that it may be subsequently compensated for when considering object motion. The estimate is computed by calculating the inter-frame homography $H_t$ between adjacent frames ($I_t$, $I_{t+1}$) using a standard RANSAC approach over sparse feature correspondences. In doing so we assume that the majority of the features detected arise from background texture. Several engineering tricks were used to robustify the homography estimation process over general video. Multiple keypoint detectors and descriptors (SIFT [135], GLOH [141]) were employed to improve the number of putative correspondences established between frames. We filtered correspondences based on back projection, after [135]. Cases of degeneracy are common via this method, and detectable through monitoring of $\|H_t\|$, where very large or small values trigger a restart of the non-deterministic RANSAC process. The result of this process yields a set of inter-frame homographies $\mathcal{H} = \{H_1, \ldots, H_T\}$ for $t = [1, T]$ for a given video clip.

These estimates are subsequently used to determine the camera-motion compensated position of each super-pixel in the clip, for which colour and semantic features are also extracted as follows (Fig. 3.2).

### 3.2.1 Detecting foreground object fragments

Each frame $I_t$ is independently over-segmented using mean-shift [54] into a set of of super-pixels $\mathcal{S}_t = \{S_t^1, ..., S_t^m\}$; typically $m \simeq 200$ with minimum area $|S_t^i| = 20$ pixels (Fig. 3.2a illustrates such a segmentation). Although such an over-segmentation does not maintain complete object boundaries, it allows for partial object boundaries to be identified and is useful in enforcing spatial consistency of subsequently extracted features.

We extract a set of 2D keypoint locations $P_t = \left[p_t^1, \ldots, p_t^m\right]$ from the centroids of superpixels $\mathcal{S}_t$ in each frame. The keypoints are transformed into a camera-motion compensated reference

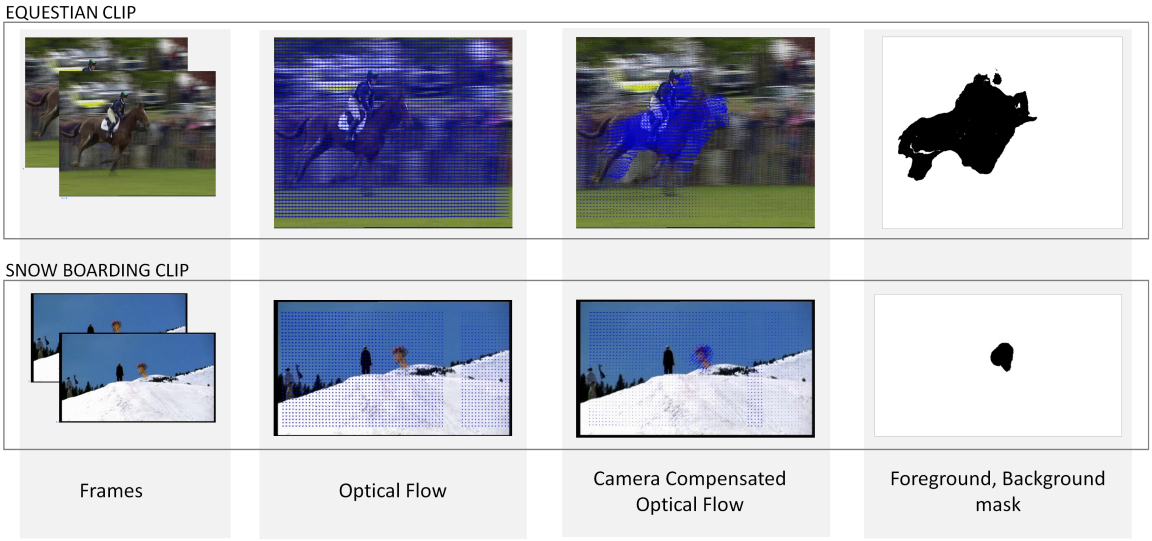



**Figure 3.3** Identifying foreground and background superpixels based on motion (two example clips). Left-right: source video frames (a) are used to estimate a dense optical flow field (b) which is transformed by $H_{t-1}^{-1}$ to compensate for camera ego-motion (c). Thresholding the magnitude of the flow vectors yields a decision mask (d) which is used to partition keypoints $P_t$ (and so their corresponding super-pixels $\mathcal{S}_t$) in to foreground and background sets.

frame yielding locations $P'_t$ s.t.:

$$p_t^{i\prime} = \prod_{h=1}^{t-1} H_h^{-1} p_t^i \tag{3.1}$$

In line with prior SBVR that identifies only moving objects as candidates for inclusion in a query sketch [53, 101], we seek to partition $P'_t$ in to foreground (moving) and background (static) keypoint sets. We identify as background, keypoints in $P'_t$ moving with speed lower than a threshold value. The speed of each keypoint $p_t^{i\prime}$ is determined using a dense optical flow field computed via the method of Brox et al. [32], transformed by $H_{t-1}^{-1}$. The threshold is determined by averaging $H_{t-1}^{-1} p_t^i$ at the frame boundaries, which we assume to contain mainly background content. Such a threshold can be considered a mask over $I_t$ demarking foreground object from background (Fig. 3.3d). Applying this threshold to the keypoints enables the set of superpixels at any time $\mathcal{S}_t$ to be decomposed into foreground and background superpixels ($S_t^{FG}$ and $S_t^{BG}$ respectively).

Several features are subsequently extracted from $I_t$ for each of the superpixels in $\mathcal{S}_t$.

### 3.2.2 Colour feature extraction

Colour features are extracted as a colour histogram computed from pixels within each super-pixel $\mathcal{S}_t^i$. A histogram representation requires quantisation of the colour space into bins,

for which we use the standard 'Macintosh 16 colour palette' (fig. 3.7c) available in the user's query sketch interface to enable rapid comparison at query-time. This idealised palette differs from the dominant colours present within each video clip, motivating a remapping.

Given a video we collate pixels from several frames sampled at equal temporal intervals, and apply colour quantisation in CIELab space to identify the set of dominant video colours $V = \{v_1, ..., v_q\}$ (where $q$ is variable but typically around 30).

Given a super-pixel $\mathcal{S}_t^i$, we produce a normalised histogram with respect to the video-derived palette $H_v(j), j = [1, q]$ by assigning each component pixel to the closest colour in set $V$. The colour descriptor $C(\mathcal{S}_t^i) = H_q(k), k = [1, 16]$ for the super-pixel (where $H_q$ is defined with respect to the query UI colour palette) is given by:

$$H_q(k) = \frac{1}{|V|} \sum_{j=1}^{|V|} H_v(j) d\left(h_c[k], h_v[j]\right). \tag{3.2}$$

where $d(.)$ is the normalised CIELab distance between colours corresponding to the $j^{\text{th}}$ and $k^{\text{th}}$ bins of $H_v$ and $H_q$. For measuring CIELab distances we apply the CIEDE2000 distance measure [187], which to provide a perceptually faithful colour comparison versus Euclidean distance. Although the use of CIELab is more relevant to the mapping from query palette to video palette than segmentation and dominant colour detection, we opt to use throughout. This is due to over quantisation in both cases, removing the concern of use of perceptually similar colour space vs another colour space.

### 3.2.3 Semantic feature extraction

Semantic features are extracted from video by labelling pixels in each frame independently, using Semantic Texton Forests (STF) [193] and aggregating these within the footprint of each superpixel $\mathcal{S}_t^i$ to form a frequency histogram across known object categories.

STF uses a Forest of Extremely Randomised Trees to classify pixels, these ensembles of decision trees are fast to train and test whilst their inherent randomness allows for flexibility to inter-class discrepancies.

The STF approach is composed of three classifiers: a standard ensemble of randomised decision trees; a global image classifier; and a second ensemble based on region information (Fig. 3.4). The standard ensemble of trees are trained based on CIELab colour value differences. A random point $p_{x_2,y_2,b_2}$ around a training point $p_{x_1,y_1,b_1}$ within a window ($width = 50$) is selected, where $x, y$ refer to location and $b$ refers to colour channel. The comparisons of values are based on a random comparison function value $p_{x_1,y_1,b_1}$ addition $p_{x_1,y_1,b_1} + p_{x_2,y_2,b_2}$ , subtraction $p_{x_1,y_1,b_1} - p_{x_2,y_2,b_2}$, absolute difference $|p_{x_1,y_1,b_1} - p_{x_2,y_2,b_2}|$. We also utilise $median_{p \in X}$

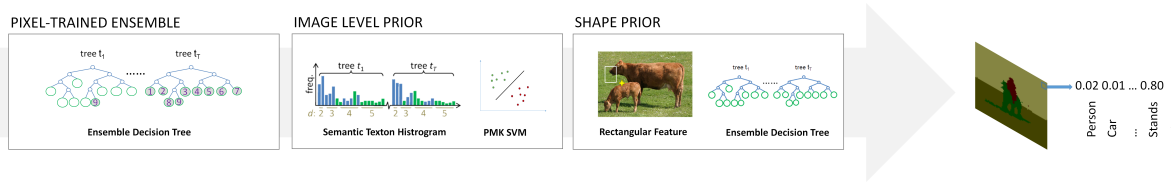



**Figure 3.4** Extraction of Per-Pixel Semantic Classification through the Semantic Texton Forest. Using a trained Ensemble of Extremely Randomised Trees on local pixel differences within a window pixels are classified, a global prior is defined through pyramid match kernel trained SVM. Finally a shape prior is trained as a second Ensemble of Extremely Randomised Trees using rectangular features. Following the method of Shotton et al. [193].

$median_{p \in y}$ as well as relative $xy$ positions to centre $\frac{p_{x_1,y_1} - c_{x,y}}{p_{x_2,y_2} - c_{x,y}}$, where $c_{x,y}$ is the coordinate of the centre of the image. In experiments these additions are beneficial due to salient objects commonly being in the centre of shot.

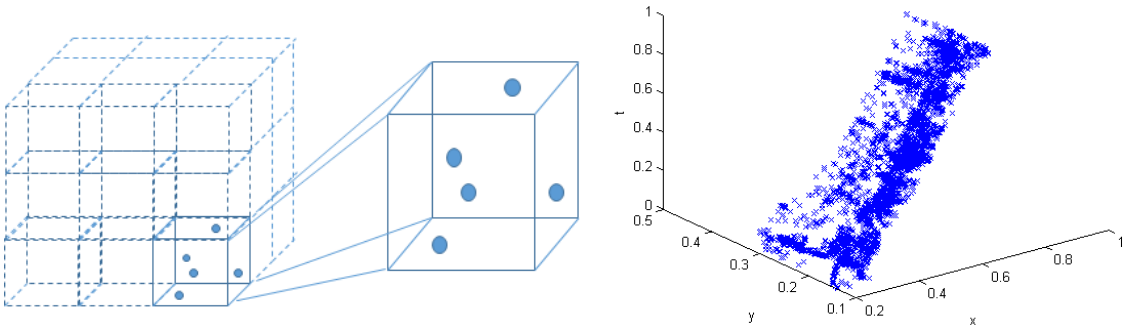
The global image classification is computed using an approach similar to Bag of Visual Words (BoVW). A hierarchical comparison of leaf distribution of the decision trees forms a Pyramid Matching Kernel for use within a one-vs-all SVM strategy. The third classifier, a second ensemble of decision trees trained on the resultant soft classified image. The probability image is sub-sampled and integral images are calculated allowing for a superpixel representation, a second ensemble is trained using rectangle feature is used based on window summation.

Each individual pixel is thus attributed a 12-D vector of probabilities describing the semantic content it depicts; this vector is averaged for all pixels within each superpixel $\mathcal{S}_t^i$ to yield $Q(S_t^i) \in \Re^{12}$.

Although it is possible to estimate such histograms directly over image regions using explicit region-based approaches [191, 125], their reliance upon complex filter banks at test time are currently prohibitive for scaling over a large video dataset e. g. comprising hundreds of videos. Current state-of-the-art execution times on $320x240$ footage are around 6 seconds/frame. By contrast the colour quantisation already performed in subsec. 3.2.2 enables significant implementation efficiencies in STF. Combined colour and semantic feature extraction take approximately one second per frame in our system.

## 3.3 Spatio-temporal Video Descriptor

Each video in the dataset is indexed by computing a descriptor from its spatio-temporal $(x, y, t)$ volume. The volume is subdivided equally into cells (Fig. 3.5a) at a coarse quantisation level resulting in $q \times q \times q$ cells. The choice of $q$ drives a speed-accuracy trade-off in the




(a) (b)

**Figure 3.5** (a) Spatio-temporal quantisation of the video clip into cells, each of which contains a number of foreground and background superpixels (each with associated colour and semantic attributes). (b) Illustrative point cloud distribution for the horse object within a horse-riding video clip. The spatio-temporal locations of the (camera motion compensated) super-pixel centroids describe the shape and motion of the objects.

retrieval algorithm and is explored in detail later (subsec. 3.5.2). A level of $q = 6$ is used in reported results.

Quantisation is performed over the full duration of the video i. e. $t = [1, T]$ and over the full height and width of the panorama formed by considering the alignment of video frames transformed under $\mathcal{H}$. Each superpixel $\mathcal{S}$ is assigned uniquely to a cell based on the location of the camera-motion compensated centroid $\mathcal{P}'$. Thus each cell is populated by a multitude of foreground and background superpixels, each of which has associated histograms encoding colour and semantic distribution.

Fig. 3.5b illustrates the distribution of foreground superpixels for a horse-riding clip within our dataset. Only super-pixels likely to correspond to the horse semantic category (i. e. with $\max Q(\mathcal{S}) = \text{Horse}$) have been plotted. The trajectory of the point cloud through the spatio-temporal volume implicitly describes both the shape and motion of the object over the course of the video. The associated feature vectors $C(.)$ and $Q(.)$ describe its colour and semantic attributes over time.

The complete video descriptor is formed by concatenating five sub-descriptors, each computed as follows:

1. **Foreground object colour**. Formed by concatenating $q^3$ 'cell descriptors' that each encode $C(\mathcal{S}^{FG})$ within the respective cell.

2. **Foreground object semantics**. Formed by concatenating $q^3$ 'cell descriptors' that each encode $Q(\mathcal{S}^{FG})$ within the respective cell.

3. **Background colour**. Formed by concatenating $q^3$ 'cell descriptors' that each encode $C(\mathcal{S}^{BG})$ within the respective cell.

4. **Background semantics**. Formed by concatenating $q^3$ 'cell descriptors' that each encode $Q(\mathcal{S}^{BG})$ within the respective cell.

5. **Background shape descriptor**. As outlined shortly in subsec. 3.3.2.

### 3.3.1 Cell Descriptors

To compute the descriptor for a given cell, we first identify the subset of super-pixels $s \subseteq \mathcal{S}$ with centroids falling within its spatio-temporal bounds. We then compute a normalised colour histogram from those superpixels:

$$H_c = \frac{1}{|p|} \sum_p C(s(p)). \tag{3.3}$$

The distribution of semantic attributes present in $p$ is similarly computed but not normalised; in the case of colour we are interested in the relative colour distribution over all points present, whereas with semantic attributes we are interested in the total evidence for each semantic category trained.

$$Q_p = \sum_p Q(s(p)). \tag{3.4}$$

### 3.3.2 Background Shape Descriptor

Weak information encoding the spatial structure of scene background is implicitly captured through the spatio-temporal distribution of $\mathcal{S}^{BG}$ within cells. However this has been derived from the centroids of super-pixels in an over-segmentation of the (often cluttered) background, which typically does not offer a usable representation of spatial structure. We therefore opt to encode an additional representation, computing the GF-HoG shape descriptor of Hu et al. [102] over a 'background panorama' created by aligning video frames under homography (via $\mathcal{H}$) and blending using a temporal median filter.

GF-HoG constructs a smoothly varying gradient field, interpolated over sparse edge points identified within the panorama using the Canny edge detector.

Given a binary mask of Canny edges $M(x, y)$ the orientation of each edge pixel $G(x, y)$ may be obtained via $\arctan\left(\frac{\delta M}{y}, \frac{\delta M}{x}\right)$. However this is defined only where $M(x, y) \neq 0$.

A dense field $\mathcal{G}_\Omega$ is derived over the full image coordinate domain $\Omega \in \Re^2$ constrained by $\mathcal{G}(p) = G(p), \forall_{x,y} M(p) \neq 0$, which minimises the Laplacian energy term:

$$\underset{\mathcal{G}}{\operatorname{argmin}} \iint_\Omega (\bigtriangledown \mathcal{G} - G)^2 \quad \text{s.t.} \mathcal{G}|_{\delta\Omega} = G|_{\delta\Omega} \tag{3.5}$$

with a discrete solution obtained by solving Poisson's equation with Dirichlet boundary conditions [160]. This can be approximated using linear equations to $\triangle \mathcal{G} = 0$ over a $3 \times 3$ grid, sampling the Laplacian of Gaussian operator:

$$\triangle \mathcal{G}(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2+y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.6}$$

Having obtained dense-field $\mathcal{G}$ the standard Histogram of Gradients (HoG) is computed local to all points $M(x, y) \neq 0$ at three resolutions. A hard-assignment Bag of Visual Words (BoVW) representation is used to reduce this set of HoG features to a k-dimensional frequency histogram, normalised via *tf-idf* after Sivic et al. [198]. The codebook for the BoVW process is computed over all HoG features sampled from the dataset, and a vocabulary size of $k = 1500$ was used following [102]. We write the resulting descriptor $A(\mathcal{S}^{BG})$.

## 3.4 Matching Query Sketches

We first outline how the query descriptor is extracted from the user-supplied sketch (subsec. 3.4.1) and then how the descriptor is matched against the database of ingested video (subsec. 3.4.2).

### 3.4.1 Construction of Query Descriptor

We employ a sketch parsing step similar to [52] to extract individual object shapes from the sketch. The method results in a set of regions corresponding to the background and each foreground object, with 2D trajectories across the canvas associated with the latter. Each segmented region has a colour distribution and semantic label associated with it by the user.

We construct a spatio-temporal descriptor from the query sketch, to enable direct comparison with the spatio-temporal descriptor of each video in the database, as follows.

We first synthesise the set of super-pixels $\mathcal{S}$ and feature points $\mathcal{P}$ from the sketched regions corresponding to foreground objects. We assume that a sketched object progresses linearly along its sketched trajectory, for the duration of the video, with the sketched position being the start position. This yields an idealised position for the object any relative time in the

video. When synthesising the position of the object, we use the coordinate mapping established between the sketch canvas and (constant width) camera-compensated video space to determine the region occupied by the object at each frame.

On this basis we synthesise a spatio-temporal representation of an 'ideal' video clip, generating $\mathcal{S}, \mathcal{P}$ progressively at each time instant in the 'ideal' clip. We cannot know the duration of this ideal clip, however this does not matter and can be arbitrary as we subsequently compute a descriptor $(q \times q \times q)$ spatio-temporal quantisation over the ideal clip duration — extracting a spatio-temporal descriptor as per (Sec. 3.3).

Background properties are extracted from colours, labels and shapes on the sketched background as per subsec. 3.3.2.

### 3.4.2 Linear Descriptor Matching

Given the common representation of the query and video spatio-temporal descriptors, matching can be achieved trivially via Euclidean ($L^2$) distance for each video descriptor in the database. Independently computing distances between the semantic and colour components (using the Euclidean and $\chi^2$ distances respectively) yields a performance gain of $\sim 10\%$. To retain the efficiency of computing a single norm between query and video descriptors, we borrow Arandjelovic and Zisserman's [8] technique of square-rooting each bin value (here, the colour histogram bins) to convert Euclidean distance within the colour sub-space to the Hellinger distance.

Sub-spaces of the descriptor could be rescaled (re-weighted) to reflect user preference $U_w$ for one modality (e.g. semantics) over another (e.g. colour); however in results presented user weights are set equally to $U_w = 1.0$ . Although this doesn't reflect an equal weighting of descriptor sub-spaces, it is seen to produce promising results.

### 3.4.3 Sub-linear Descriptor Matching

Linear search over video datasets of size $< 1k$ using the descriptor can be performed in real-time speeds, but with large datasets a matching process of complexity that scales sub-linearly with dataset size is essential. Following previous Information Retrieval techniques we opt to index our video descriptors using a *kd*-tree. The FLANN toolbox in OpenCV [232] is used in our implementation, and each modality or 'facet' of the video descriptor (e. g. colour, shape, etc.) is stored within a separate *kd*-tree.

At query time the forest of *kd*-trees are searched for the top N results. The top N results are then intersected with each other to form a short-list of relevant clips that are finally ranked

according to their distance as per subsec. 3.4.2. Producing a short-list for ranking relies upon an intersection process that in turn relies on commonality among results. We found $N = 350$ yielded approximately 50 common results for the final ranking step.

## 3.5 Evaluation

We describe evaluation of the proposed algorithm as a stand-alone (i. e. single iteration) retrieval system. The system is extended to incorporate iterative user-feedback via relevance feedback in Sec. 3.6.

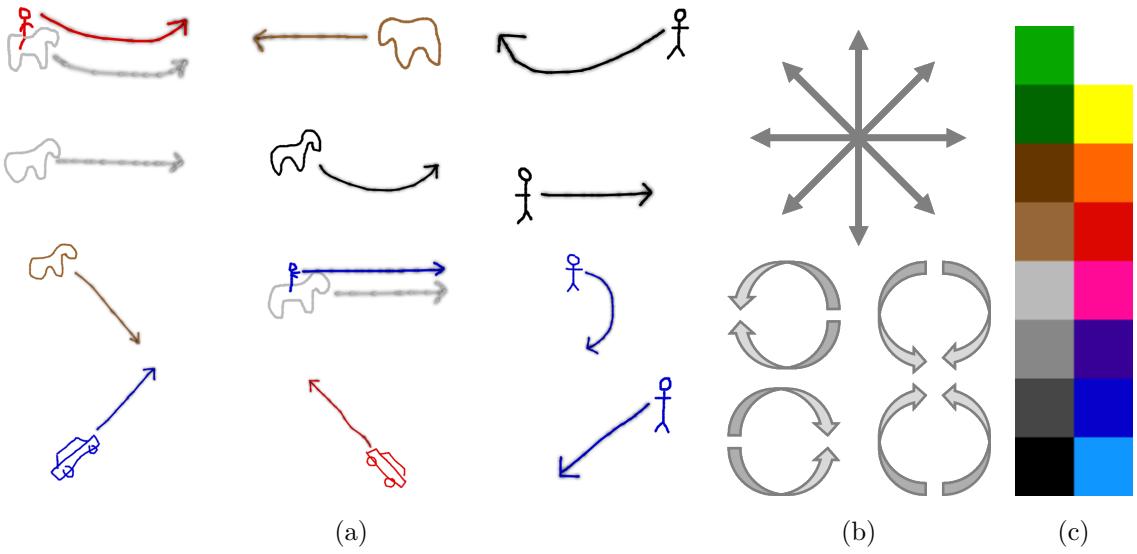
We first describe our evaluation dataset (subsec. 3.5.1) before exploring the accuracy-speed trade-off within quantisation parameter $q$ in subsec. 3.5.2. Performance is reported both in terms of query time (subsec. 3.5.3) and retrieval accuracy (subsec. 3.5.4) in terms of mean average precision (MAP) against our ground-truth.

### 3.5.1 TSF700 Dataset

The proposed system was evaluated over a dataset of 700 TV broadcast sports video clips (TSF700), following earlier work by Hu et al. and Collomosse et al. who evaluated their hybrid-SBVR systems over 500 and 298 such clips respectively. Hu et al.'s dataset was a super-set of the sports content with Collomosse et al.'s TSF (Television Sports Footage) dataset and was extended due to the consideration of semantic class in Hu et al.'s work. Our TSF700 dataset is in turn a super-set of Hu et al. incorporating a greater number of semantic object classes, namely: *person, horse, car, grass, snow, road, stands (audience), tree, obstacle, sand sky,* and *water*. The duration of each clip is 4-10 seconds at 25fps with a mixture of low resolution PAL (720x576) and HD (1920x1080) footage. Clips were selected that exhibited motions similar to the quantisation of fig. 3.7b with expressive colour. The groundtruth was derived through extending the groundtruth of Collomosse & Hu, to incorporate the extended: 10 classes, 16 motion directions and 10 colour palette. An overview is shown in Fig. 3.6.

For consistency in evaluation, a constant query-set comprising of 12 sketches covering 7 different object colour and 8 motion trajectories were produced. These are used throughout the evaluation. In order to establish a quantitative benchmark for performance, a ground-truth annotation was manually generated for the 700 clips and each of the queries. For each clip, a note is made for each modality covered by our system. A note is made of the dominant colours (closest query palette colours) and semantic labels associated with each moving foreground object in the clip. The direction of motion of the object is encoded in a quantised manner; linear motion in any of the 8 compass directions, or in 4 arced paths that cover all motion types in TSF700 (Fig. 3.7).

**Figure 3.6** A collage illustrating content of the TSF700 evaluation dataset comprising one keyframe from each of the 700 TV broadcast sports clips.



(a) (b) (c)

**Figure 3.7** (a) Query-set used to evaluate performance on TSF700. (b) Quantised motion directions used to encode ground truth for quantitative evaluation of TSF700. (c) The query colour palette.

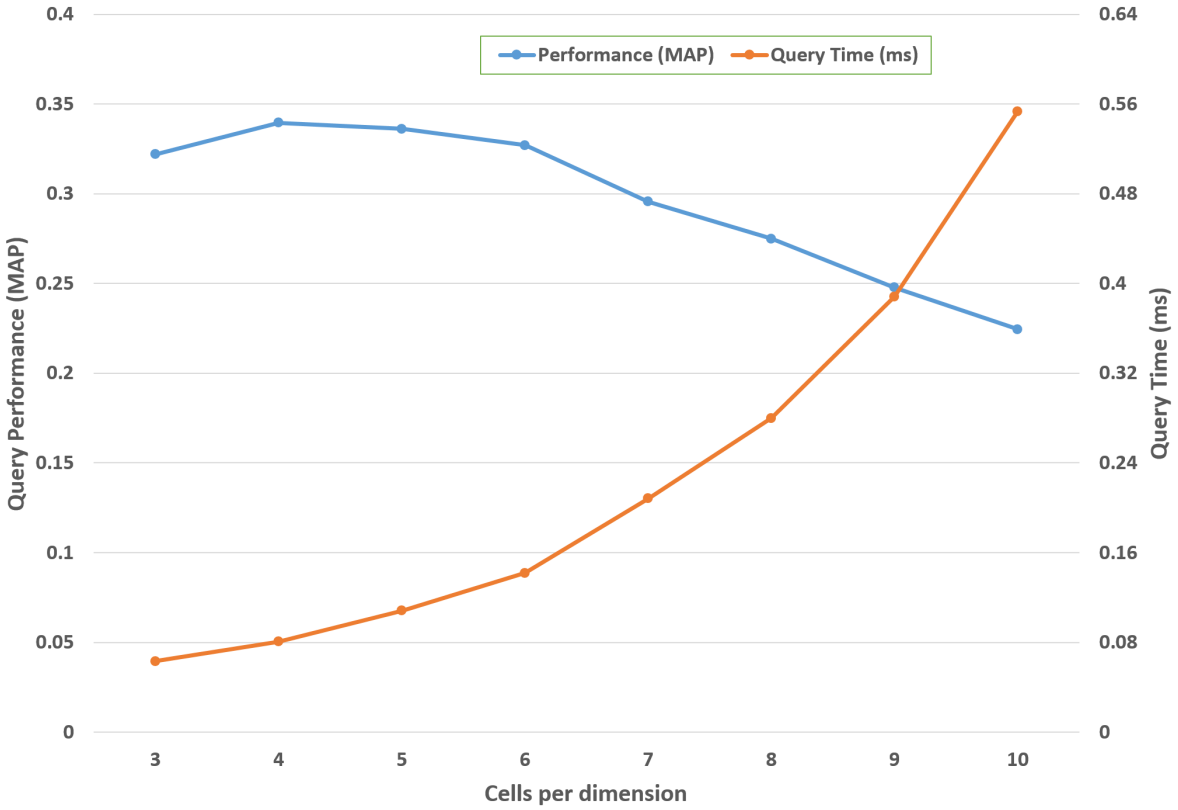



**Figure 3.8** Performance of SBVR as space-time quantisation is varied in range $q = [3, 10]$. Significant decline in accuracy is observed beyond $q = 6$ with query times becoming unacceptable (more than half a second) beyond $q = 10$.

Quantitative analysis of accuracy is performed by analysing the relevance of results using the standard measures of Precision and Recall, from which a scalar representing absolute performance (mean average precision; MAP) is derived. To determine the ground-truth relevance of a result, it is compared against the ground-truth markup for the query being run. A result will be defined as relevant for purposes of ground-truth, if the sets of colour and semantic labels for a query are both subsets of (or equivalent to) the sets of colour and semantic labels annotated to the result. The labels indicating motion direction on the cues must also match exactly. This protocol matches that of Hu et al. whom we later compare to [102]. The increasing combinatorial requirements for exactly matching each modality considered illustrates the challenge of achieving a high MAP for multi-modal (hybrid) SBVR.

### 3.5.2 Evaluation of Quantisation Level

The quantisation level ($q$) of the space-time video descriptor is a compromise between granularity of representation versus the query time. A high granularity results in a larger descriptor increasing search time. Fig. 3.8 evaluates both the effect of varying descriptor size on perfor-

mance in terms of precision (MAP), as well as the effect on query response time. Experiments are run over the full query set, averaging over all 12 queries. Decimation of the space time volume is varied between $q = [3, 10]$ in unit increments.

Fig. 3.8 quantifies the trade-off identifying the best performing quantisation level in terms of precision to be $q = 4 \times 4 \times 4$ for $x, y, t$ with an MAP of 34% and query time of $0.08ms$. Quantisation levels between 4-6 give negligible difference in query performance with a variation of 1% MAP.

Given only minimal drop in accuracy for a modest gain in query-time speed we opt for a level of $q = 6 \times 6 \times 6$. This constitutes a decrease of 1.2% MAP and time increase of $0.06s$, just before the turning point where MAP drops considerably as $q$ increases. It also intuitive to carry-forward a higher degree of granularity, where we later show how an increased precision can be achieved through relevance feedback integration(Sec. 3.6).

### 3.5.3 Query-times

Continuing the analysis of retrieval time we compare our proposed index-based approach to hybrid SBVR against prior approaches, which all adopt an optimisation (model fitting) approach. For fairness of comparison we adopt a linear search, comparing each dataset record to the query, rather than any sub-linear optimisation (e. g. via *kd*-tree). Both the trellis-distance [100] and Markov Random Field (MRF) [102] approaches of Hu et al., and the linear-dynamical-systems approach of Collomosse et al. [53] are compared using values given in their papers extrapolated to single and 700-element dataset timings.

| Method | Proposed | Collomosse [53] | Hu-Trajectories [100] | Hu-MRF [102] |
|---|---|---|---|---|
| Speed Per Clip (s) | 0.0002 | 0.24 | 0.02-0.03 | 0.10684 |
| Dataset Speed (s) | 0.14 | 120 | 17.5 | 74 |

**Table 3.1** Comparing query time of our proposed method under linear search vs. [53, 100, 102]. When using *kd*-tree our system is significantly faster still, searching the entire 700-clip dataset in fractions of a millisecond.

Table 3.1 summaries the results at $q = 6$. The proposed system is three orders of magnitude faster in comparison per clip than prior hybrid SBVR techniques taking on average 140ms to linearly search the entire dataset. As indicated by the timings in Fig. 3.8 the system is four orders of magnitude faster still when exploiting sub-linear complexity search via a *kd*-tree and It can be made fractionally faster again by tweaking $q = 4$. This demonstrates potential for significant scalability as dataset size increases.

As with prior approaches, the time taken to ingest a video is significantly larger than the per-clip query time. It can take several minutes (typically under 10 minutes) to ingest a

single video from TSF700 with our approach due to the large number of pre-processing steps to stitch and extract features from each video frame. This is slow but aligned with prior approaches that propose similar multi-stage pipelines for video pre-processing. In a future cloud-based deployment scenario such times would be practical via distributed computing.

Subsec. 3.4.3 presented a technique to scale the system to achieve sublinear query time. We evaluate varying N (the number of results returned from the forest of *kd*-trees), and the effect upon the performance of the system in terms of query-time speed and number results returned. Accuracy is not evaluated since the results returned do not vary from the linear case already examined. Table 3.2 reports average results of 100 query sketches.

| N | 200 | 250 | 300 | 350 | 400 |
|---|---|---|---|---|---|
| Results returned | 6 | 16 | 29 | 51 | 82 |
| Mean query time (s) | 0.0029 | 0.0034 | 0.0042 | 0.0051 | 0.0059 |

**Table 3.2** Investigating the effect of parameter N, comparing both the performance (speed) and number of results returned at query-time. Setting N=350, query time is 27x faster than simple linear search yet a sufficient number of results ($\sim 50$) are returned to the user for browsing.

### 3.5.4 Retrieval Accuracy

Fig. 3.9 quantifies the accuracy of the proposed system at $q = 6$ over the TSF700 dataset. A mean average precision (MAP) of 35% is achieved using matching semantics shape and motion cues alone – falling to 32% when colour is also incorporated. A similar small drop in performance is reported in other hybrid SBVR systems [102] due to the increased difficulty in accurately matching across all query modalities.

Fig. 3.10 illustrates representative queries of foreground objects and the corresponding top 5 clips returned. We observed that shapes augmented with motion cues alone are easier to match. Queries containing the *car* semantic class were the most challenging, due to the difficulty of the semantic segmentation algorithm (STF, subsec. 3.2.3) is identifying this class. This is likely due to the reliance of STF on colour appearance which exhibits high intra-class variation for this class. Nevertheless opting for this algorithm enables us to extract semantic features in just under 1 second per frame, this constitutes a saving of over one week of time on video ingestion for our 700 clip dataset and brings us in line with existing hybrid SBVR approaches [100, 102].

For the purpose of comparison with the state of the art we align our ground-truth to the more permissive methodology of Hu et al. [102] (matching identically on colour and motion but on motion only in the 4 major compass directions). We compare using their 500 clip dataset using 7 queries matching those in their paper. We achieve MAP of 30% versus their

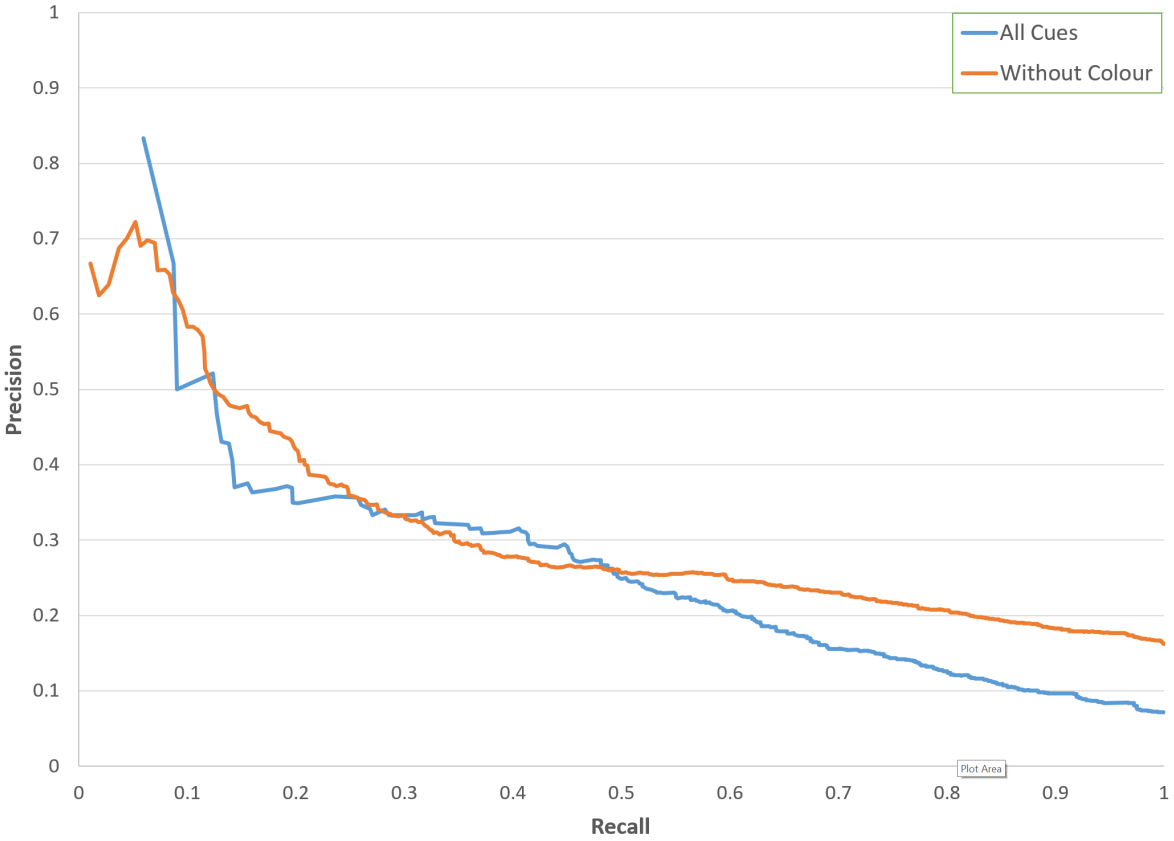



**Figure 3.9** Precision-Recall curve characterising performance (accuracy) for the proposed hybrid SBVR system.

reported 48%. Our accuracy is lower under this methodology, however our approach is 3 orders of magnitude faster per clip and can scale sub-linearly whereas [102] scales linearly and comprises an expensive matching function that is already intractable for interactive retrieval, taking over a minute for our dataset. We later discuss (Sec. 3.6) how our accuracy can be raised beyond that of Hu et al. via relevance feedback whilst retaining the scalability afforded by the lower query-time complexity [102].

Structural (shape) information is a novel modality not addressed extensively in SBIR but omitted from previous hybrid SBVR systems [101, 102]. Fig. 3.11 demonstrate the results of an representative query that depicts stylised background structure. We visualise the video panorama generated for the best matching clip and overlay the query sketch for illustration.

## 3.6 Relevance Feedback

The significant performance benefits of an index-based matching approach for SBVR are near-instantaneous full database search over hundreds of videos (Sec. 3.5). This raises the opportunity of working with the user 'in the loop' to interactively refine results. After candi-

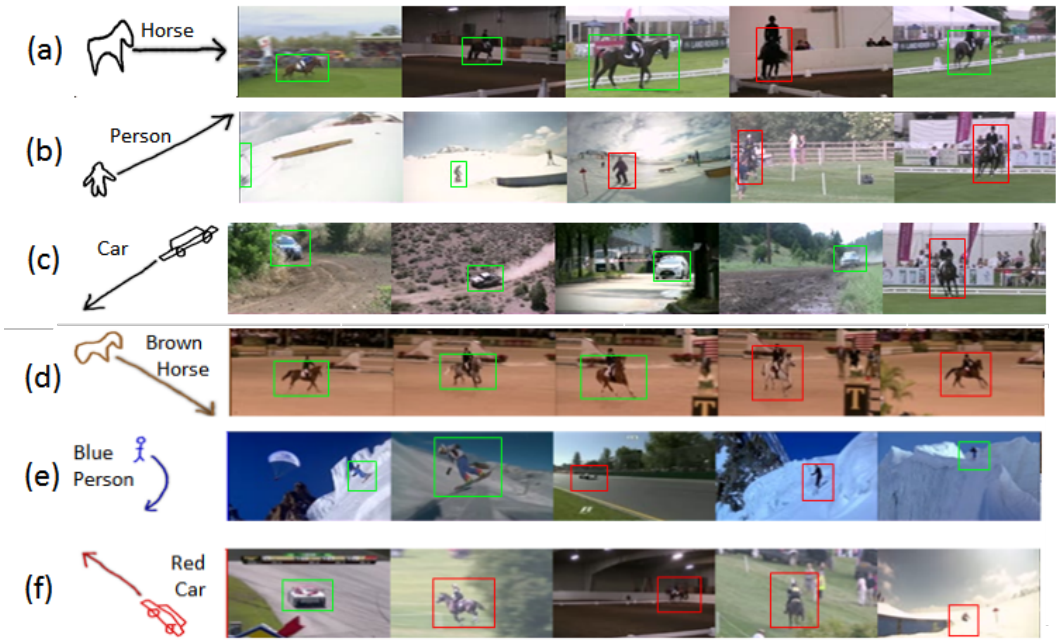



**Figure 3.10** Five representative hybrid SBVR queries and top 5 results: (a)-(c) specify semantics and motion, (d)-(f) specify semantics, colour (as shown as keyword for clarity) and motion. Shape is specified implicitly by the sketch in both. Relevant results in green, irrelevant in red.

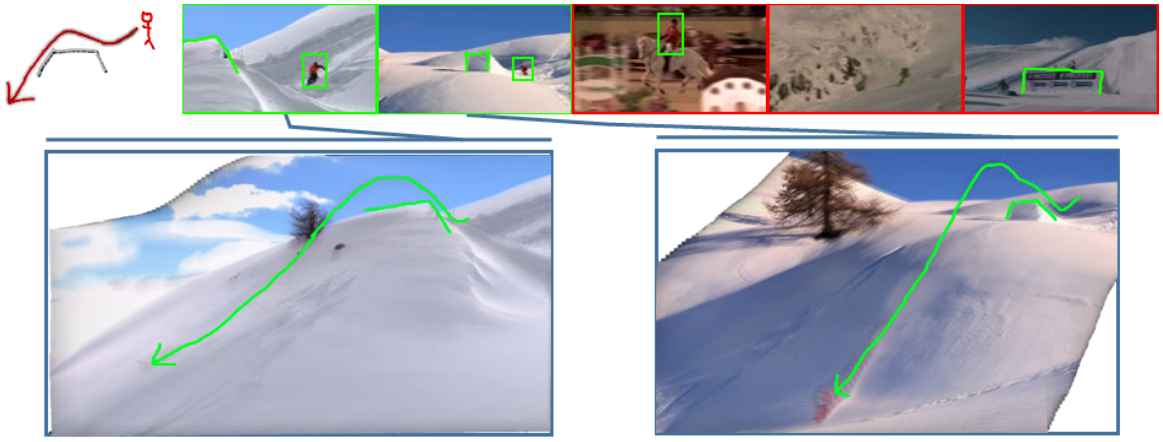


**Figure 3.11** Top: Query incorporating background, and top 5 ranked results; the first 2 ranked results are relevant. Bottom: Video panorama depicting correctly retrieved object motion.

date results are presented via our initial matching process (subsec. 3.4.2), the user is invited to label results indicating a few positive (relevant) and negative (irrelevant) examples. Results are then re-ranked using this input; a process referred to as Relevance Feedback (RF).

In classical information retrieval, RF is implemented by training a linear SVM within the descriptor space, using the relevant and irrelevant results labelled by the user. The distance from the SVM decision boundary constitutes the re-ranking function. However unlike classical contexts, our hybrid sketches exhibit multiple modalities (or 'facets'; namely colour, shape,

motion, semantics, and background structure) which under such formulation would imply such a distance function to be dependent on some *a priori* specified weighting between the various modalities.

We explore how RF can be applied over multiple feature types using an ensemble of classifiers, one per facet. We then evaluate the updated system under the methodology of Sec. 3.5.

### 3.6.1 Classifier Ensemble based Relevance Feedback

Multiple Classifier Learning (MCL) is a popular approach in feature weight selection. The approach works by training a classifier on an individual feature channel. For each channel a singular Support Vector Machine (SVM) is trained using the sub-space of the descriptor $x_i \in^n, i = 1, \ldots, l$ and label vector $y \in^l$ such that $y_i = \{1, -1\}$ where the traditional function

$$\min_{\omega} \quad \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}(\max(0, 1 - y_i\omega^T x_i))^2 \tag{3.7}$$

is optimised. In the case of RF, each facet's classifier is trained within the sub-space of our spatio-temporal descriptor relevant to the facet; e. g. the colour components of our descriptor form a sub-space in which the classifier for the colour facet is trained. Each facet's classifier takes the form of a linear SVM ($\mathcal{M}_i$) trained using the marked-up relevant and irrelevant results. A confidence weight $\mathcal{C}_i$ is assigned to each facet's SVM, estimating the discriminative power of that facet for the current query. For each facet, samples specified as positive or negative become the training set $\mathcal{X}$ with their respective labels $\mathcal{Y}, \in [-1, 1]$. The trained SVM $\mathcal{M}_i$, then yields weight $\mathcal{C}_i$ as:

$$\mathcal{C}_i = \frac{1}{n}\sum_{j=n}^{j=1}\begin{cases} 1 \text{ if } \operatorname{sgn}(S(\mathcal{M}_i(\mathcal{X}_j))) = \mathcal{Y}_j \\ 0 \text{ otherwise.} \end{cases} \tag{3.8}$$

Where $S(.)$ normalises the kernel score using a sigmoid function:

$$S(\mathcal{M}_i(\mathcal{X}_j)) = \tfrac{1}{1+\exp(Af+B)} \tag{3.9}$$

where $A$ and $B$ are optimised by a sigmoid fitting function using the method of Platt [167].

The position of a video in the re-ranked results is simply the product of each facet's SVM decision function $S(\mathcal{M}_i)$ and the confidence in that facet being discriminatory (useful) for the query in hand, determined automatically $\mathcal{C}_i$ and via the user defined weighting $U_{w_i}$ of Sec. 3.4.2.

On each iteration of relevance feedback, the user-supplied relevant and irrelevant examples augment the training set, and $\mathcal{M}$ are retrained. During evaluation we found that presentation of only 10-15 results to be necessary to achieve significant improvement within a couple of iterations of relevance feedback (Sec. 3.7.1).

## 3.7 Evaluation of Relevance Feedback

The number of results the user is asked to mark up as feedback at each iteration of RF represents a trade-off. On the one hand, the greater the volume of mark-up captured each iteration the more information we have to train subsequent iterations (hopefully leading to closer alignment of results with user expectation). On the other hand, there are practical limits to the utility of a system that requires significant volumes of user interaction over several iterations.

We therefore explore the impact of varying the number of results presented (referred to hereafter as $N$) vs. the accuracy of the retrieval, for a fixed number of iterations (subsec. 3.7.1). We then explore, for a fixed $N$, the impact of number of iterations on final accuracy, i. e. at what point the presentation of additional RF iterations to the user has diminishing returns (subsec. 3.7.2).

### 3.7.1 Number of user indications per RF iteration

Over a fixed number of RF iterations (4) we ask users to indicate whether a number $N$ of results are relevant or irrelevant. Different runs of the experiment are conducted for four values $N = \{5, 10, 15, 20\}$ capped on the basis that in practice few users will be prepared to annotate more than 20 results per iteration.

For each experimental run we execute 12 searches using each sketch in our query set. The MAP is evaluated as per Sec. 3.5 and reported in Fig. 3.12.

The MCL framework delivers increased accuracy (MAP) as $N$ increases. Interestingly if too few results are marked up, overall accuracy decreases due to the under-informed MCL learning a model of relevance inconsistent with the user requirements. At least $N = 10$ results must be marked up each iteration to reap consistent benefit from RF, with absolute MAP increases of 10%, 17%, and 22% for $N = 10, 15, 20$ respectively over four iterations (with $N = 20$ peaking at 24% at 3 iterations).

Note that at each iteration of training the MCL includes data points fed back from prior iterations, so at iteration 4 the MCL would have 80 data points in the $N = 20$ configuration.

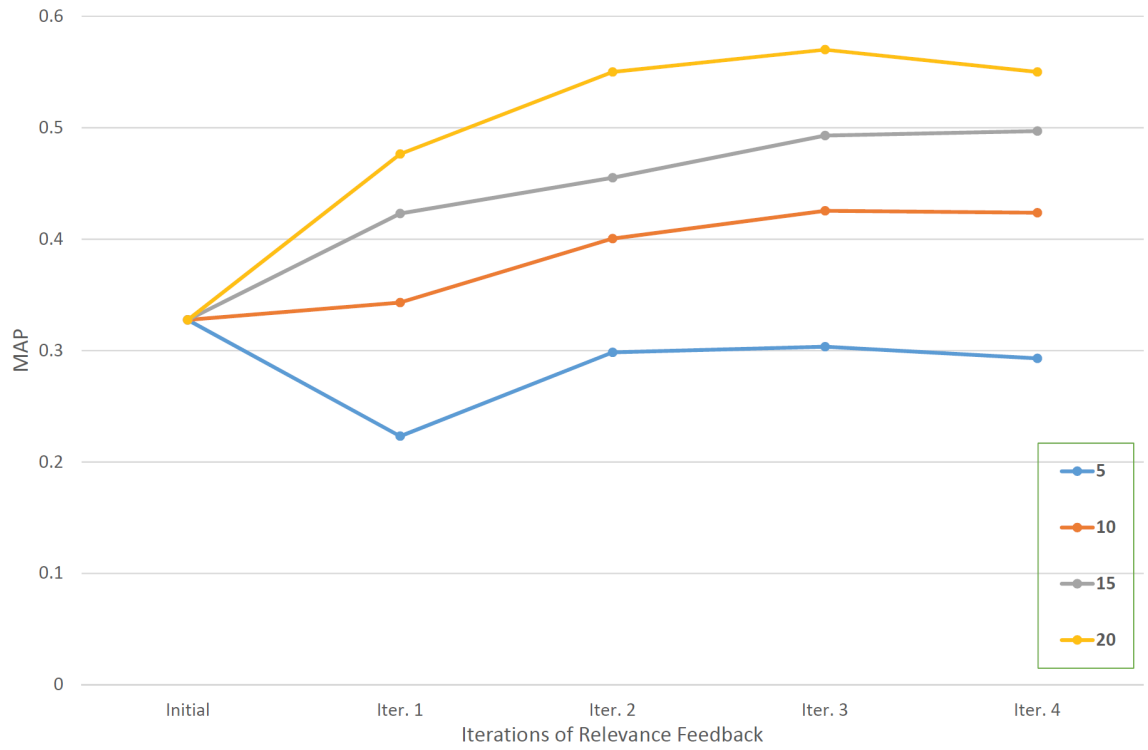



**Figure 3.12** Plotting retrieval accuracy vs. the number of relevant / irrelevant results marked up by user per RF iteration (for up to 4 iterations).

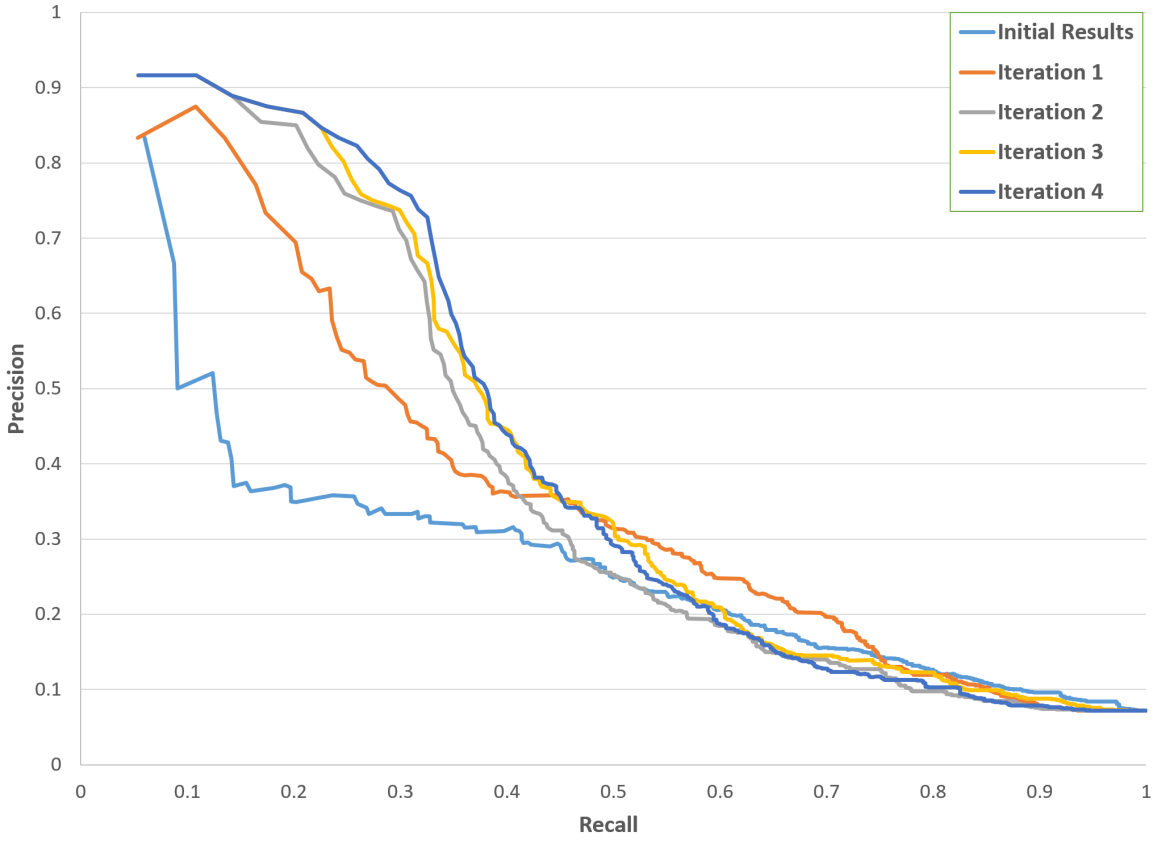



**Figure 3.13** Comparison of iterations for relevance feedback using Classifier Ensemble

On our dataset of 700 clips, this may lead to over-training and the observed decline in MAP at the final iteration for this configuration.

For subsequent evaluation of RF we opt for $N = 15$, providing a compromise between performance gain over user load and the risk of over-training.

### 3.7.2 Accuracy gain due to Relevance Feedback

To characterise any improvements in accuracy due to RF, a precision-recall (P-R) curve was plotted at each iteration of RF for $N = 15$ (Fig. 3.13). The curve is generated as per Sec. 3.5 using an average of all 12 sketches in the query-set with precision and recall computed over the top 1-700 ranked images. Table 3.3 also records the MAP for the system with and without consideration of colour when evaluating against the ground truth. As observed in evaluation of the non-RF system (Sec. 3.5) the latter case is slightly easier and achieves a higher MAP, confirming the observations of Hu et al. with respect to the challenges of multiple modalities.

Fig. 3.13 demonstrates the significant performance benefits achievable with just one iteration of RF. A gradual improvement from 32% to 50% MAP is observed over 4 iterations of feedback, for the full TSF700 dataset. In general only a couple of iterations are required to significantly improve the precision of the top few tens of results, as indicated by the shape change on the left of the P-R curve. Negligible performance gain is achieved between iterations 3 and 4, indicating that further user feedback would not warrant the additional user load beyond around 3-4 RF iterations.

The improvement over the top 10 results for the 4 iterations over a couple of representative queries is illustrated in Fig. 3.14a, which depicts two of the more challenging queries. For example, the query sketch containing the car was earlier highlighted (Fig. 3.10) as problematic in the non-RF system, demonstrating that it is possible to overcome the challenges of a correct semantic segmentation during pre-processing, recovering 4 in 5 correct results from an original 1 in 5 in this difficult case.

As in Sec. 3.5 we compare our RF based SBVR system to the state-of-the-art hybrid SBVR system of Hu et al. [102] by adapting our definition of ground-truth and restricting our query set to Hu et al.'s 500 clip dataset. Note that Hu et al. do not perform any RF.

Applying our RF approach yields 30.6%, 38.0%, 46.4%, 50.7%, and 52.8% for the pre-RF and RF iterations 1,2,3, and 4 respectively on this dataset. We therefore exceed Hu et al.'s reported MAP of 48% by $\sim 5\%$ absolute MAP. Their MAP is exceeded after 3 RF iterations.

Over TSF700 dataset for one iteration of RF the system takes 0.2 seconds for the results to update, based on the MCL method. A more appropriate measure of retrieval time is a

product of the RF iteration count with the sum of user-interaction time and the retrieval time of the non-RF based system (reported in Sec. 3.5.3). Interaction time varies highly among user or user task. Although some results can easily be discounted, based on representative thumbnails shown the user, e. g. colour of object, for ambiguous clips it can be required to play the clip, which leads to a significantly increased cumulative retrieval time. User interaction time could be improved by alternative clips representation, e. g. video summarisation.


(a)


(b)

**Figure 3.14** Representative results for two queries in the test set, visualising the top 10 results over four iterations of RF (relevant results in green, irrelevant in red). Significant precision improvement is shown within the top 10 using MCL based RF, with just a few user interactions.

| | Original | Iter. 1 | Iter. 2 | Iter. 3 | Iter. 4 |
|---|---|---|---|---|---|
| All Modalities | 32.1% | 42.3% | 45.5% | 49.3% | 49.7% |
| Without Colour | 35.0% | 33.5% | 42.9% | 50.8% | 54.4% |

**Table 3.3** Mean average precision (MAP) of the SBVR system with RF over TSF700, averaged over the full (12) query set. MAP has been computed with and without consideration of colour correctness in the ground-truth enabling comparison with [102].

| | Original | Iter. 1 | Iter. 2 | Iter. 3 | Iter. 4 |
|---|---|---|---|---|---|
| Hu [102] | 48.0% | * | * | * | * |
| Proposed | 30.6% | 38.0% | 46.4% | **50.7%** | **52.8%** |

**Table 3.4** Comparison to Hu et al. [102] via MAP using their 500 clip dataset and evaluation methodology.

## 3.8 Conclusion

We have proposed a novel descriptor for indexing video for sketch-based video retrieval (SBVR). The descriptor encodes the colour, shape, motion, and semantic class of foreground objects as well as shape in the scene background. We have applied this descriptor to build a novel SBVR system accepting query sketches that encode information in these modalities; a so called 'hybrid' SBVR system. In doing so we have advanced the state of the art in hybrid SBVR in two main ways:

1. **Constructing the first index-based hybrid SBVR system.** Prior hybrid SBVR work adopts a model-fitting approach to search, fitting the sketch to evidence in each clip within the dataset. By extracting video descriptors from clips and the query sketches and comparing these using a standard distance measure (e. g. $L^2$ norm) we can search a dataset containing several hundred videos in milliseconds, rather than in minutes. Furthermore using scalable index representations such as *kd*-tree we achieve the same in fractions of a millisecond (0.008ms for 700 clips). Such representations scale sub-linearly with dataset size and so are *prima facie* more scalable than the linear searches adopted by model-fitting approaches.

2. **Applying relevance-feedback (RF) to SBVR for the first time.** The real-time retrieval speeds delivered by (1) enable RF to be applied to hybrid SBVR for the first time (and more generally to SBVR for the first time). This enables user-in-the-loop refinement of search results, leading to near-doubling of MAP from 32% (without RF) to 50% (with RF); the latter representing the highest MAP recorded to date for hybrid SBVR over the largest dataset so far considered (700 videos). User refinement is important for hybrid SBVR because the relative importance (weightings) between the different modalities: colour, semantic class, motion, etc. are not encoded within the sketch.

A secondary contribution of this work is the development of the TSF700 dataset; a super-set of the current largest SBVR dataset due to Hu et al. comprising 500 videos.[1].

We compared our approach to the current state of the art in hybrid SBVR due to Hu et al. [102] who report 48% MAP over their dataset. We achieved a MAP of 30.6% without RF, but exceed their MAP at 50.7% after 3 RF iterations. We achieve up to 52.8% MAP with 4 iterations. In all cases our retrieval system operates instantaneously, whereas Hu et al. report several minutes to perform the retrieval which is not scalable or practical for most use cases. Although placing the user-in-the-loop with RF provides an advantage over a purely

[1]This data has been released publicly at `http://cvssp.org/projects/sketch/cvmp14`

automated approach, with hybrid SBVR it is desirable to do so to help overcome ambiguities in depiction and priority of modalities within the sketch. Our experiments also confirmed Hu et al.'s observation that maintaining high MAP scores as additional modalities are added to the system becomes more challenging.

### 3.8.1 Future Work

One issue raised during this study is the role of shape. Our system implicitly encodes and matches on object shape as one of several modalities in the sketch, yet within our evaluation methodology we do not consider it when quantifying performance against the ground-truth. The descriptor implicitly averages any shape variation present over the duration of the sequence — an approach that seems intuitive since (in the work of this Chapter) only a single sketch of the object may be provided as a query. Consequently any intra-class variation would need to be quite pronounced, which is atypical in the case of people, horses, and racing cars within TSF700 where any such variation is dwarfed by changes in shape e. g. due to the gallop of the horse, or orientation of a person on a snowboard. Indeed quantifying the similarity in shape within any video (representing an integration of shape information over time) versus a sketched query (representing a single instant in time) is both challenging to automate and highly subjective. This problem would be further exacerbated when considering general video 'in the wild'. We return to the issue of shape directly in subsequent chapters where we specialise our problem domain to video of dance performance, containing a single class of object (person) but exhibiting considerable variation in shape (pose) and in motion.

It is likely that a more robust method for extracting semantic features from the video would lead to higher MAP, as some queries (especially those containing cars) tended to produce lower accuracy results due to mislabelling of pixels in this semantic class. The STF method implemented was chosen for speed and simplicity, and is not representative of the state of the art. More sophisticated methods would slow video ingestion, but could enhance performance through classification of the video at the super-pixel level, or through consideration of cues other than colour. Labelling could also be propagated over time to improve temporal consistency. Due to the focus of this work on SBVR we treat this as a component of the pipeline, an easily substitutable black-box process.

MCL is just one of many feature selection methods that could have been explored for RF. The process of training and testing using the multiple classifiers often comes at high computational cost, yet despite these issues it may be worthwhile to explore Multiple Kernel Learning approaches as an alternative. The combination of kernels encoding the multiple facets into a singular classifier could simplify re-ranking of results since the ranking is then based on a singular distance from decision boundary.

# Chapter 4

# Sketch based Pose Retrieval and Estimation

This Chapter presents an algorithm for searching videos of dance performance for instances of a given human pose. The desired pose is specified using a free-hand sketch, which is parsed into a feature descriptor encoding skeletal joint angles. The skeletal descriptor is matched against image descriptors extracted from a video by analysing bounding boxes containing detected performers within each frame. To facilitate this matching, a mapping is learned between the manifold of likely poses occurring in both the skeletal and image descriptor spaces. The mapping is derived from geodesic distance defined over piece-wise linear models of the manifolds and is learned by training the system with sketches of exemplar poses sampled from a training video. To enable the visual search to generalise well over multiple videos, domain adaptation is used to transfer global statistics of the image descriptor manifolds occurring in the training and test videos.

## 4.1 Introduction

Video repositories are dominated by footage of people, documenting the events or performances to which they contributed. However, very little existing SBVR work tackles the issue of retrieving video content in which people are the primary subject. Prior work studying 112 free-hand sketch depictions of 10 video events drawn by 14 individuals reported that people were represented as pictograms (stick-men) 84% of the time, with exceptions being portrait close-ups [52]. That initial user study indicated the shape of a person in the video to be frequently depicted by the pose of the stick-man. Although an SBVR system was not an outcome of that work, it indicates the potential for an SBVR system that exploits pose information within sketched stick figures.

The previous Chapter proposed a coarse representation for capturing intra-class shape variation in general video (spatio-temporal occupancy within a video volume). That approach was capable of discriminating intra-class shape variation averaged over the duration of the clip. In this chapter we focus specifically on the issue of shape, proposing a per-frame (rather than per-clip) approach to visual search that enables more granular localisation of particular time instants (frames) within one or more videos containing the sketched shape. Recognising that people are frequently the subject of video, we focus on the specific problem of *human pose search using sketch*. The semantic gap in this context is particularly challenging, given the significant difference in appearance between a person in real video and the stylised depiction of a human as a stick-man in the query sketch.

We have focused on the domain of contemporary dance, given the rich variations in human pose expressed within videos of dance performance. Such a focus is timely as the performing arts are increasingly turning to digital archives for online dissemination of videos containing performances of historic note. Dance in particular has launched several major online archives of historic dance footage, including the EU GAMA and Digital Dance Archives (DDA). The latter hosts several collections curated within the UK National Resource Centre for Dance (NRCD) at Surrey, providing a convenient dataset for our research. As with most online video repositories, search technologies within these archives are predominantly text-based and focus on archival metadata rather than the visual content (i. e. the choreography) itself.

Sketch based pose search has not been previously investigated for SBIR/SBVR, however initial investigation of the pose search problem for video-realistic query has been undertaken within the wider field of Computer Vision [74, 109]. Such 'Pose Search' systems explicitly perform human pose estimation (HPE), either making an inference of the skeletal joint angles directly from an image ('hard' estimation [172]), or labelling pixels in an image with a probability vector representing the confidence that the pixel belongs to a particular limb ('soft' estimation [74]). This inferred data forms a pose descriptor for matching during the search.

Explicit HPE on general imagery 'in the wild' remains an open research challenge, and current approaches operate only with reasonable precision under constrained conditions. In particular leading hard HPE approaches operate reliably only in upright postures, making them unsuitable for dance and many sports. Soft HPE carries a high computational overhead due to the need to match over multiple channels (body part probability distributions estimated independently). Both classes of technique require high-resolution and relatively low-noise imagery to yield a robust pose estimate. By contrast, much of the dance video used within our work is low-resolution archival footage exhibiting contrast bleaching, motion blur and ghosting due to poor or varying illumination during capture, and multiple transfers between analogue and digital media over the years. Frequently footage is shot with a hand-held static camera, zoomed out from the performance to capture the entire stage. Consequently perform-

ers are infrequently more than 100 pixels high, and out of focus with limited textural detail available for feature extraction. In short, the use of HPE on such footage is inappropriate (also reported in [174]) motivating an alternative basis for pose search.

### 4.1.1 Dataset of Archival Dance Footage

We discuss and evaluate the algorithm proposed in this Chapter using a dataset comprising four videos of contemporary dance, using both low and high fidelity footage, licensed for research purposes from the UK National Research Centre for Dance (NRCD). Videos were selected based on having a singular performer, with no props within the sequence. Using this criteria a set of low and high fidelity videos were identified. Although more clips may satisfy this criteria within the NRCD archive, the selected clips exhibit a wide variety of pose, with varying visual characteristics to contend with.

For low fidelity footage we use archival performances of "Blueprint" and "Three Dances" (ThreeD) from the 1970s Extemporary Dance theatre collection, available within the Digital Dance Archives (DDA) repository. Both performances are digitised from footage originally shot on Cinefilm at 25fps, of duration 5:18 and 2:49 minutes respectively. The footage is PAL ($720 \times 576$) resolution. Challenging features of this footage are its grainy, low contrast nature and heavy motion blur.

Our samples of higher fidelity footage, are "Autumn" and "Expressive" from the Natural Movement series within the Extemporary Dance Theatre collection. These are sourced from a MPEG4 video of the dance performances, shot digitally at 25fps, of duration 2:05 and 6:40 respectively. The video resolutions are PAL and HD ($1920 \times 1080$) respectively.

All videos were encoded as interlaced footage. A sample of frames is shown in Fig. 4.1, highlighting the challenges present in the footage including background noise, lighting, motion blur and camera motion. In this work Blueprint is used as a training video, therefore performing preliminary evaluations on this clip. In Sec 4.4.1, Blueprint is additionally used to provide the correspondence between Sketch descriptors and Image descriptors for the manifold retrieval.

The pose search algorithm proposed within this Chapter is driven by supervised machine-learning, requiring an *a priori* manual training step. In our work a few hundred frames from the performance of Blueprint are used for training, with the remaining frames from Blueprint and the three other videos used as test data.

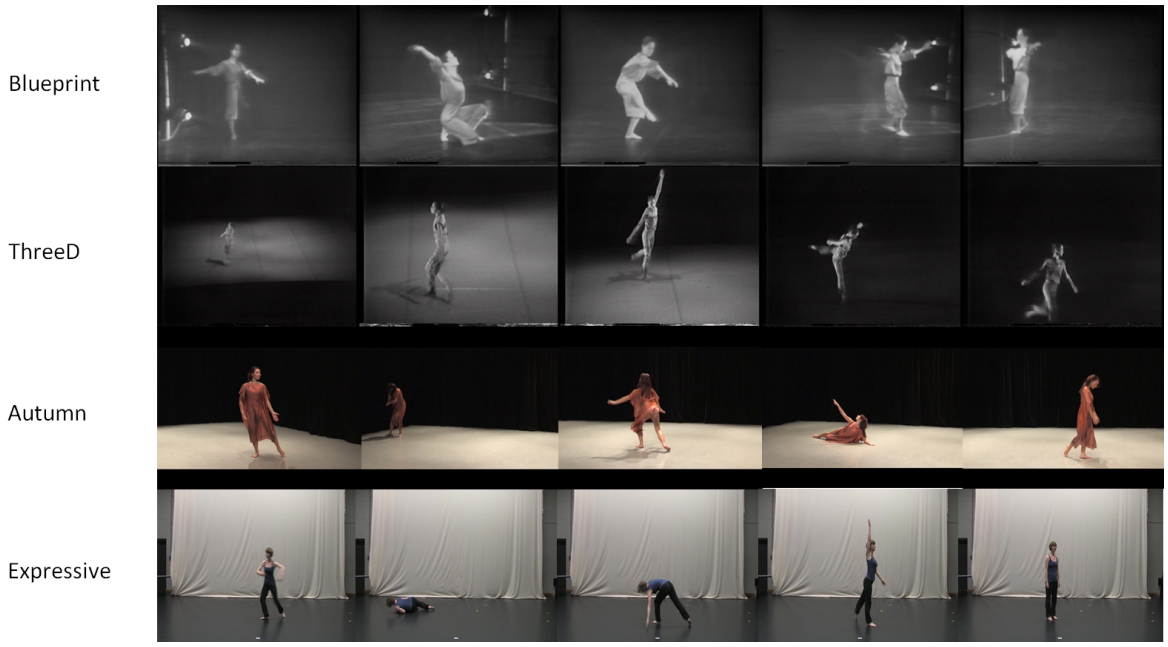



**Figure 4.1** Example frames from our dance video dataset demonstrating soft edges, lighting, camera motion and other artefacts present in this challenging footage.

### 4.1.2 Overview and contributions

The core contribution of this Chapter is the first SBVR system for pose search, driven by sketched 'stick-men' that depict the target pose to be located within the video dataset. The key technical contributions are:

1. **Pose search via a learned mapping between sketch space and image space.** We propose two descriptors, extracted from a free-hand sketch and a video frame respectively. We describe a supervised training method for constructing a manifold within each descriptor space describing the plausible space of human pose. We establish a mapping between the two using the training data, enabling matching of a sketched pose to video and thus pose search.

2. **Domain adaptation for multi-video search.** Since the manifold of plausible human poses can vary significantly between videos, a refined approach enabling the mapping of one video manifold to another is proposed to enhance retrieval accuracy. An auxiliary method based on HPE is used to automatically establish correspondence between the manifolds. The correspondence exploits video temporal coherence to ensure robustness to the high error rate typical of HPE over archival video footage.

## 4.2 Video Description

We begin by considering the design of a descriptor that implicitly captures the shape (full body pose) of performers within each video frame.

Since performers in our dataset are typically devoid of any rich texture information, appearance based methods e. g. based on gradient features typically fail to detect any useful signal within the raw frame containing the person. We therefore employ a salience measure that fuses several visual cues to extract the silhouettes of any performers present within the frame (Sec. 4.2.1). The resulting set of bounding boxes, each containing a silhouette, form the document set indexed by our SBVR system. A shape descriptor is computed from each bounding box (document) to construct this index (Sec. 4.2.2).

### 4.2.1 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 or colour models) that may succeed on higher resolution, comparatively noise-free footage. To adequately handle diverse, general footage we propose a machine learning approach that adapts to the appearance characteristics of the footage. The adaptive approach is presented in fig. 4.2. In the case of high-resolution footage e. g. "Expressive" the approach adapts for efficiency, using GMM based approach [118] as avoiding the expensive texton computation over large frame sizes. We outline the learning of the texture based appearance model (sec. 4.2.1) and the application (sec. 4.2.1).

#### Learning Performer Appearance

For each low resolution video clip, we learn an appearance model for texture likely to represent the performer. We apply a bank of Gaussian filters to all frames, to collect Texton features (c.f. Sec. 3.2.3) and perform coarse vector quantisation of this feature space using $k-$means clustering ($k = 100$). Each pixel is thus assigned to one of $k$ codewords, and a texton descriptor can be computed as the normalised histogram of codeword occurrence within a given spatial window.

We train a support vector machine (SVM) with positive and negative examples of performer texture from the clip. The features used to train the SVM are Textons computed within $10 \times 10$ windows from areas likely to be, and not to be, the performer.

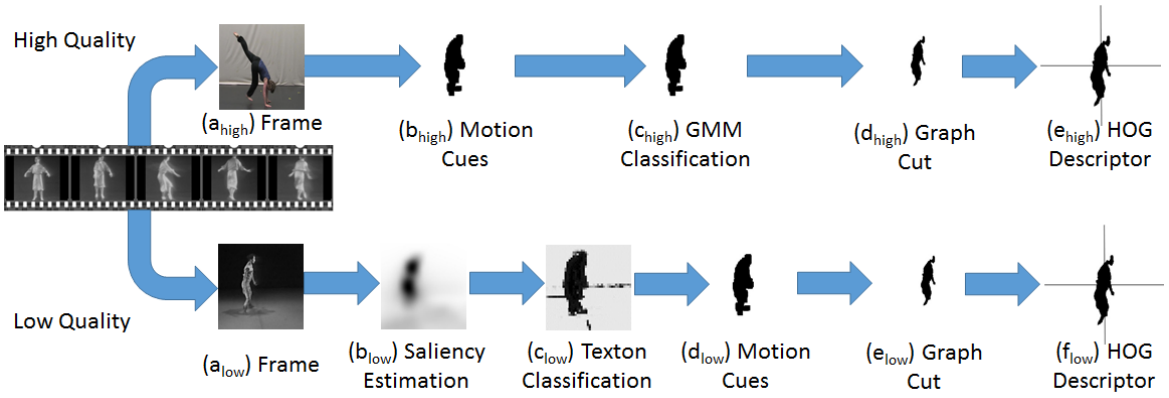



**Figure 4.2** Adaptive approach for High and Low quality footage. For low quality: Source Image (a) drives (b-d) where a Graph cut (e) combines saliency (b), texture (c) and motion (d) to form the unary term, with the standard pair-wise [27] term yielding a silhouette. For high quality: Frame-to-Frame Motion cues (b) are combined with GMM classification cues to form the unary terms Graph cut (d). Both approaches result in a HoG descriptor from the silhouette, computed per cell of a $2 \times 2$ grid.

To obtain the training examples we apply an adaptive threshold to an Itti and Koch image saliency field [90] to extract an approximate silhouette of the performer in each frame. As positive training candidates we identify the largest connected component produced during the saliency thresholding.

On the assumption that the majority of such regions gathered over the video will mostly be the performer, the collection of candidate regions is culled by removing outliers regions that exhibit unusual boundary shapes (i. e. significantly differing from the average of the candidate set). This is measured by examining an Eigenmodel of Fourier Descriptors computed from the contours and identifying outliers using the Mahalanobis distance.

Negative training candidates are sampled at random from the remaining region of each frame. A balanced number of positive and negative training examples are used to train the appearance (Texton) SVM.

### Applying the Learned Model

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 local Texton descriptor.

We enhance the spatial coherence of this probability map by using a binary graph-cut [27], 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 [177].

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 in the foreground, which we obtain simply by differencing neighbouring frames.

### 4.2.2 Implicit Pose from Silhouette

Work on human pose estimation (HPE) by Eichner et al. [65] has resulted in descriptors for pose search, using either 'hard' or 'soft' representations of explicit pose. Hard representation refers to the direct estimation of joint angles and the use of these to form a pose descriptor. Soft representation refers to a per-pixel probability vector being estimated over the image, encoding the likelihood of each class of limb being present. A pose descriptor is then formed by spatially grouping probabilities. This approach struggles with low resolution footage or complex poses [174].

Ren et al. [174] use a combination of spatial pyramids and the self similarity descriptor [188] to describe the performer, with a combination of topic modelling (LDA) and vector quantisation used to form *visual sentences*. The visual sentence descriptor does not explicitly estimate pose, but implicitly encodes pose information within a bounding box. Unfortunately the approach is not applicable to the challenging archival footage we process, as there is limited texture within the performer from which to compute the self-similarity descriptor.

Our technique is most closely aligned with the Visual Puppetry technique of Brand et al. [28] who seek to obtain a pose descriptor from a binary silhouette. Brand et al. used Central Zernike moments computed over a bounding box.

We evaluated Zernike moments and three other promising approaches to identify a suitable candidate for our implicit pose descriptor. All descriptors are computed over the bounding box surrounding the performer (i.e. of the extracted silhouette).

**Zernike Moments**

2D Statistical moments are frequently used to describe shape in object and shape recognition. Zernike moments are an affine invariant form of moment. We extract and normalise the 16-dimensional descriptor as per [28].

**Gridded Zernike Moments**

In the spirit of spatial pyramid kernel (SPK) that computes descriptors within a spatial grid, and concatenates these to form a descriptor, we explore the same for Zernike moments. A $2 \times 2$ grid is used in our implementation, yielding four localised Zernike moments that are concatenated to form a 64-dimensional descriptor. The use of gridded descriptor removes the rotational and translational invariance of the descriptor, which may be desirable given the expressivity of pose in dance.

**Gridded Histogram of Gradients (HoG)**

Computing HoG is a common approach for Shape recognition, but descriptors are often densely sampled within the region of interest at a high cost in terms of dimensionality. Therefore a simplified version is used wherein a $2 \times 2$ grid 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.

**Fourier transform descriptor**

The Fourier Descriptor is a very common approach for shape representation. The spectral distribution of a periodic signal obtained by iterating along the perimeter of the silhouette is used to form a 16-dimensional descriptor.

In a preliminary study some additional descriptors were explored including Image PCA, Central Moments, Geometric Descriptor, GF-HOG [102] and Visual Sentences [174]. These were disregarded for either low performance or high computational complexity.

### 4.2.3 Selection of Implict Pose Descriptor

To identify the most appropriate of the four descriptors to use in our system, a quantitative comparison was undertaken.

A set of $n = 15$ frames were selected manually from the training video "Blueprint" as a query set, covering representative poses in the sequence. To evaluate the performance of a particular descriptor, we first computed that descriptor over all 7962 frames in the Blueprint sequences (including the query frames). An $k$-NN clustering was then performed in the descriptor space using the $n$ query frames as the mean. A Precision@50 (i.e. precision of the 50 closest points) was computed for each of the $n$ clusters by manually deciding on the relevance of each clustered pose by inspection. Higher precisions imply a more consistent and semantically aligned descriptor.

To determine the relevance of a returned result we adopt Eichner et al.'s criteria [65], now regarded as the de facto standard in evaluating HPE systems. Specifically, given a ground-truth position of a limb (specified by it's endpoints) the detected limb position must lie within half the limb's length from the ground truth position. We use this criterion for all evaluations of our sketch based pose retrieval work.

Fig. 4.3 presents the results averaged over the $n$ frame query-set. The MAP values for the other descriptors are 52%, 62%, 77%, 42% for Zernike moments, Gridded Zernike Moments, Gridded HoG, Fourier Descriptor, respectively. Gridded HoG is the best performer by a margin of $\sim 15\%$ and is therefore adopted for our system.

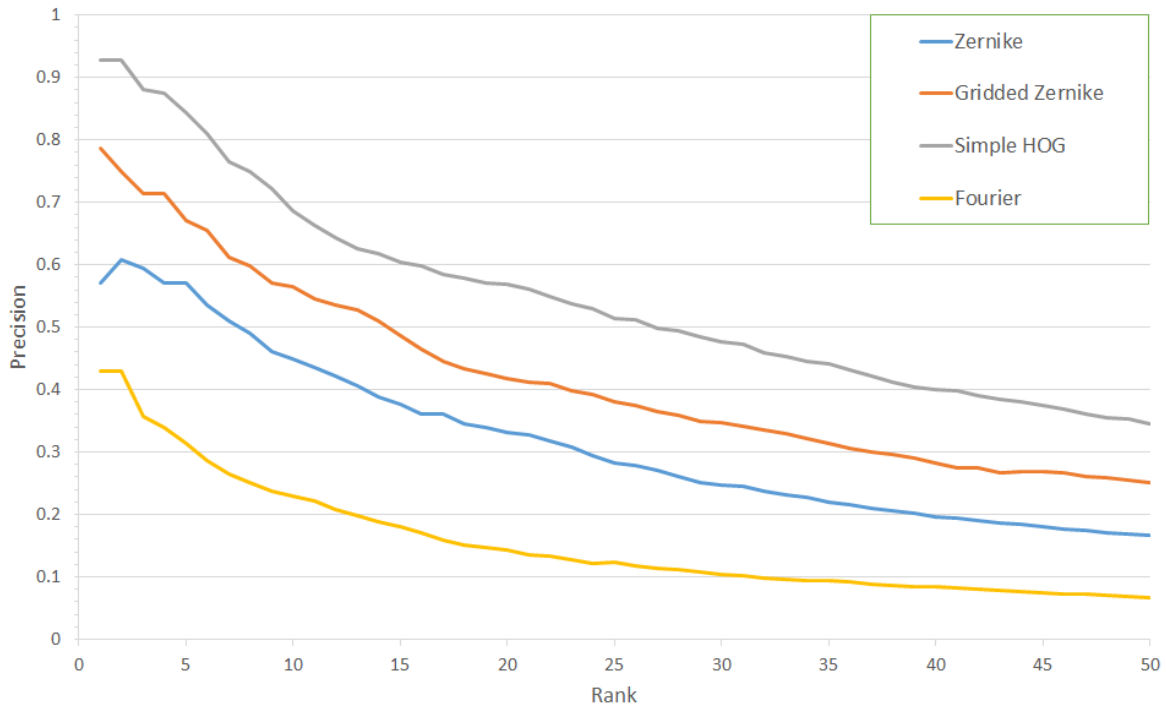



**Figure 4.3** Precision @K comparison for forms of Zernike, HoG and Fourier descriptors

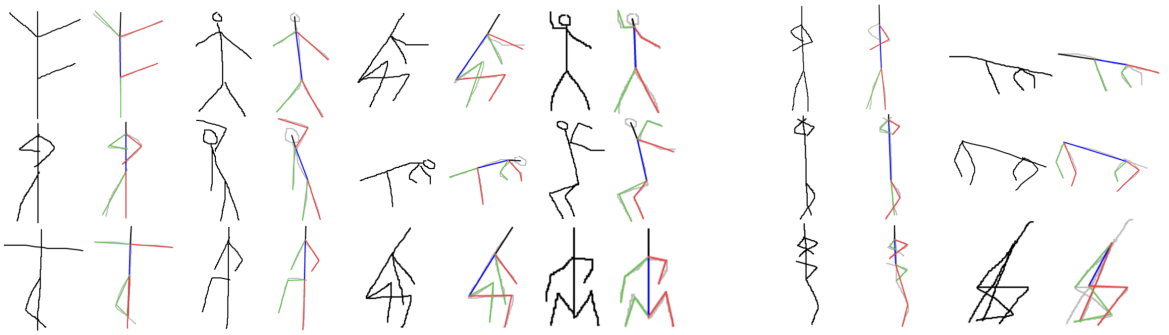
## 4.3 Sketch Description

Our system accepts a free-hand sketched stick-man as input, that we wish to represent as a concise pose descriptor. Describing a pose in terms of a loose collection of sketches strokes can be difficult as stroke order and shape could vary for a given pose. To overcome this issue we parse the sketched strokes into a stick-figure, and encode the angles between limbs within our sketched pose descriptor. The stick-figure is displayed back to the user after parsing to allow them to verify and manipulate the joint angles.

### 4.3.1 Pictogram Parsing

Our web-based interface accepts a sequence of free-hand sketched strokes as a query. We use a set of heuristics to label strokes to the components of a canonical stick-man.

Although stick-men are conceptually simple pictographic representations, when used to represent dance poses, we can end up with complex drawings like those shown in Fig. 4.4, where arms or legs cross, or where both limbs are on the same side of the body. In addition to the complexity of the poses, there is also the way users draw a stick-man. For instance, not all users draw the head. Some draw the neck, the torso and one leg using a single stroke, or the two limbs with a single stroke. In some cases the limbs intersect the torso, while in others



(a) (b)

**Figure 4.4** Representative output of the skeleton-from-sketch parsing step (Sec. 4.3.1). With (a) successful parsed and (b) failure cases. Each showing different sketched poses and the corresponding skeleton with limbs donated by green and red for left and right respectively. Despite stroke inter-occlusion and pose complexity our approach is able to recover an articulated skeleton in many challenging cases.

they do not. A few of these cases are illustrated in Fig. 4.4. Furthermore users' depictive ability is quite variable and very often poor.

The arbitrary ordering of strokes makes Markov Model approaches (e. g. [185]) ineffective at this parsing task. Our approach is able to take free-hand drawings of stick-men without requiring: stroke order; multiple strokes for body parts (lower/upper arm); stroke intersection. The only assumption is that a component of the skeleton cannot be drawn using two (or more) strokes. For instance, we cannot draw the upper arm and then the forearm using two strokes.

Sketches consist of a multitude of 2D points often very closely placed, and which sometimes overlap, especially when the user draws slowly. To eliminate this excessive number of points, and reduce noise, our algorithm starts by eliminating points from the sketch that are too close (in our case 5 pixels).

Since our sketches are composed mainly by straight lines and/or two straight line segments (e.g. a bent arm), we further simplify based on the angle defined by three temporally consecutive points. Under this simplification the middle 2D point is eliminated in order to straighten lines and reduce point count. We also tried the Douglas-Peucker algorithm [64] to simplify, but for our case this simplistic algorithm produced better results.

After simplification, we check if any of the strokes that compose the stick-man is the head. We do that by finding the stroke that is most similar to a circle or an ellipse. If we identify any, that stroke is removed from the list of strokes and the position of the head is noted.

The next step is to find the stroke that corresponds to the torso of the skeleton. To do that we use a voting system composed by the following features: size of the stroke; number of

intersections; position of the centre of mass; similarity to a straight line and closeness to the extreme points of the other strokes. The stroke that receives more votes is considered the torso stroke and is removed from the list of strokes, with the position of the torso noted.

After identifying the torso, we compute the intersections of the other strokes with it. When we have more than one intersection, we always choose the one that is closer to the top of the torso. If the strokes do not intersect the torso, we extend them until they intersect. With all the intersections identified, we have a list of potential arms and legs. From each of these lists we select the two biggest lines as limbs and we unify the shoulders and the waist to identify the common point for the arms and the legs, respectively. The final step is to identify the sides for each limb, by checking the position of the elbow/knee relatively to the torso. When both limbs are in the same side of the torso, we verify which is more to the right (from the user point of view) and consider it he right limb.

The user is able to manipulate the left-right orientation of the stick-man (e.g. whether the figure faces toward or away) as this information is absent in the sketch.

At the end of the parsing process, we have a semantic skeleton, where we know exactly how each stick-man component is defined both in terms of geometry and limb label. This provides us with the rich information necessary for computing a descriptor to describe and compare skeletons of dance poses.

### 4.3.2 Pose Descriptor from Articulated Skeleton

The joint angles of the articulated skeleton parsed from the query sketch are used to form the sketch (query) descriptor.

With the advent of Kinect, a number of recent works have explored pose retrieval using joint angle based descriptors. The majority of these are 3D based and commonly using quaternions to describe the properties of a joint. Jammalamadaka et al. [109] proposed a skeletal front-end to Eichner et al.'s 2D Pose Search system, using a descriptor concatenating the sine and cosine of each joint angle, so avoiding the discontinuity at $\{0, 2\pi\}$.

Our 2D articulation is defined in a standard manner with the torso as a root node, attached to upper-limbs, head and lower-limbs which are in turn attached to the extremities. Given a set of 10 absolute joint angles in the articulated skeleton (i. e. the angle of each component to the vertical) $\{\psi_0, \psi_1, ..., \psi_9\}$, and 10 relative joint angle $\{\theta_0, \theta_1, ..., \theta_9\}$ i. e. the angle between a joint and its parent. In contrast to Jammalamadaka et al., we explored the inclusion of the torso $i = 0$ orientation to the vertical, (i. e. $\theta_0 = \psi_0$). Various approaches to the encoding joint angles within our pose descriptor are explored.

Writing our skeletal pose descriptor $S = \left[\mathbf{v}_0^T \mathbf{v}_1^T ... \mathbf{v}_9^T\right]$ the various possible encodings for $\mathbf{v}_i$, with varying range of $i$ were explored as follows:

**Method 1: Sine Absolute Angle ('$Sin$')**

$$\mathbf{v}_i = \sin(\psi_i), \qquad 0 \leq i \leq 9 \tag{4.1}$$

**Method 2: Cosine Absolute Angle ('$Cos$')**

$$\mathbf{v}_i = \cos(\psi_i), \qquad 0 \leq i \leq 9 \tag{4.2}$$

**Method 3: Jammalamadaka et al. [109] ('$CosSin$')**

$$\mathbf{v}_i = \begin{bmatrix} \cos(\psi_i) \\ \sin(\psi_i) \end{bmatrix}, \qquad 1 \leq i \leq 9 \tag{4.3}$$

**Method 4: Sine Relative Angle ('$JointSin$')**

$$\mathbf{v}_i = \sin(\theta_i), \qquad 1 \leq i \leq 9 \tag{4.4}$$

**Method 5: Sine Relative Angle ('$JointSinExt$')**

$$\mathbf{v}_i = \sin(\theta_i), \qquad 0 \leq i \leq 9 \tag{4.5}$$

**Method 6: Cosine Relative Angle ('$JointCos$')**

$$\mathbf{v}_i = \cos(\theta_i), \qquad 1 \leq i \leq 9 \tag{4.6}$$

**Method 7: Cosine Relative Angle ('$JointCosExt$')**

$$\mathbf{v}_i = \cos(\theta_i), \qquad 0 \leq i \leq 9 \tag{4.7}$$

**Method 8: Relative (Modified) Jammalamadaka et al. [109] ('$JointCosSin$')**

$$\mathbf{v}_i = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}, \qquad 1 \leq i \leq 9 \tag{4.8}$$

**Method 9: Relative (Modified and extended) Jammalamadaka et al. [109] ('$JointCosSinExt$')**

$$\mathbf{v}_i = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix}, \qquad 0 \leq i \leq 9 \tag{4.9}$$

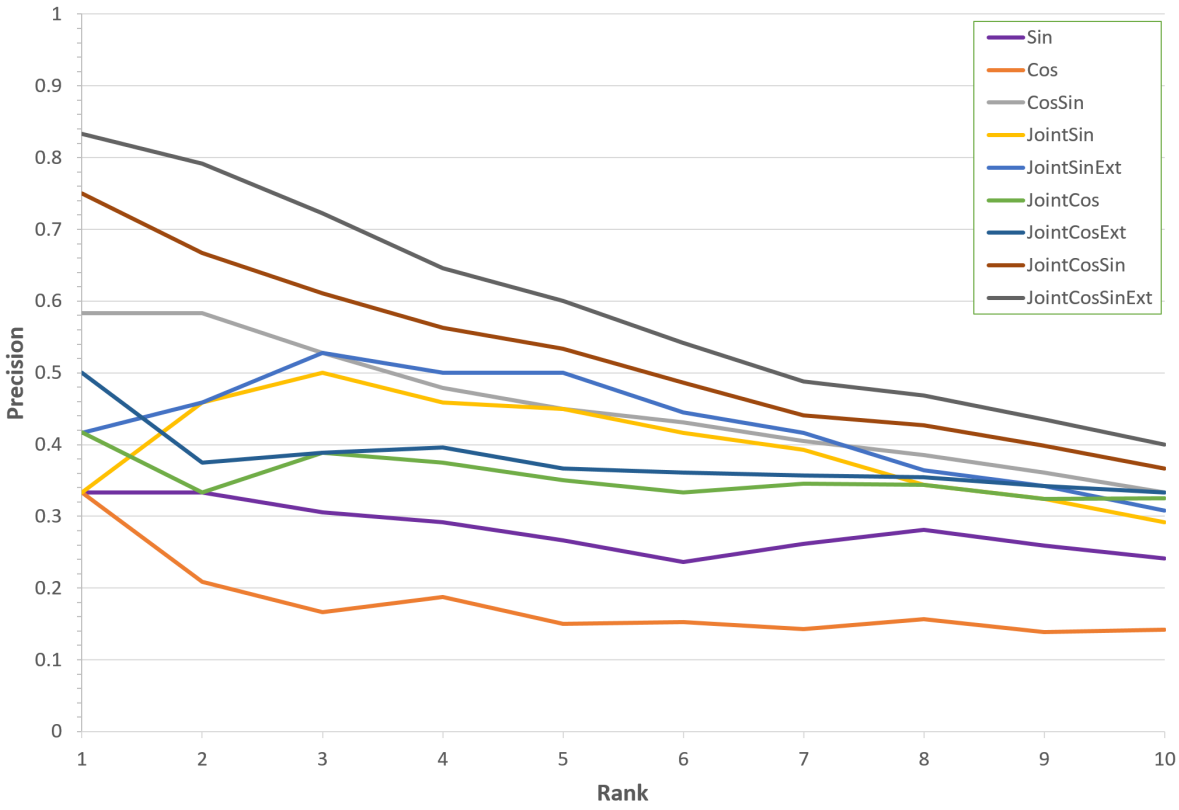



**Figure 4.5** Comparative evaluation of 8 methods for encoding the skeletal pose descriptor. Precision@k ($k = [1, 10]$) comparison indicates the superiority of finite difference methods over the alternatives including prior approach of Jammalamadaka et al. [109].

### 4.3.3 Comparative Evaluation of Pose Descriptor

The nine competing candidates for $\mathbf{v_i}$ (eq. 4.1-4.9) are evaluated using a set of 230 poses from the Blueprint video. A skeleton is manually sketched (and parsed) for each of the 230 poses using the video frame as the user's visual guide. These poses are selected manually to be diverse and representative of poses likely to be encountered in dance video.

Similar to the video descriptor methodology for comparison outlined in subsec. 4.2.3 we treat a sub-set of these poses as queries (12 queries) and run a nearest-neighbour assignment using Euclidean distance against all 230 poses encoded via a given candidate method. The resulting rankings are interpreted manually as the results of a retrieval tasks — the top 10 ranking poses are scored relevant or irrelevant enabling the computation of the Precision@k score (for $k = [1, 10]$).

Fig. 4.5 summarises the Precision@k scores for all nine candidates. Our results reflect the results of Jammalamadaka et al. [109] that claimed superiority of *CosSin* over *Sin* or *Cos* encoding. However we note that significant benefits can be gained considering the relative

angles of joints and also the inclusion of the torso $\theta_0 = \psi_0$. We adopt candidate 9 (eq. 4.9) over the alternatives, based on the performance benefits indicated by this comparison.

## 4.4 Sketch based Pose Retrieval

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. 4.4.1). A graph-based strategy is used to compute similarity between a query and candidate video frame (pose) by approximating geodesic distance in piecewise-linear across these manifolds (Sec. 4.4.2). 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 4.4.3).

### 4.4.1 Manifold construction

The Quality Thresholding (QT) clustering algorithm [94] (outlined in alg. 1) is used to identify the set of *training poses* from training footage by clustering data points (frames) within our descriptor space (Sec. 4.2.2). 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 ∼4500 frames) sampled from the performance 'Blueprint' (*c.*1978).

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 $w(n_s \mapsto n_t)$ on each edge are proportional to the Euclidean distance between the nodes connected.

$$w(n_s \mapsto n_t) = \begin{cases} 1 - \exp(|n_s - n_t|^2) & \text{if } |n_s - n_t| < T; \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

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 the shortest path between two nodes (see sec. 4.4.2).

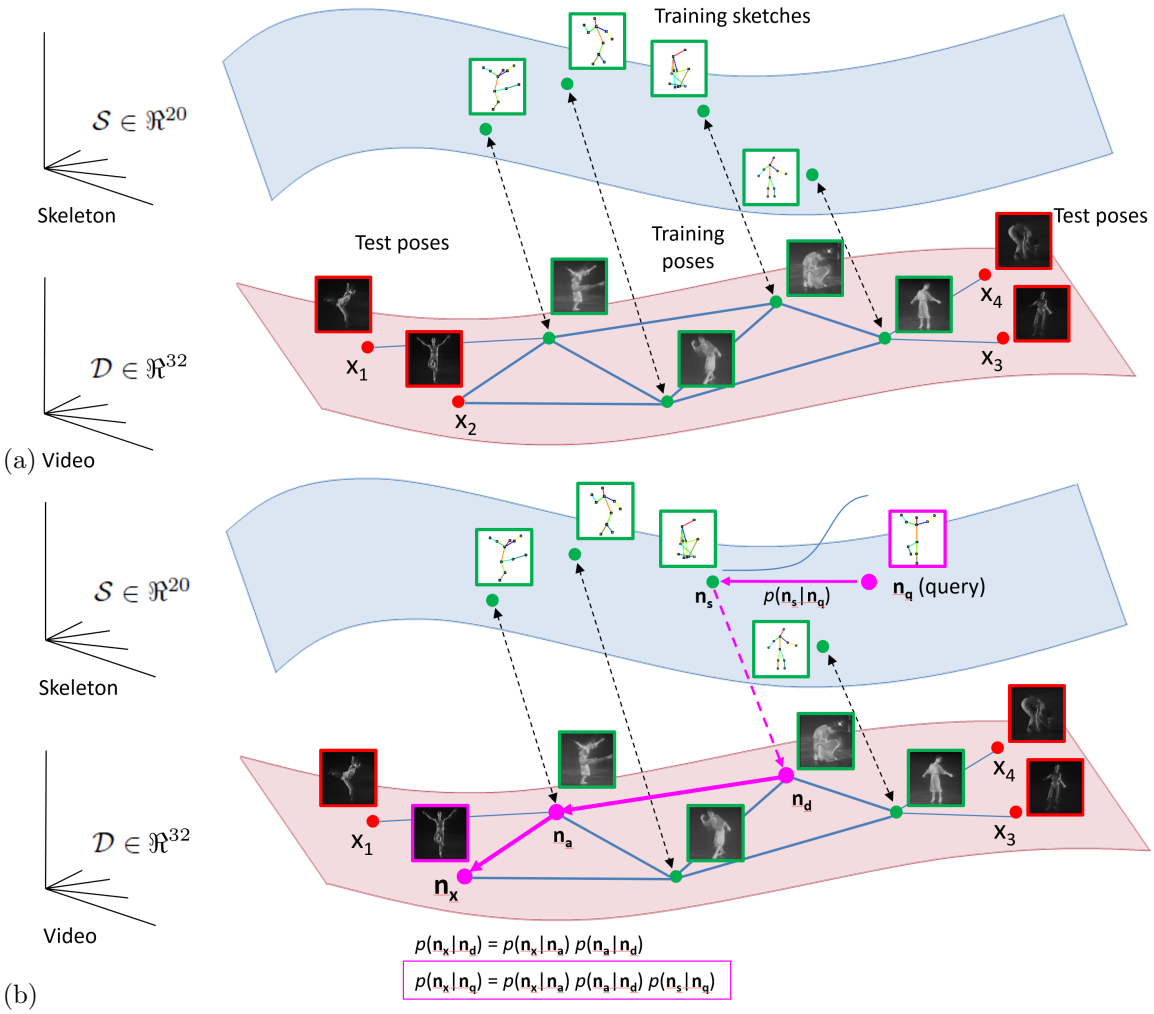



**Figure 4.6** Manifold mapping underpinning our retrieval system. (a) manifold construction. Training poses (green) are manually marked up creating sparse correspondence between $\mathcal{S}$ and $\mathcal{D}$. The search graph is constructed across training points in $\mathcal{D}$ to approximate the manifold. Additional points (red) are added to $\mathcal{D}$ from each video frame in dataset. 'Confident' frames (section 4.4.1) 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 $x_i$.

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 categorise 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 4.10a. The geodesic distance across the manifold for any two

**Algorithm 1** Quality Threshold Clustering

```
1: procedure QTCluster(d_1, ..., d_N,thresh)
2:     C ← []
3:     if ||d|| < 1 then
4:         return
5:     end if
6:     for n = 1 to N do
7:         flag ← TRUE
8:         A_i ← i
9:         while (flag = TRUE)and(A_i ≠ D) do
10:            find J ∈ (G − A_i)
11:            if diameter(A_i ∪ A{j}) > thresh then
12:                flag ← FALSE
13:            else
14:                A_i ← A_i ∪ {j}
15:            end if
16:        end while
17:    end for
18:    return C
19: end procedure
```

poses in the dataset is now approximated by a shortest-path computation over $\mathcal{G}$.

### 4.4.2 Learning Mapping $\mathcal{S} \leftrightarrow \mathcal{D}$

We learn a mapping between $\mathcal{S}$ and $\mathcal{D}$ as a one-off process using the set of training poses identified in sec. 4.4.1. We manually annotate each pose with a sketch, from which the joint angles are obtained via section 4.3. 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:

$$p(s|q) \propto \exp - \frac{|q - s|^2}{2\sigma}. \tag{4.11}$$

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 normalised similar to eq. 4.11, but where the distance between $s$ and $d$ is the geodesic distance. So the normalised probability of frame $n_d$ and an arbitrary frame $n_x$ being similar are:

$$p(n_x|n_d) = \prod_{\{a,b\}\in\mathcal{G}} 1 - p(n_a|n_b). \tag{4.12}$$

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.

### 4.4.3 Matching a sketched pose to a video frame

Combining equations 4.11 and 4.12 we compute the conditional probability of any video frame in our database ($n_x$) being similar to our query ($q$) as:

$$p(n_x|q) = p(s|q)p(n_d|n_x). \tag{4.13}$$

where $p(n_d|n_x) \propto p(d)p(n_x|n_d)$ by Bayes' rule, 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. 4.10b.

Performing video retrieval is simply a matter of evaluating eq. 4.13 for each video frame within the graph and ranking each frame based on this probability.

### 4.4.4 Bi-directional mapping for Visual Summarisation

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. 4.7, 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

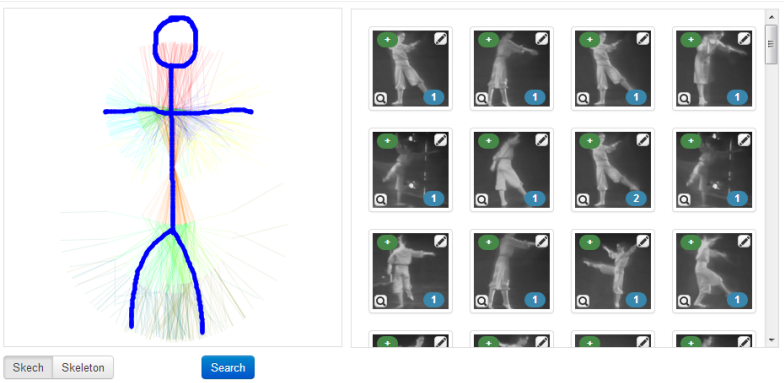



**Figure 4.7** Screen-shot of the resulting sketch based pose retrieval interface. Results for the sketched query (left) and laid out in a panel (right) with temporally close clips grouped together (expandable via the green plus) to ensure diversity of results. The user is able to manipulate joints of the skeleton, once parsed, to refine the query. Ghosts of the most relevant poses found (inferred through $\mathcal{D} \rightarrow \mathcal{S}$) are indicated behind the sketched query.

readily identify temporally disjoint segments of the video that closely match the query pose of interest.

The ability to map bidirectionally 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 visualised beneath the users sketch as outlines to assist in the retrieval process. This produces an interactive interface reminiscent of the ShadowDraw clipart retrieval system [130].

In addition we leverage the ability to estimate explicit pose directly from video later in Chapter 5, via a visual summarisation tool that enables users to select pose sequences of interest within video clips.

### 4.4.5 Evaluation of Manifold based Pose Search

Retrieval is evaluated using the learned manifold for Blueprint, generalised onto unseen videos (ThreeD, Autumn, Expressive) with performance quantified using Average Precision (AP).

We evaluate the ability of the pose retrieval system to generalise 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 4.9). 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 judged to be out of

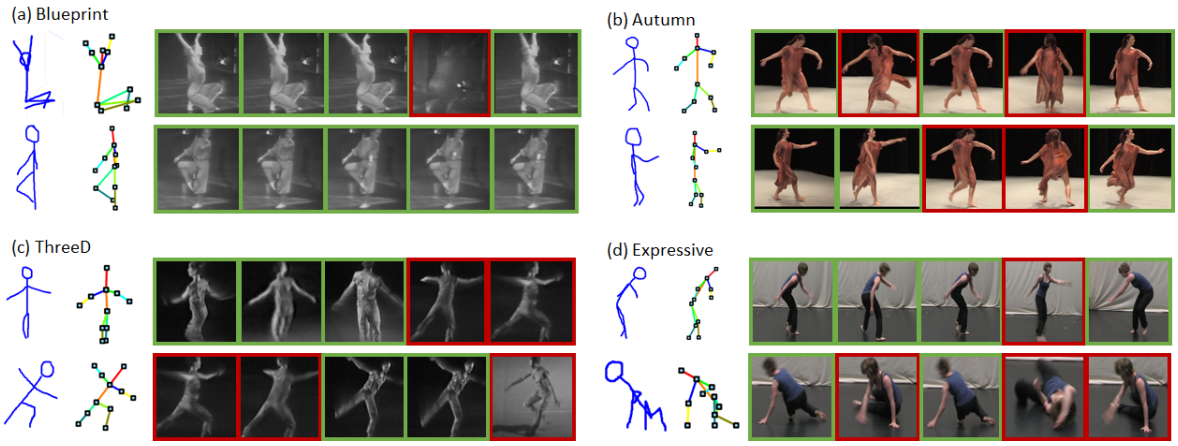



**Figure 4.8** Top 5 results of the sketch(left most) / skeleton query(second left) for: (a) Blueprint(Train) video, (b) Autumn(test) video, (c) ThreeD(test) video, (d) Expressive(test) video

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 4.8 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 4.8c 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 4.9. The MAP scores for Blueprint (60.0%), ThreeD(32.0%), Autumn(47.5%), Expressive (38.4%) indicate the system was able to generalise well from minimal training, while maintaining an acceptable precision content that differed from the training exemplars. Later in this chapter we demonstrate how adaptions to this approach can bring these figures closer to that of Blueprint. 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

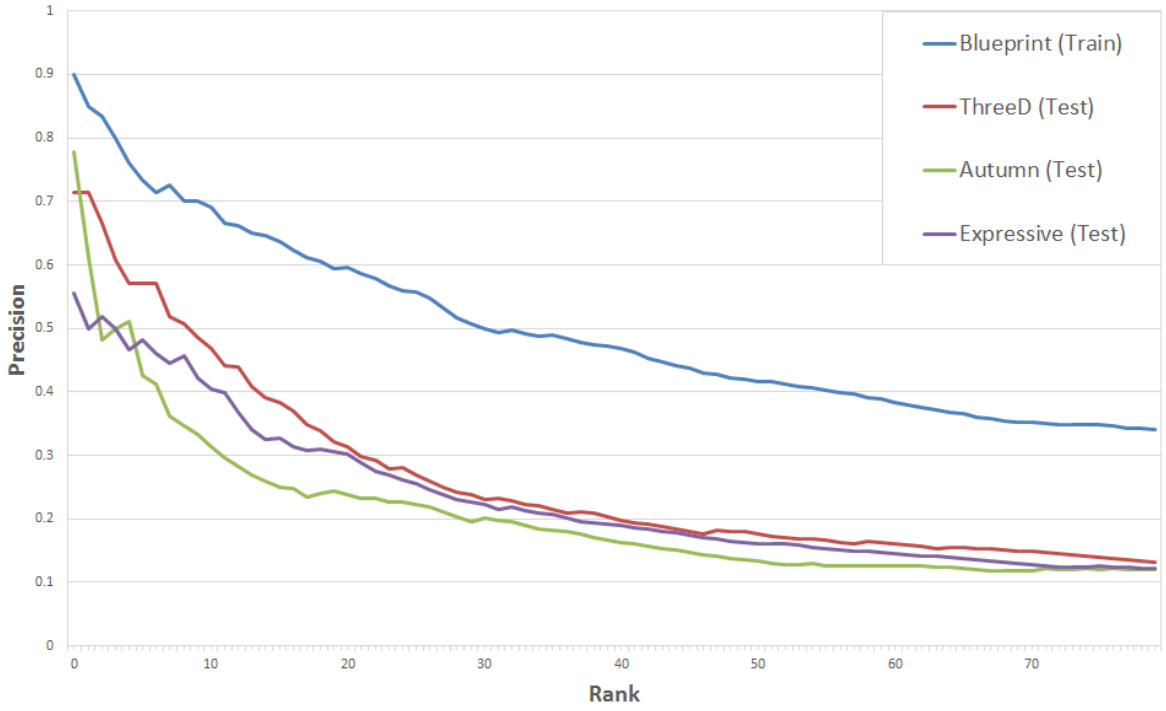



**Figure 4.9** Plotting Precision@k for Blueprint (training), ThreeD (test) Autumn (test), Expressive (Test) over the top $k = [1, 80]$ ranked results

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.

## 4.5 Domain Adaptation for Indexing Multiple Videos

The manifold mapping approach to pose search, as described, makes two assumptions about the structure of spaces $\mathcal{S}$ and $\mathcal{D}$. First, the (230 Blueprint) training poses used to map the manifold in $\mathcal{S}$ are representative of all possible poses in dance video. Second, the manifold in $\mathcal{D}$ is representative of all real dance video.

Whilst the former assumption is reasonable (the training set is selected to be representative, by design) the latter is not realistic since the appearance of the performer in a particular pose within one video may significantly differ from that of a performer in the same pose within a different video. Although appearance variation is somewhat mitigated through the use of silhouettes rather than raw pixel data, variations in performer shape (due to build, or costume) are likely.

We therefore explore the use of domain adaptation techniques to apply the manifold learned

in $\mathcal{D}$ for one video (hereafter the 'target' video, written $\mathcal{D}_t$ in effect the training video for which $\mathcal{S} \leftrightarrow \mathcal{D}$ was learned) to the appearance space of another video (hereafter the 'source' video, written $\mathcal{D}_s$).

Specifically we wish to learn mapping $\mathcal{D}_s \leftrightarrow \mathcal{D}_t$ so that descriptors (and thus frames) from arbitrary 'source' videos may be added to the graph search structure defined by the 'target' (training) video from which the manifold is initially constructed. In Sec. 4.4.1 descriptors from 'source' videos were simply added to the graph (search structure), ignoring any potential problems due to domain variation. Under this modified scheme, 'source' descriptors are first subject to a transformation, determined by mapping $\mathcal{D}_s \leftrightarrow \mathcal{D}_t$, to align them to the target domain prior to being added to the manifold. We show in Sec. 4.5.3 this leads to improved accuracy.

### 4.5.1 Cross-video correspondence

We learn mapping $\mathcal{D}_s \leftrightarrow \mathcal{D}_t$ using a set of corresponding pairs of points across both domains, established using an auxiliary pose matching technique. As noted in Sec. 2.5.2, explicit human pose estimation (HPE) is unreliable for our challenging archive footage. Yet on a small subset of frames, a recent HPE algorithms (Yang et al. [235]) can occasionally give an accurate result. Whilst the sporadic nature of these more reliable results makes HPE impractical for general pose search (i. e. requiring reliable estimation over all frames) it is practical for identifying a small set of sparse correspondences between $\mathcal{D}_t$ and $\mathcal{D}_s$, assuming the reliability of the estimate can be quantified to detect such sporadic occurrences.

HPE provides a skeleton from a video frame, i. e. a mapping $S(\mathcal{D}) \mapsto \mathcal{S}$. Based on empirical observations we make the assumption that temporal stability of $S(.)$ is directly correlated to the reliability of the HPE. Given a video sequence $D(t) \in \mathcal{D}$ we quantify the reliability $r(t)$ (over time $t$) as:

$$S_\mu(t) = \frac{1}{2n+1} \sum_{i=t-n}^{i=t+n} S(D(t)). \tag{4.14}$$

$$r(t) = \frac{1}{2n+1} \sum_{i=t-n}^{i=t+n} |S(t) - S_\mu(t)|^2. \tag{4.15}$$

For a temporal window of half-size $n$ (we use $n = 5$, i. e. quantifying stability over around half a second). Thresholding the score $r(t)$ enables identification of frames $D(t)$ that appear reliable (we use a threshold of 0.15 based on manual visual verification of the quality of HPE at this level).

A set $\mathcal{P}$ containing pairs of corresponding points $P_s(i) \in \mathcal{D}_s \mapsto P_t(i) \in \mathcal{D}_t$ is established between the two domains (videos) where $S(P_s(i)) \cong S(P_t(i))$ considering pairings across all frames where $r(t) < 0.15$. These pairs describe known correspondences between 'anchor points' in two domains, i. e. points that share the same pose. It is from these anchors that the domain adaptation for all points in the domains is derived.

### 4.5.2 Piecewise domain transformation

Common approaches to domain adaptation involve learning a global transformation that maps features sampled under one domain ($\mathcal{D}_s$) onto another domain ($\mathcal{D}_t$). This transformation is commonly a global translation, a global scaling, or a general affine transformation computed in the high dimensional space using anchors established between the two domains (i. e. those of Sec. 4.5.1. This process is referred to as transductive transfer learning (TTL) [155]. We experimented with variants of affine global transformation for TTL but were unable to obtain good results.

Concluding that a single global TTL transformation is insufficiently expressive to warp between the two domains we therefore adopted a piece-wise TTL transformation, i. e. a set of transformations each applicable to different parts of manifold $\mathcal{D}_s$. The transformation identified as yielding good results was a simple translation, defined in piece-wise fashion one translation per anchor.

Specifically for a given anchor pair ($\{P_s(i), P_t(i)\}$) the perfect transformation (translation vector $\nu(i)$) between those points is, by definition:

$$\nu(i) = P_t(i) - P_s(t). \tag{4.16}$$

To transform a general point $Q_s \in \mathcal{D}_f$ in the source domain, to $Q_t \in \mathcal{D}_\sqcup$ in the target domain, we translate it by a weighted combination of perfect translation vectors obtained from nearby anchors:

$$Q_t \leftarrow Q_s + \sum_{i \in |\mathcal{P}|} \hat{w}_i \nu(i). \tag{4.17}$$

$$w_i = \exp(-\frac{\chi(P_s(i), Q_s)}{\tau}). \tag{4.18}$$

where $\tau$ controls the spatial decay of influence of each anchor point, and normalised weight $\hat{w}_i = w_i / \sum_{i \in |\mathcal{P}|} w_i$. The impact of tuning parameter $\tau$ is explored in Sec. 4.5.3. Distance measure $\chi(u, v)$ is the geodesic distance in $\mathcal{D}_s$. In order to compute this, a temporary manifold is formed via the process of Sec. 4.4.2 using only frames in $D_s$, and distance computed using the similarity score of eq. 4.12. In early experiments we found this to give superior results

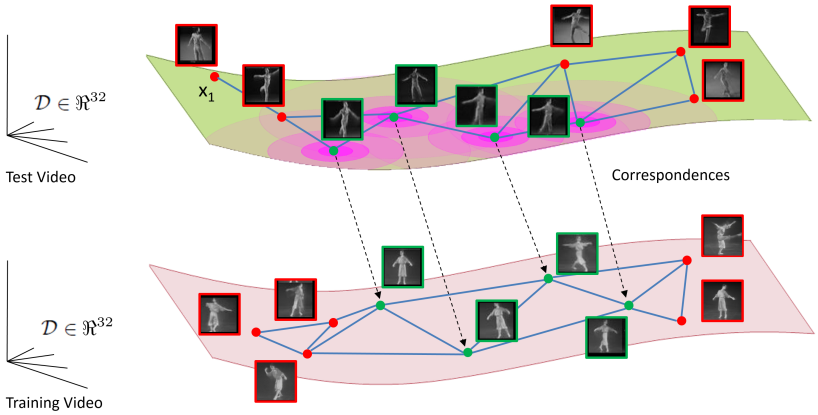



**Figure 4.10** Domain Adaptation between ingested video ($\mathcal{D}_s$, upper) and the training video ($\mathcal{D}_t$, lower). Automatically identified correspondences (green) using the HPE method of Yang et al. [235] create anchors each of which describes a localised translation between $\mathcal{D}_s$ to $\mathcal{D}_t$. A weighted combination of such translations are used to map any frame (red) between the two domains. The sphere of influence of each locally modelled translation is determined via parameter $\tau$ (Sec. 4.5.2).

that simple Euclidean distance $|a - b|$, again supporting the suggestion that the manifold of plausible poses in $\mathcal{D}$ is non-linear.

Having computed the modified video descriptor $Q_t$, for a given $Q_s$ from the original video, it is added to the graph search structure as per the process of Sec. 4.4.1.

### 4.5.3 Evaluation of Domain Adaptation

We evaluate the Domain Adaptation extension of our pose search first by investigating the input of decay parameter $\tau$, and second in terms of overall performance against our baseline system with no domain adaptation (as proposed in Sec. 4.4).

For each of our three test videos we vary the value of $\tau$ from $0 \rightarrow 0.2$ at 0.025 increments, plotting the accuracy (MAP) against the value of $\tau$. The same evaluation methodology as Sec. 4.4.5 is adopted to compute the MAP. Note that the case where $\tau = 0$ implies anchor points cast no influence over their surrounding volume within $\mathcal{D}_f$ and thus is equivalent to our baseline with no domain adaptation.

Fig. 4.11 illustrates the impact of $\tau$ indicating values in range [0.025,0.1] to produce optimal performance across the three videos. Given the similar neighbourhood in which MAP peaks, the system appears not to be too sensitive to the setting of $\tau$ however for optimum performance (and if use cases permit) the value could be tuned for each video as it is ingested to the system. We identify the best value of $\tau$ for ThreeD, Autumn and Expressive to be 0.075,0.1,0.025 respectively, performing with 47.3%, 55.3%, 50.60% MAP in contrast to the

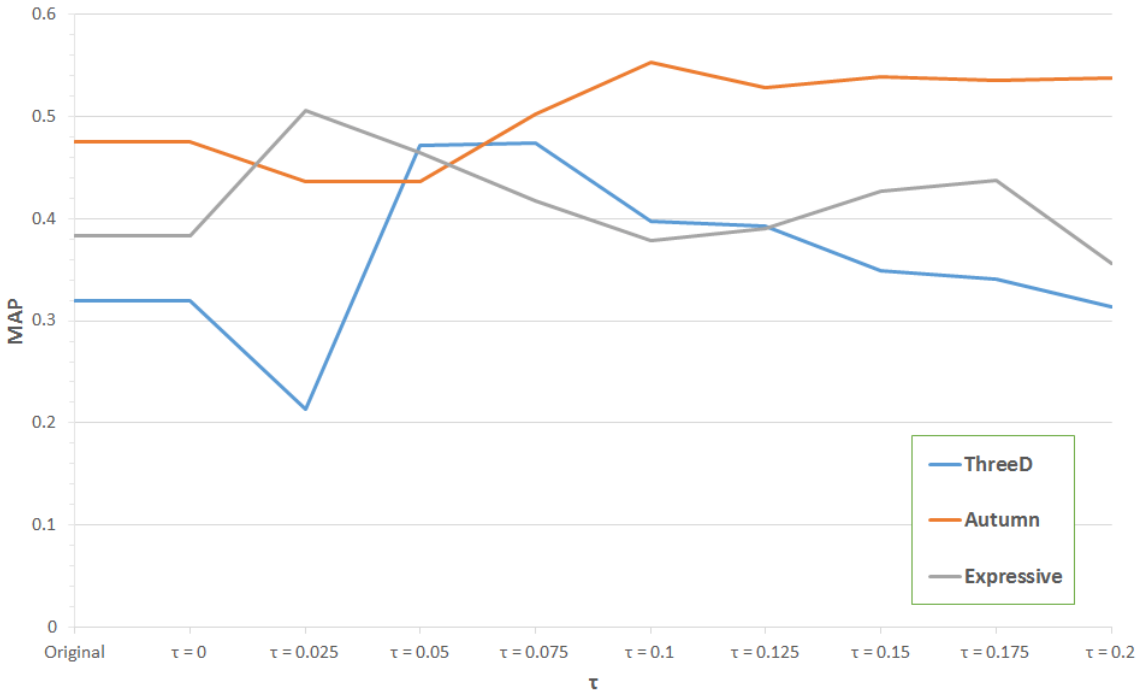



**Figure 4.11** Plotting the effect of decay parameter $\tau$ versus accuracy (MAP) for the three test videos ThreeD, Autumn and Expressive. Performance is shown to peak within a local range of $\tau = [0.025, 0.1]$ in all cases.

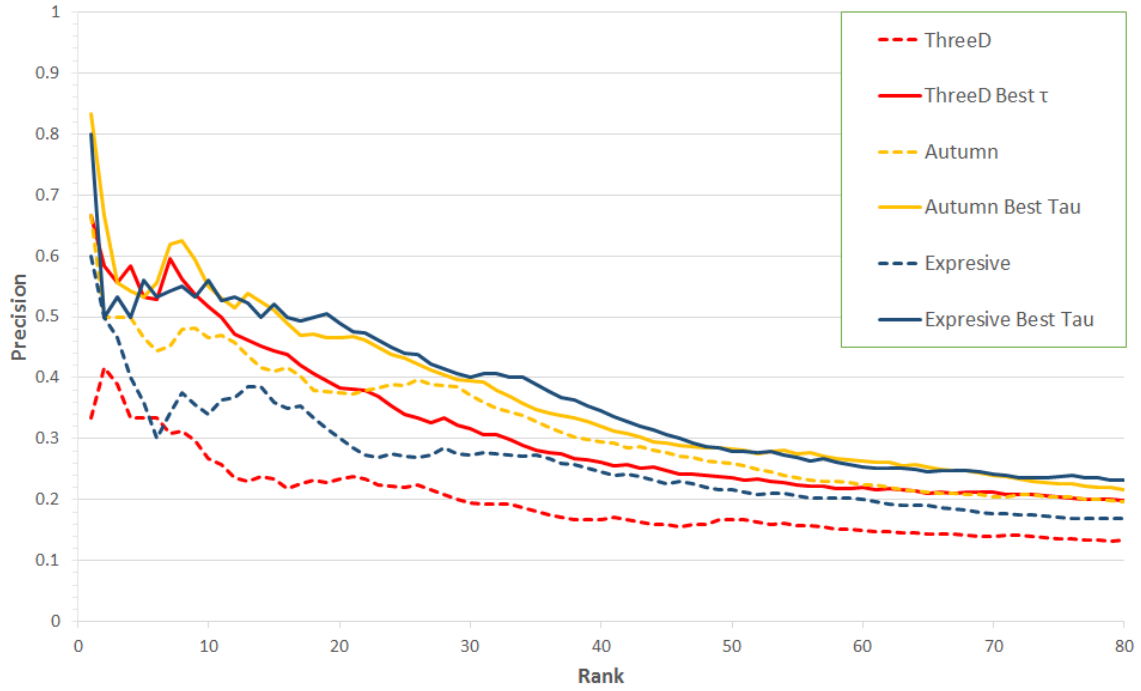



**Figure 4.12** Precision comparison of ThreeD, Autumn and Expressive for before and after Domain Adaptation

original 32.00%,47.5%,38.4% MAP.

Using the identified optimal values of $\tau$ for each video we compare the Precision @K profile of the baseline (no domain adaptation) approach in Fig. 4.12. A consistent improvement is seen for all clips versus the baseline proving benefit of domain adaptation within our framework. Relative improvement of MAP versus the baseline is 47.9%, 16.4%, 31.9% for ThreeD, Autumn and Expressive respectively. Constituting an average improvement of 32% over the baseline method.

## 4.6 Construction for the Multiple Video Manifold

Sec. 4.4 outlines the construction of the manifold within $\mathcal{D}$. Frames from the training video are used to create a base manifold in the form of a graph, to which other (test) frames from that video and additional videos are attached to complete the search index.

This strategy was adopted for two reasons: 1) the manifold of plausible poses is structured around the training video under this approach, which by design is chosen to cover all plausible poses so seems an intuitively reasonable approach; 2) the computationally expensive process of building the manifold (relative to the simpler task of appending frames to it) is limited to just a single (training) video.

However alternative strategies for building the manifold exist for indexing multiple videos and are now explored:

#### Method 1: Single Video Manifold (SVM)

As outlined above and in Sec. 4.4, the manifold is constructed using a single (training) video and frames from other videos are then attached to this template manifold using $k$-NN.

#### Method 2: Multiple Video Manifold (MVM)

The manifold is constructed globally over all videos (training and test). The manifold more comprehensively models all variations in pose within the dataset but is significantly more expensive to build, and must be completely rebuilt each time a new video is ingested to the system.

#### Method 3: Domain Transfer Video Manifold (DT-VM)

As per 'Single Video Manifold' but domain adaptation is applied to conform the video descriptors to the training video domain prior to their appending to the search index (i. e. following the approach proposed in Sec. 4.5).

| | SVM | MVM | DT-VM | DT-MVM |
|---|---|---|---|---|
| ThreeD | 32.0 | **50.0** | 47.3 | 48.9 |
| Autumn | 47.5 | 49.1 | 55.3 | **56.1** |
| Expressive | 38.4 | 41.3 | 50.6 | **53.5** |

**Table 4.1** Accuracy (MAP) of each of the four manifold construction strategies proposed in Sec. 4.6 for each of the three test videos.

#### Method 4: Domain Transfer Multiple Video Manifold (DT-MVM)

Domain adaptation (Sec. 4.5) is applied to conform all videos to a single domain (that of the training video) and then the approach of 'Multiple Video Manifold' is adopted i. e. a manifold is built using all of the training and test data.

### 4.6.1 Comparative Evaluation of Construction Strategies

Fig. 4.13 plots Precision@k for the top $k = [1, 80]$ ranked results, following the evaluation methodology of previous sections. The curve characterises performance of all four strategies for each video, and the mean average precision (MAP) for each video is tabulated in Tbl. 4.13.

In all cases the construction of the manifold using all available video (MVM/DT-MVM), rather than a single test video (SVM/DT-VM), delivers a performance increase (17.6%, 2.7%, 3.8% for ThreeD, Autumn, Expressive respectively). The use of domain adaptation when creating the manifold also delivers a performance increase of similar magnitude in most cases. Yet whilst domain adaptation is computationally cheap, the significant computational overhead of multiple video manifold approaches must be weighed against their benefits. Scalability of a video retrieval system is always a consideration for a practical system, and a system that must be fully re-indexed with the ingestion of each new videos is not likely to be workable in real situations. Note that the additional complexity of the manifold topology produced by the MVM and DT-MVM methods will also decrease the query-time speed of the system, as the geodesic distance over the manifold is computed using Dijkstra's algorithm. The recommendation is therefore to adopt the DT-VM strategy for scalability and performance.

## 4.7 Conclusion

In this chapter we have presented a method for Sketch based Pose Retrieval. Our retrieval system operates by learning a mapping between a query space (skeletal joint angle) representation parse from a free-hand sketch, and a video descriptor space. The learning process uses around two hundred hand-annotated video frames. Once learned, this mapping is shown

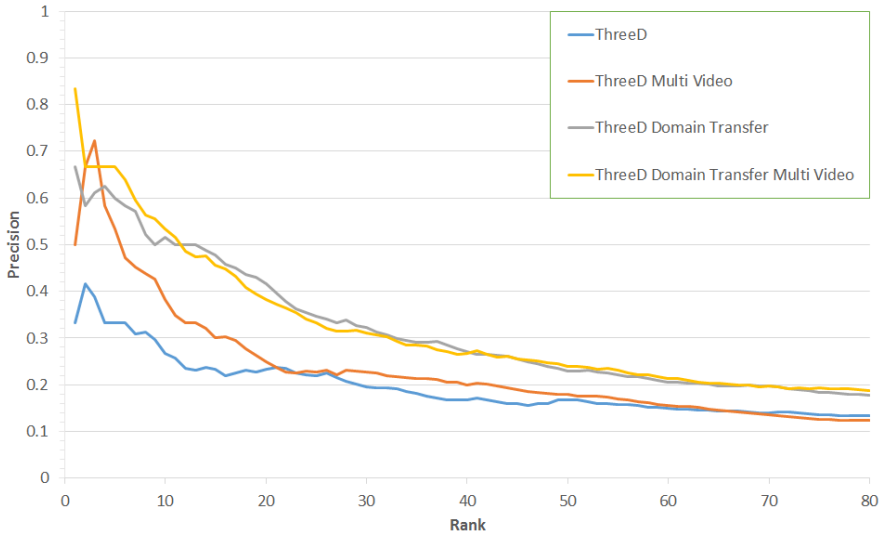
to generalise to unseen archival video footage with an average (over test videos) MAP of $\sim 39\%$ without domain adaptation (SVM, Sec. 4.4), and $\sim 51\%$ with domain adaptation (DT-VM, Sec. 4.5). We extensively evaluated possibilities for both the skeletal and the video descriptors, adopting representations based on finite differences of relative joint angles, and spatially gridded HoG descriptors respectively.

Applying domain adaptation to our pose search framework delivered a significant accuracy benefit, and relied upon the use of an auxiliary HPE method to map correspondence between the appearance domains of a pair of videos. Although HPE was shown in Sec. 2.5.2 to be highly unreliable over archival footage and so infeasible as a descriptor for pose search it was sufficiently robust to enable transductive transfer learning (TTL) as a means of domain adaptation. Various ITL transformations were explored and piece-wise translation was adopted. It was demonstrated that performance gains are similar, and in addition to, gains made via domain adaption could also be achieved through more comprehensive modelling of the video descriptor manifold, i. e. performing more comprehensive sampling of the space of plausible dance poses. These results were reported but recommendation against this approach was made due to concerns over scalability for larger datasets and the practicalities for deploying such a system.

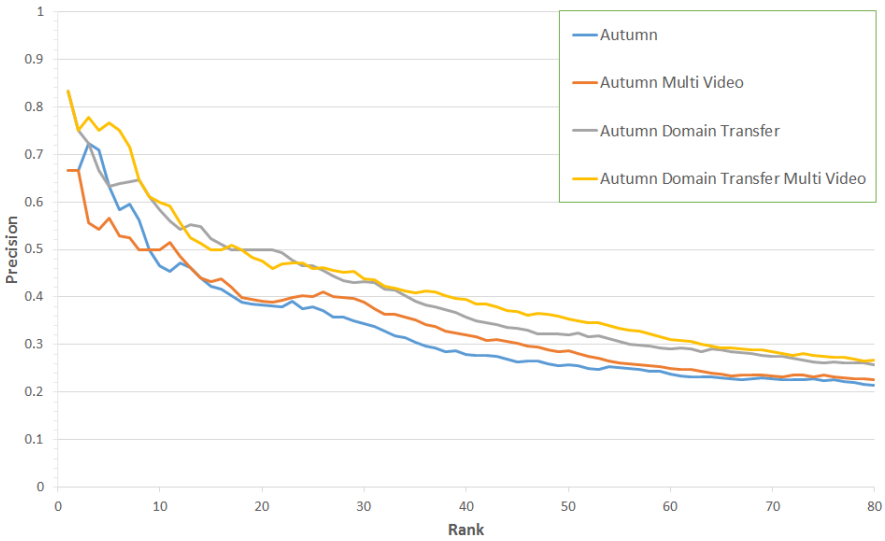
### 4.7.1 Future Work

The most fruitful improvements to the work of Chapter 4 seem to lie within our domain adaptation technique. These could focus on automating the fine-tuning of decay parameter $\tau$ which is currently performed manually on a per-video basis (or set to a default value in identified range $\tau = [0.025, 0.1]$. It also can be seen empirically from $r(t)$ that certain anchor correspondences are less reliable (stable) than others, and this confidence score could drive a local modifier on $\tau$ on a per-anchor point basis to improve the accuracy of the domain transformation (or drive a selection process that further filters anchor points).

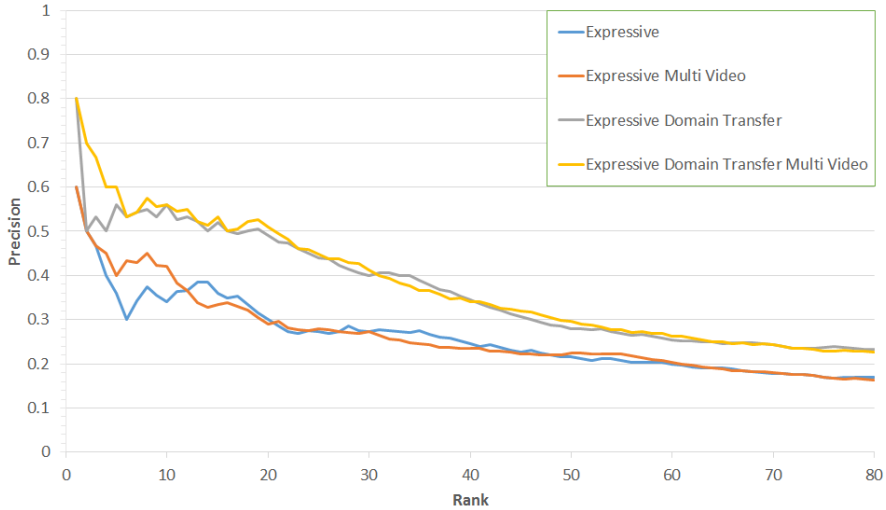
In contrast to the prior work of Chapter 3, the proposed retrieval method focuses entirely upon shape and indexes content at the frame rather than clip level. Yet similar to Chapter 3, only a single sketched representation is used as a search query. In the next Chapter we explore multiple extensions to our pose search framework, including the ability to search for a sequence of poses and inter-connecting actions, for a variety of applications including video synthesis and retrieval.




(a) ThreeD




(b) Autumn




(c) Expressive

**Figure 4.13** Comparison of manifold construction strategies. Plotting accuracy as Precision@k for ranks $k = [1, 80]$ for each of the three test videos: ThreeD, Autumn, Expressive for all of the four methods outlined in Sec. 4.6

# Chapter 5

# ReEnact: Graph Representation for Search and Synthesis

We propose a graph-based representation for sketched Visual Narratives that describe events comprising more than one action. Extending the pose retrieval system of the previous chapter, we explore the application of our graph representation to three problems. First, an application to video search. We demonstrate the ability to retrieve a sequence of poses interspersed by actions linking those poses. Second, an application to video synthesis in which Visual Narratives can be used to specify a desired video and fragments of existing video footage retrieved and concatenated seamlessly to produce the requested new sequence. This second contribution relies upon the first for video retrieval, and we refer to the combined system as "ReEnact". Third, the synthesis of Visual Narratives themselves as a way of summarising video footage leveraging the bi-directional sketch-pose mapping of Chapter 4 to perform explicit human pose estimation and so visually summarise dance performance fragments as a sequence of stick-men. We use these visualisations within an optional user-interface to enable interactive refinement of the sequence created via ReEnact.

## 5.1 Introduction

Visual Narrative (VN) queries accepted by the search algorithms of previous chapters were restricted to a single action. In this Chapter we explore the extension of this idea to multiple action VNs, and develop a graph based representation for such queries based on the concept of "Motion Graphs" [123] (referred to as 'Move Trees', within the Games industry).

Motion graphs are used to model the space of plausible pose sequences for animated characters, and are typically built from performance capture data (for example skeletal motion

capture data). A sequence of source motion capture data is analysed by comparing every frame to every other. Frames containing similar poses are flagged as 'transition points'. Such frames become nodes in a graph structure. A pair of transition points exhibiting similar poses are linked via strings of additional nodes (frames) in a graph, each connected via directed edge in the graph that link an earlier frame to a later frame. It is then possible to execute a non-linear play back of the source sequence, by walking over the graph and rendering each frame as it is encountered. In effect, frames of the source sequence are seamlessly concatenated in a new order, defined by the graph path, to produce a synthetic sequence containing plausible movement. Although originally proposed for character animation using frames of motion capture data, motion graphs have been extended to other domains including direct application to 2D video frame data — so called 'Video Textures' [181].

This Chapter proposes a modified form of motion graph representation, for the purposes of both searching and synthesising video specified by a VN. To search video, we identify nodes (frames) within a motion graph that closely match each sketch within a VN. The task of finding the video sub-sequence that closely matches the sketch sequence of VN becomes that of finding the 'best' paths across the graph that link the matched nodes in the correct order. In order to synthesise video from a VN the process is similar; we identify nodes that match each sketch within the VN and find the 'best' paths to connect these under a set of user-specified constraints. Video can be synthesised by rendering frames along these paths. We discuss the path constraints, optimality criteria and specifics e.g. topology of the motion graph-like representation under-pinning both of these algorithms later within this chapter.

We continue to focus on the domain of dance performance, using the archival dance footage described within Sec. 4.1.1. The duration of these clips, and the diversity of pose sequences available within them, provide a rich dataset for developing and evaluating our algorithms. Consequently we rely upon the specific human pose matching techniques of the previous chapter to match individual sketches within the VNs, and this chapter may be considered an extension of that basic technique in the two complementary directions of search and synthesis.

The technical contributions of this Chapter are therefore three-fold:

1. **VNs for Choreographic Synthesis** Novel algorithm and representation for the synthesis of video-realistic choreographic footage from a sequence of sketched dance poses, interspersed with dance actions (e. g. jump, twirl, etc.). Synthesis is performed by seamlessly stitching together video fragments from archival footage to produce new choreography in the style of existing footage. In many cases the creation of such sequences would be impossible due to unavailability of historic costumes or the performers depicted in the footage. We refer to this system as "ReEnact". ReEnact for the first time combines both Motion Graphs path optimisation with sketched sequence of key

framed poses identified using our SBVR pose search, and user specified actions linking the sketched poses.

2. **Interactive Choreographic Synthesis** Extending the basic ReEnact contribution, we introduce the ability cluster and visualise multiple alternate paths through the motion graph short-listed via our search algorithm. To perform the visualisation we leverage the bi-directional mapping described in Sec. 4.4.4 to represent each motion fragment visually as an abstract sequence of stick-man poses, so affecting a form of video summarisation. These summaries are clustered hierarchically before being presented to the user. Introducing a user interface to allow for inspection and selection of desired movement fragments allows for greater degree of interactive control over the synthesised video, and affects a form of relevance feedback by placing the user in the loop.

3. **VNs for Video Sequence Retrieval.**

   Searching for sub-sequences within video is a challenging problem. Prior approaches have explored graphical models such as linear dynamical systems [53] or Markov Models [132] to solve this problem with computationally expensive inference steps. Observing a strong relationship between key-frame based synthesis and key-frame based search, we leverage an adapted form of the Motion graph representation to re-purpose the optimal path finding typically used to identify motion fragments for concatenative synthesis to the problem of finding relevant motion fragments in video retrieval. Since the ranked list of video fragments results often exhibit significant temporal overlap, we introduce a filtering step to ensure useful diversity in the results presented to the user.

## 5.2 Visual Narratives for Video Synthesis

We now describe a system for sketch based choreographic design using VNs as a storyboard through which the desired choreography is specified. Fig. 5.1 illustrates the work-flow for the system. The user draws a free-hand sketch of a human pose, which is dropped on to a time-line. This process is repeated for several 'key poses', and the interface provides for manipulation and re-ordering of the key poses as the user works. The user is thereby able to specify a sequence of key poses that the choreographic sequence must pass through in the user-prescribed order. The user also specifies intermediate actions that link each key pose; for example, a twirl may be specified to transition from one key pose to the next. A predefined vocabulary of actions is available within the user interface, which may be dragged and dropped between the key poses to provide the user control over the choreography. The user may also specify a desired duration for these actions. Having fully specified the timeline in this manner, the ReEnact system then searches through a single archival performance

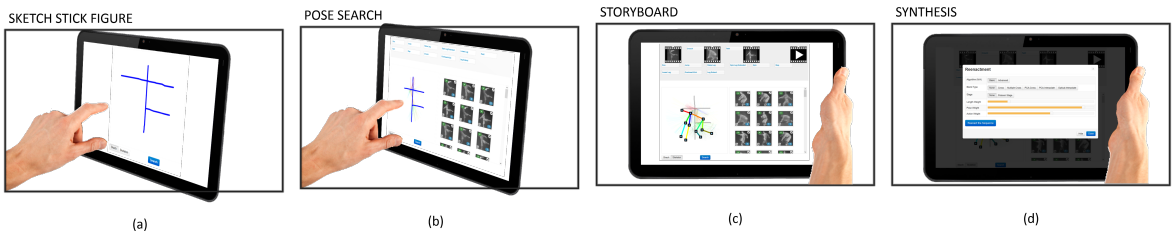



**Figure 5.1** Specifying a choreographic sequence using our VN based interface. A set of poses are sketched (a-b) and placed on a timeline interspersed with actions (c) and timing preferences (d) that drive the motion graph optimisation process to synthesise a novel video. Web interface visualised on a tablet

video to find suitable video fragments that can be seamlessly stitched together to produce the desired choreographic sequence.

We first describe our approach for constructing a motion graph (subsec. 5.2.1) from the dataset of subsec. 4.1.1. We describe an algorithm for finding the path across the graph (subsec. 5.2.2) maximising the similarity of poses within the VN whilst factoring in action and time constraints.

### 5.2.1 Graph Construction

A *motion graph* [123] is constructed 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. 5.3). Random walks across the graph could generate novel sequences in perpetuity (as with [181]); however our system plans paths across the graph to produce user-guided output.

The motion graph is constructed from a single, long, video of a dance performance. Although our matching technique is not limited in principle to a single sequence, it is necessary to restrict the data in this manner so that concatenation of frames results in a visually seamless novel sequence.

Using the technique of subsec. 4.4.1 a manifold is constructed in $\mathcal{D}$, using video pose descriptors computed from the pixels within the bounding box that encloses the performer (subsec. 4.2.2). Recall that each video frame represents a point on a manifold embedded within $\mathcal{D}$.

Transition points are identified by exhaustively comparing video descriptors via eq. 4.12 from all pairs of frames of the video, i. e. computing the geodesic distance between frames, and retaining those above a similarity threshold as transition candidates. This distance comparison is recorded within a symmetric matrix, and smoothed using an isotropic filter to penalise

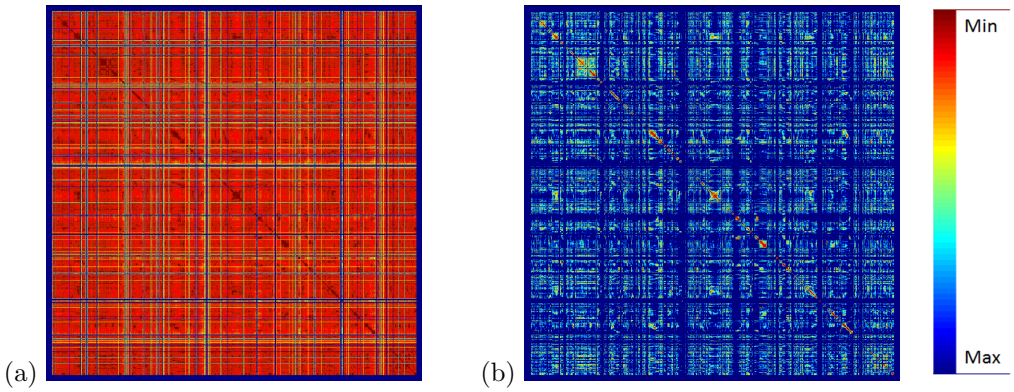



**Figure 5.2** (a) Geodesic distance between all frames in 'Blueprint'. (b) Clipped distance matrix highlighting the transition frames containing closest pose similarity.

local temporal incoherence in pose. Fig. 5.2a visualises the results of the distance comparison for video 'Blueprint'. Fig. 5.2b visualises only the selected transition points (below threshold) and their relative similarities. The non-zero elements of this clipped transition matrix form candidate transition points for the video.

#### 5.2.1.1 Motion based filtering

As visual dissimilarity may be observed in video even in the presence of similar poses, optical flow [32] is used to calculate the visual dissimilarity of frames at candidate transition points (i. e. non-zero points within the clipped distance matrix). The magnitudes of flow vectors within the performer bounding box are summed and transition points discarded that exceed a threshold.

During synthesis, concatenation of frames with similar poses yet mismatched motion directions will not result in a visually seamless transition. Given each candidate transition point, the direction of performer motion is computed for each frame of the pair. This is achieved by averaging bounding box centroid motion over 10 frames leading into and out of the transition point. If motion vectors differ above a threshold, the candidate is discarded.

The remaining candidate transition nodes are used to form the motion graph. First, nodes are added to the motion graph for each frame. Temporally adjacent frames are linked with a directional edge from the earlier to the later frame. The weight of the edge is set using eq. 4.12 to measure pose similarity. Additional edges are then introduced to link the pair of frames comprising each transition point with an edge weight defined by eq. 4.12. This completes the motion graph construction. However the graph representation used during video synthesis is not based purely upon this graph, but a derivative as we now explain.

#### 5.2.1.2 Virtual nodes

At query-time, when the VN has been defined by the user, a graph representation is formed using the motion graph. For each pair of key 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 eq. 4.13). The successive (end) key pose is added as virtual sink node with similarly defined connectivity.

This sink node also becomes the source node on a second copy of the motion graph, which serves as this start pose and the subsequent end pose (Fig. 5.3 illustrates). Thus for $k$ keyposes, $k-1$ copies of the motion graph and chained together by virtual source and sink nodes.

Frame-to-frame edge weights in the motion graph are computed via eq. 4.12, and sketch-frame weights via eq. 4.13.

In addition we require frames to be labelled to indicate the likelihood each of the 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. [223]) using our pose descriptors ($\mathcal{D}$) as a basis for activity classification.

### 5.2.2 Video Path Optimisation

A shortest-path optimisation is used to find the optimal route passing from 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 ($C_{Time}$):

$$C = w_p C_{Target} + w_a C_{Action} + w_t C_{Time}, \tag{5.1}$$

with user-specified weightings $\{w_p, w_a, w_t\}$ for pose, action and time respectively, which are configurable through the UI as shown in fig. 5.1.

Pose similarity encoded by term $C_{Target}$ is analogous to edge weights accumulated in the classical shortest path algorithm. In the case of frame-to-frame moves within the motion graph (black edges in Fig. 5.3) similarity is determined by geodesic distance between the two frames across the manifold $\mathcal{D}$ (eq. 4.12). In the case of edges between the virtual nodes (start/end poses) and a frame in the motion graph (magenta edges in Fig. 5.3) the cost is determined by the combined probability (i.e. overall similarity function) of our pose retrieval algorithm (eq. 4.13). Optionally, the interface allows for retrieved video frames rather than

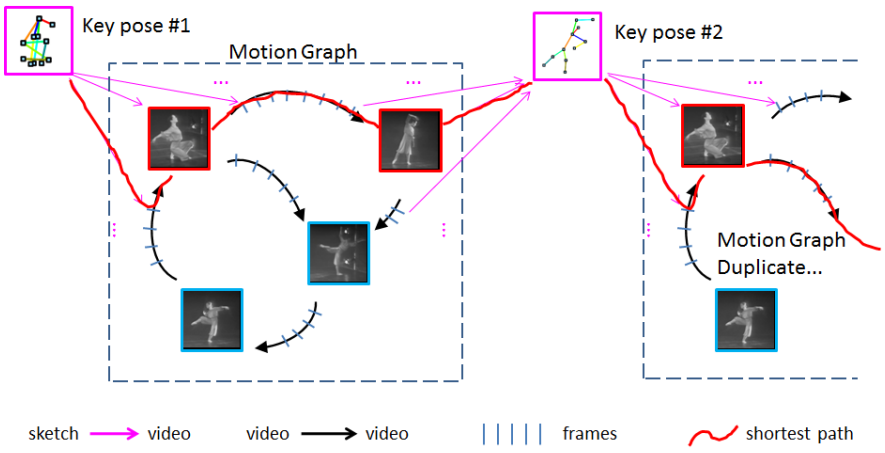



**Figure 5.3** Video synthesis using the graph representation. A directed graph is constructed from video fragments comprising sequential blocks of frames (blue marks on edges) linked seamlessly at transition frames (blue nodes). Sketched key poses (magenta nodes) are added as virtual nodes linking copies of the motion graph. The path with lowest cost (red), by eq. 5.1, from first to last key pose yields the new choreographic sequence.

fresh sketches to create the key poses e. g. to make use of previous search results in a video synthesis tasks. In such cases the virtual nodes are frames and so the attached edge weights are defined via eq. 4.12 rather than eq. 4.13.

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 locally 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:

$$C_{Action} = \frac{1}{k-1} \sum_{i=1}^{k-1} |A(l_i) - A(q_i)| \,. \tag{5.2}$$

It was found that a better similarity measure is to use a soft-assignment of action. Therefore we adapt eq. 5.2 to incorporate similarity of pose, so *run* and *walk* are more similar than *run* and *twirl*, these weights are set empirically expressed as $\mathcal{W}$. The soft weighted action cost is given by:

$$C_{Action} = \frac{1}{k-1}\sum_{i=1}^{k-1}\frac{1}{|A|}\sum_{m=0}^{|A|}\mathcal{W}_m\left|A_m(l_i) - A_m(q_i)\right|. \tag{5.3}$$

The temporal cost $C_{Time}$ is derived from a count of the number of frames on the proposed path. The absolute difference between 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.

$$C_{Time} = S(\sum_{i=1}^{k-1}\big||l_i| - L\big|). \tag{5.4}$$

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

$$\mathcal{S}(x) = x^2(3-2x). \tag{5.5}$$

### 5.2.3 Compositing and Rendering

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 stage locations 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 rendered onto the stage using gradient domain (Poisson) blending [160], which is applied only within the region of the performer bounding box. The silhouette used to compute the video descriptor for a frame (Sec. 4.2) is used to determine the foreground mask during the compositing.

Additionally simple cross-fading is applied at the points of transition between video fragments to mitigate remaining visual discontinuity in playback. For a transition point comprising frame pair $\{I_t, I_{t'}\}$, the frames leading into the transition i. e. $I_{t-5}, \ldots, I_t$ and $I_{t'-5}, \ldots, I_{t'}$ are averaged on a per-pixel basis (and a similar process is performed on the lead-out from the transition).

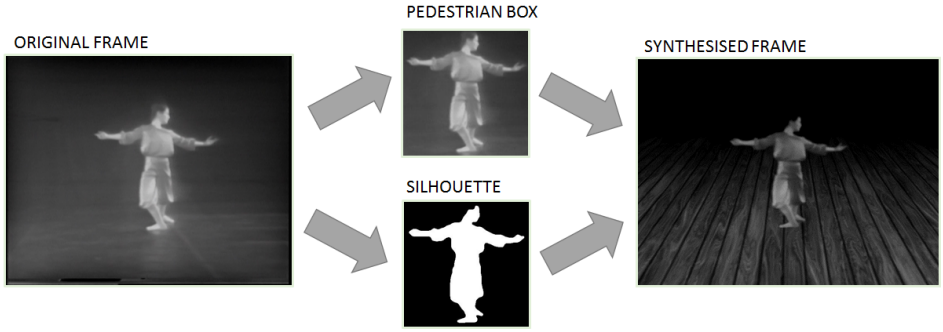


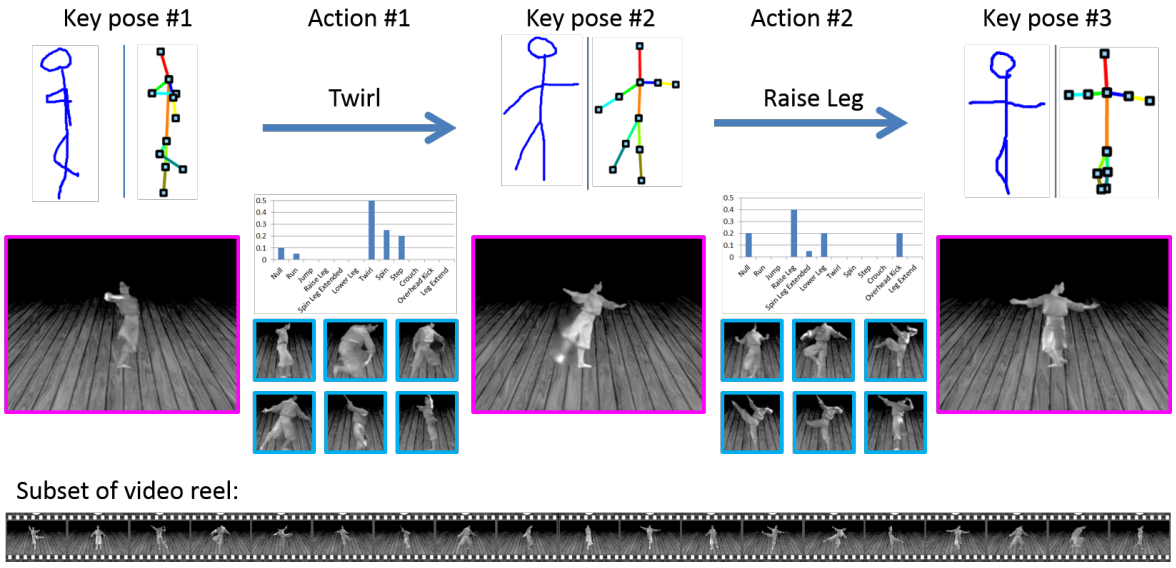

**Figure 5.4** Compositing and rendering. A video frame is synthesised via gradient domain compositing within the performer's bounding box. The silhouette obtain during the video pose descriptor extraction of sec. 4.2.1 is used as the foreground mask during the compositing. The performer is composited on to an infinite stage that scrolls with the global motion of the synthesised performance.

### 5.2.4 Evaluation

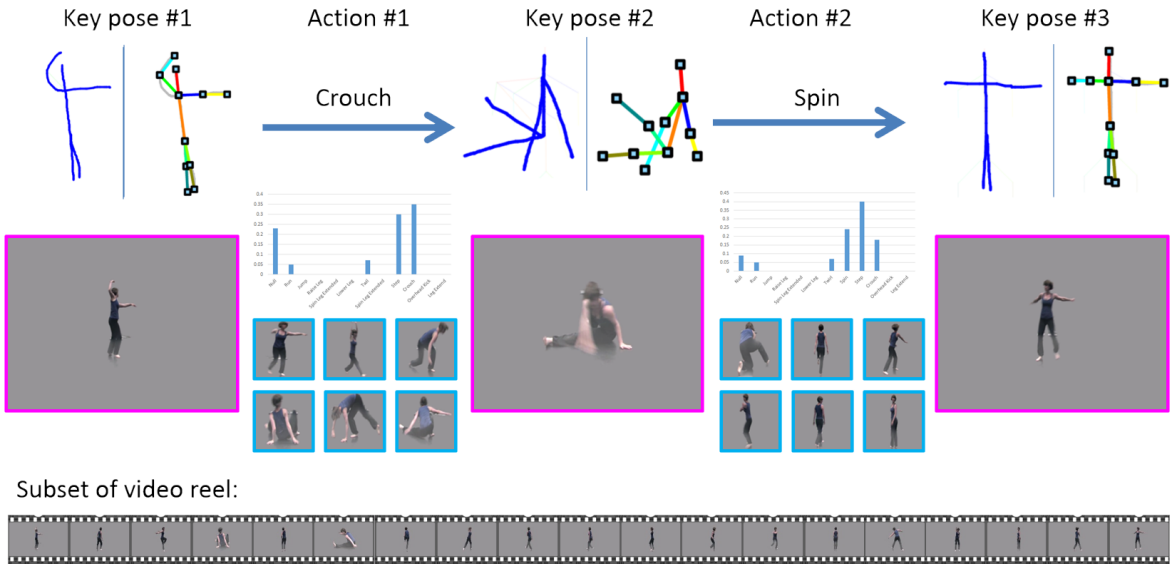
We demonstrate the ability of ReEnact to create new choreographic sequences from sketched key poses, over both the training ("Blueprint") and test ("Expressive") videos from our dance performance dataset. Fig. 5.5 showcases the VN queries and output videos synthesised.

For each sequence showcased in Fig. 5.5, the best matched key frame for a sketch pose is indicated in purple. This corresponds to the frames connected to the virtual nodes in the graph via the shortest path computed for the VN query. The frames in blue are regularly sampled from the interval between those key frames in the synthesised sequence to illustrate the actions of the performer in the synthesised output. For illustration, the probability distribution of actions between the key frames is displayed as a histogram indicating how closely the synthesised video content matches the actions specified. We demonstrate synthesis on a textured stage (fig. 5.5a) as well as onto a uniformly coloured background (fig. 5.5b) which can be aesthetically preferable if the performance video contains complex background that proves challenging for the gradient-domain compositing.

Although durations for each interval between key poses 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 5.5 the synthesised 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. The user interface exposes these weights to the user in the form of sliders. Fig 5.5b highlights how this weighting may result in spurious pose matches due to the complexity of the video. In this case Expressive exhibits many close-




(a) Query VN and corresponding video synthesised from 'Blueprint'




(b) Query VN and corresponding video synthesised from Expressive

**Figure 5.5** Results of video synthesis for two sample videos a) 'Blueprint' and b) 'Expressive'. For each video — Top: Query VN composed of three key poses and two actions connecting these. Bottom: Frames identified on the shortest path corresponding to the key poses (purple) and a sampling of frames along the path connecting these key poses (blue). Histograms indicate the probability distribution of actions within the frames along the shortest path between the respective key frames.

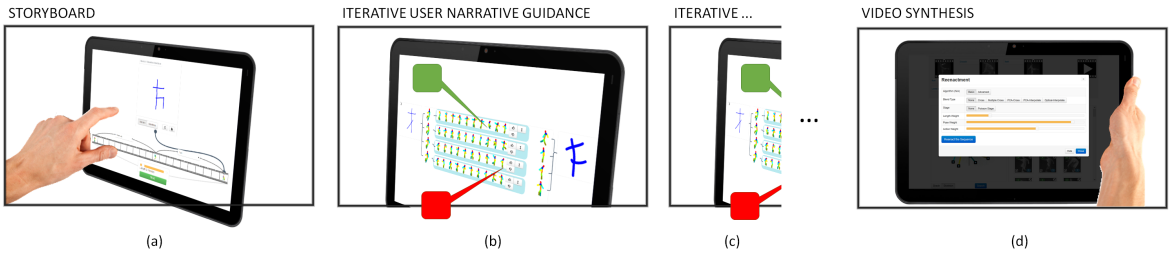



**Figure 5.6** Modified Storyboard for the process of interaction. User sketches stickman into a timeline (a), these stickmen are used to identify a set of paths through the motion graphs that can be selected as positive or negative based on a summary stickmen narrative (b-c) after which a new synthetic video is created (d). Web interface visualised on a tablet

to-body poses with relatively short limbs, which challenges the pose matching algorithm of Chapter 4.

New sequences are typically generated in under 10 minutes, though the majority of this runtime is spent on the gradient-domain compositing without which a sequence may be generated in under one minute.

## 5.3 Interactive Visual Narratives for Video Synthesis

For a given VN, ReEnact will synthesise a single video based on the optimal (shortest) path identified over the graph. However, in practice there may be multiple routes over the graph that yield similar total path costs via eq. 5.1. A small departure from the pose specified within a sketched key frame might result in a completely different path being promoted to the 'best', containing different action sequences. Iteratively fine-tuning the user weights can partially mitigate this issue, however precise control is difficult via the weight sliders leading to unpredictable results for users.

To mitigate these problems, we present a modified form of the ReEnact in which the user may interactively select from a set of promising paths across the graph generated from their VN, to guide the creation process.

Under the proposed interactive system the workflow is modified as follows (Fig. 5.6). As before, the system invites the user to draw free-hand stick-men to indicate a set of key poses. The user drags the poses on to a time line to indicate the relative duration of each interval between key poses. A target total duration for the synthetic video is also specified (thus providing estimates for the durations of each key pose interval). In contrast to the prior system which solved globally and automatically for all intervals in one step, under the modified ReEnact system the video is synthesised one interval at a time, with the user invited

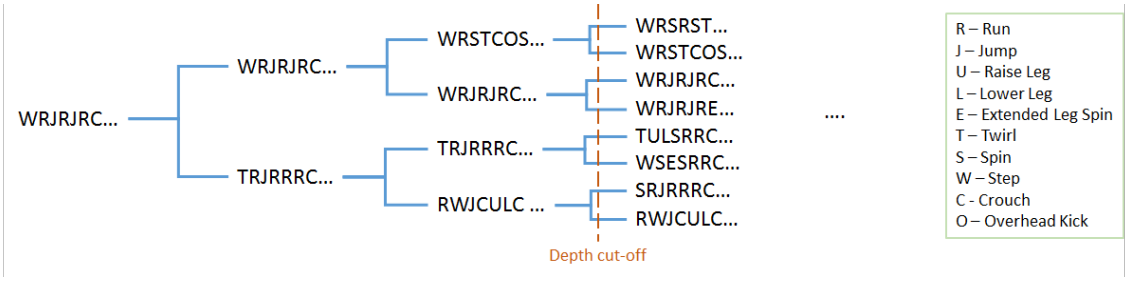



**Figure 5.7** Representation of the output of Agglomerative clustering, in a binary tree format. The tree is cut at a depth of 3, to present to the user.

to score preferred sequences for the interval as a form of relevance feedback (RF) to inform the video synthesis process.

### 5.3.1 Path Clustering

To solve for a given key pose interval, multiple putative paths are generated between the pair of virtual source and sink nodes corresponding to the key poses on either side of the interval. The path optimisation technique of Sec. 5.2.2 can be directly applied over this sub-graph. We identify the 200 paths with shortest distance, building set $\mathcal{R} = \{P_1, \ldots, P_{200}\}$. A limit of 200 is chosen for reasons of both computational efficiency (discussed later) and to avoid visually overloading the user.

We cluster the short-list of putative paths to enable them to be browsed by the user. The intention is to allow the user to quickly navigate the space of short-listed paths and mark any of them as 'preferred' or 'non-preferred' as a form of relevance feedback to the system.

We use the action labels associated with each frame as the basis for clustering the putative paths. Recall that each frame is associated with a probability distribution indicating the likelihood of one of several pre-defined actions occurring at that time. The label of the most likely action may be determined trivially from the maximum of this distribution for any given frame. A putative path $\mathcal{P}_i$ may therefore be interpreted as a sequence of actions (one per frame). Changes in action label over time can be identified, and in addition to the initial action label, used to summarise the action sequence, e. g. $A(\mathcal{P}_i) = \{Run,\ Walk,\ Jump,\ Walk, \ldots\}$ simplified as sequence of tokens $\{RWJW\ldots\}$. Fig. 5.7 (right) summarises the dictionary of 10 action labels used within our system. Note that $A(\mathcal{P}_i)$ is a variable length string of at least one action token.

Agglomerative clustering [120] is performed on all strings $A(\mathcal{R})$ to group them for presentation. The algorithm requires only a (non-metric) definition of distance between a pair of data items in order to make grouping decisions. The Levenshtein string edit distance enables

us to define the similarity between a pair of variable-length action strings. Under this distance metric, the minimum number of edits (token insertions, deletions and substitutions) to transform one string into another is the distance between them. We fix the costs of insertions at a constant 0.5, and penalise substitutions using a matrix of weights empirically defined to penalised misclassification, e. g. of a twirl to a jump. These weights were also used within Sec. 5.2.2 to compare the different actions associated with frames.

Clustering proceeds over a distance matrix $C$ describing the edit distance between all paths $\mathcal{R}$:

$$C = \begin{bmatrix} edit(\mathcal{P}_1, \mathcal{P}_1) & \dots & edit(\mathcal{P}_{200}, \mathcal{P}_1) \\ \vdots & & \vdots \\ edit(\mathcal{P}_{200}, \mathcal{P}_1) & \dots & edit(\mathcal{P}_{200}, \mathcal{P}_{200}) \end{bmatrix}. \tag{5.6}$$

Agglomerative clustering is a bottom-up approach, treating all elements in $C$ as independent clusters then merging the nearest two clusters, this process is recursively applied until all points belong to a single cluster. Agglomerative clustering is described Alg. 2, where C is the precomputed distance matrix of eq. 5.6 and $sim(i, m, j)$ is the similarity of cluster $j$ with putatively merged clusters $i$ and $m$.

**Algorithm 2** Agglomerative clustering algorithm

```
1: procedure AGGLOMERATIVE(C_{1...N,1...N})
2:     A ← []
3:     while ⟨i, m⟩ ← argmax_{{⟨i,m⟩ i≠m∧I[i]=1∧I[m]=1}} C[i][m] do
4:         A.append(⟨i, m⟩)
5:         for j = 1 to N do
6:             C[i][j]=sim(i, m, j)
7:             C[j][i]=sim(i, m, j)
8:         end for
9:         I[m] ← 0
10:    end while
11:    return A
12: end procedure
```

The use of a hierarchical clustering declutters the interface, grouping very similar paths and enabling users to coarsely flag large numbers of paths to be of interest (or disinterest) as well as 'digging into the data' to fine-tune ratings along their preferred path (Fig. 5.7). Although well-suited to our goal, agglomerative clustering has a high computation cost at $O(n^3)$ in the number of paths. This motivated our restriction of the approach to just a couple of hundred paths, as in principal all paths over the graph might be available to choose from.

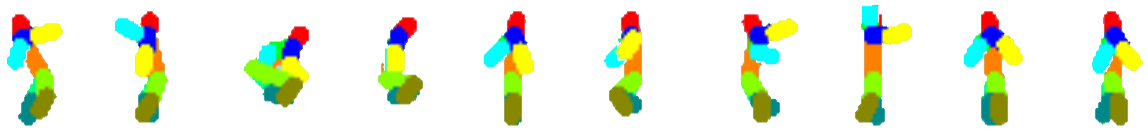


**Figure 5.8** A sequence of stick figures synthesised from the 'Blueprint' video. The path contains a crouch and a twirl action within the sequence, which can be readily observed in this summarisation of the video fragment.

### 5.3.2 Path Visualisation

A useful property of our manifold based approach to pose search (Sec. 4.4) 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 $i^{\text{th}}$ training frame in set $n_t$, is available via eq. 4.12. 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:

$$s(n_x) = \frac{1}{|n_t|} \sum_{\forall n_i \in n_t} s(n_i) \mathcal{N}\left(1 - p(n_i|n_x), \sigma\right). \tag{5.7}$$

where $\mathcal{N}$ is a Gaussian distribution with empirically set standard deviation $\sigma$. It is worth noting that generated descriptors do not imply any limb length, therefore an exact estimation of the skeleton is not possible, so we opt to use a template skeleton and adjust this. We demonstrate this inference of skeletons in Sec 5.3.4. Frames within each path clustered within the binary tree may be visualised as a sequence of stick-men figures. To reduce clutter we represent the path using a set of up to 10 frames sampled at each instant of activity change (i. e. that generated a token within the string used to cluster the path). Fig. 5.8 illustrates such a path visualisation; a stick figure has been generated at the beginning of each action summarising the content of the video sequence.

### 5.3.3 Relevance Feedback

Fig. 5.9 illustrates the interface through which the user is asked to provide feedback on paths presented within the hierarchy, selecting positive (*thumbs up*) and negative (*thumbs down*) paths from those on display. When selecting feedback for a path the user is implicitly stating that all examples below the path within the binary tree structure are also positive (or negative) so allowing a large selection of positive and negative training data to be collected quickly.

We adopt a similar approach to Sec. 3.6, learning a model of user preference by training a linear SVM model over the user feedback. Recall that the generation of paths is subject to

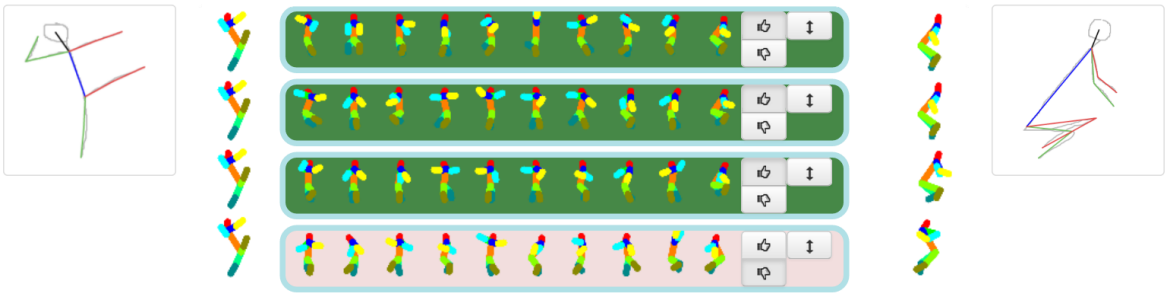


**Figure 5.9** Paths clustered between a source and sink from the query, end poses and their respective query. Interface allows expansion of the binary tree, and selection of positive ('thumbs up') or negative ('thumbs down') ratings for any path.

shortest-path computation over the graph using eq. 5.1, which is in turn dependent on timing, pose and action constraints. Since the former two are fixed, the relevance feedback process boils down to modifying the query action distribution to align more closely with the users' requirements. Since we deal with 12 pre-defined actions, the goal of the relevance feedback is essentially to find the optimal point in $\Re^{12}$ for the query. The action distributions along paths marked as relevant or not, are used as positive and negative training data for the SVM in order to deduce that point.

### 5.3.4 Evaluation

Figs. 5.10-5.11 illustrate one iteration of relevance feedback using the proposed interactive scheme. By contrast Fig. 5.5a-5.5b illustrate the clip generated through the fully automated ReEnact proposed in Sec. 5.2.

Although explicit human pose estimation is not intended as contribution of this thesis, it is nevertheless interesting to observe that reasonable skeletons can be obtained from low fidelity footage where state of the art methods [74, 235] currently fail. We qualitatively demonstrate this property through visual comparison on video frames that fail under these algorithms. Fig. 5.12 contrasts the skeleton obtained from a single frame using the public implementations made by Ferrari et al. [74] and Yang et al. [235] 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.

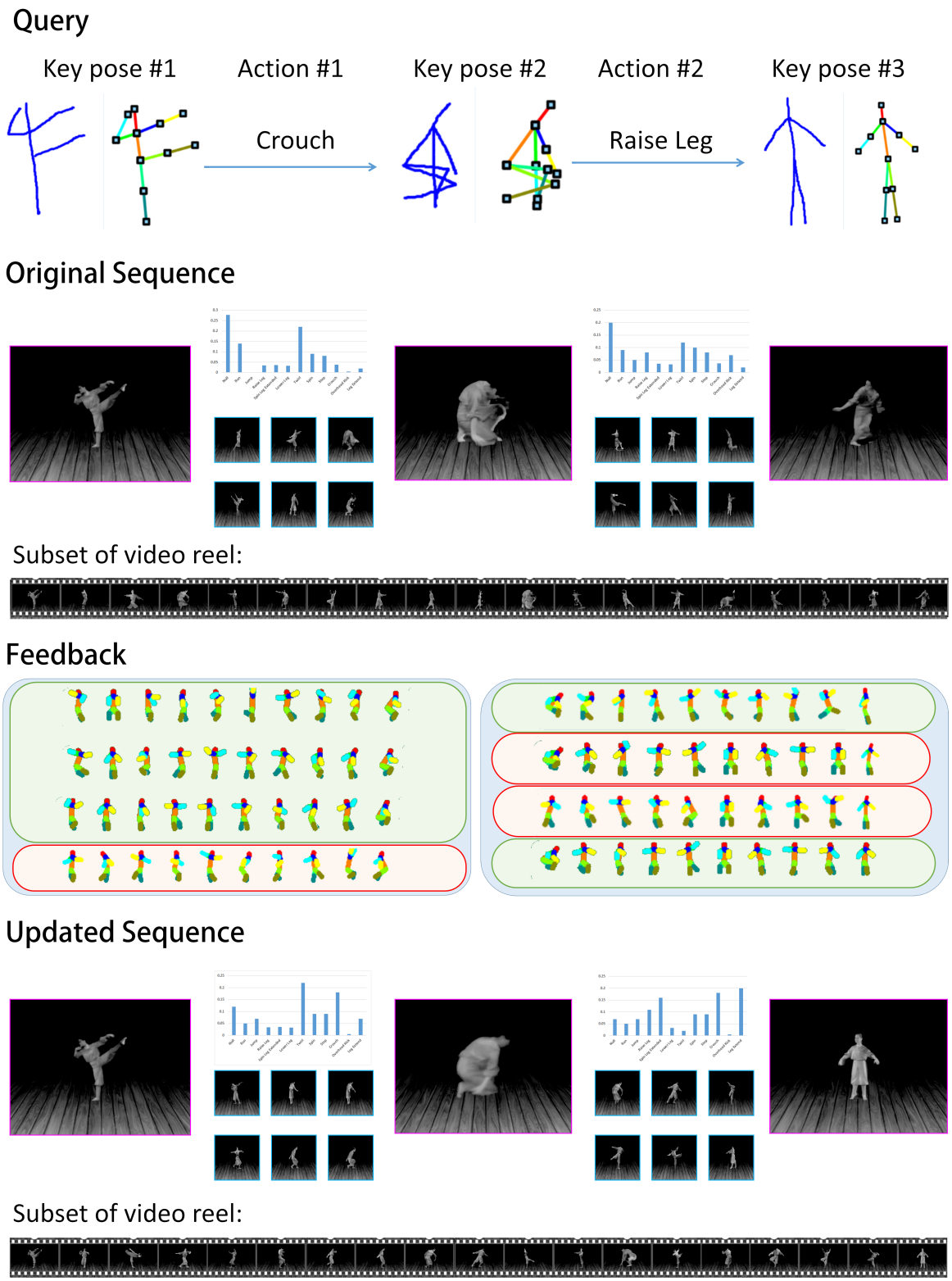



**Figure 5.10** Video synthesis over **Blueprint** – A sample query storyboard composed of three key poses and two actions connecting. Showing the original sequence generated from 5.2. A subset of selected paths provided as feedback by the user for both sections of the storyboard. Bottom, the updated video based on the feedback provided by the user.

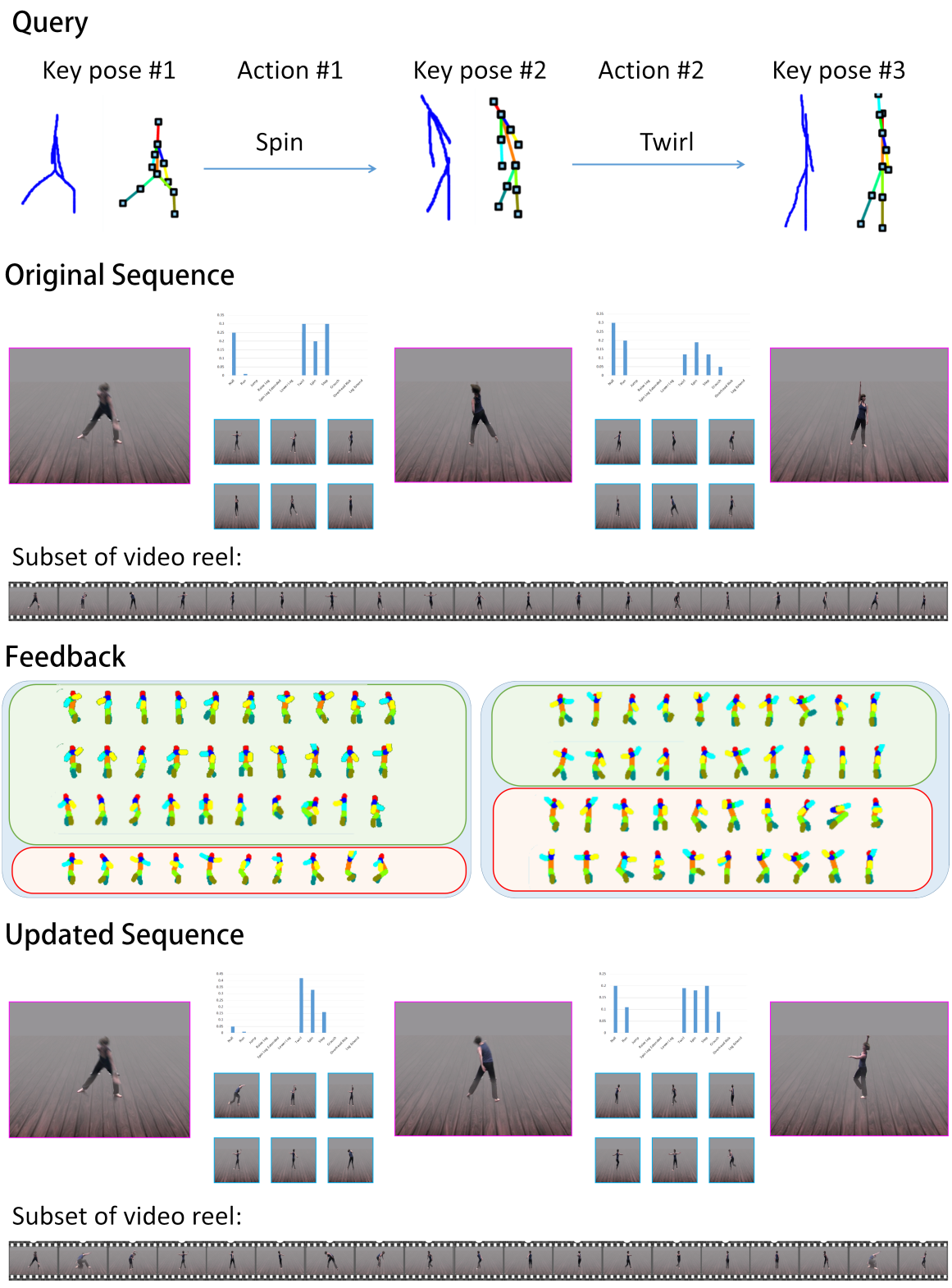



**Figure 5.11** Video synthesis over **Expressive** – A sample query storyboard composed of three key poses and two actions connecting. Showing the original sequence generated from 5.2. A subset of selected paths provided as feedback by the user for both sections of the storyboard. Bottom, the updated video based on the feedback provided by the user.

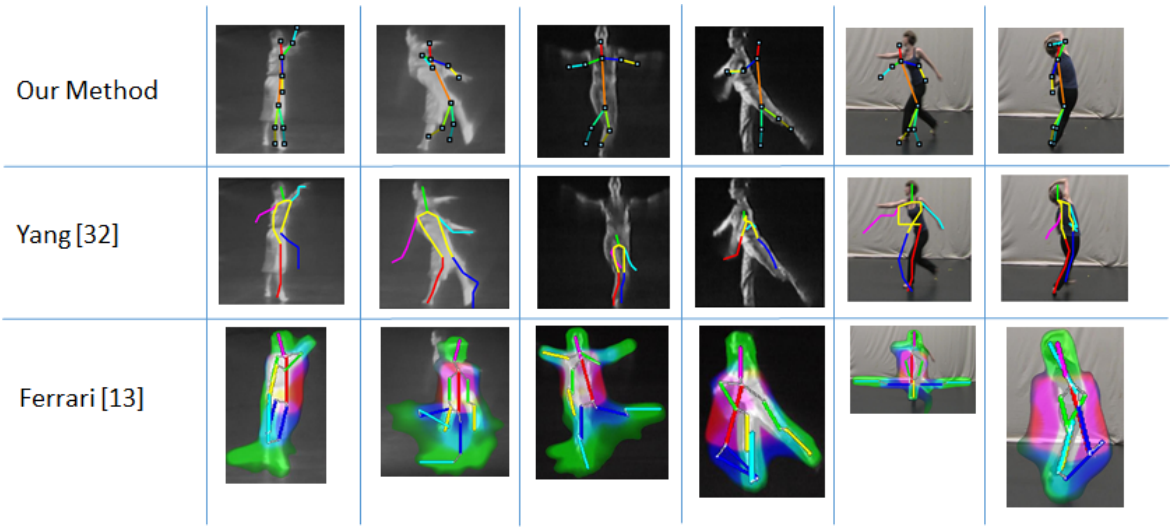



**Figure 5.12** Comparison of Articulated skeleton estimation between our method, Yang [235] and Ferrari [74] over Blueprint (2x left), ThreeD (2x middle) and Expressive (2x right).

## 5.4 Graph based Visual Narratives for Retrieval

The graph representation that enables concatenative video synthesis may also be applied to SBVR, i. e. to retrieve contiguous sub-sequences of frames from a video, matching a VN. The form of the VN query is identical to that used for video synthesis (Sec. 5.3); a sequence of sketched key poses interspersed with desired actions and relative durations of those actions. The user interface for VN specification is therefore identical however the manner in which results are displayed is modified (Fig. 5.13), since an additional user parameter $\Gamma$ to constrain temporal overlap in the retrieved video sequences is also introduced (discussed in subsec. 5.4.2).

### 5.4.1 Graph Construction

As with the ReEnact system, a motion graph is constructed to represent valid frame sequences that may be presented to the user. Each node within the graph corresponds to a video frame, and directed edges connect temporally adjacent frames (earlier frames are linked to later frames). However unlike ReEnact we do not detect or link transition frames. The topology of the graph is restricted to linear playback, since we desire only contiguous subsequences within the video to be returned as results.

At query-time, virtual nodes are again inserted into the graph for each sketched key pose. As was the case with ReEnact (subsec. 5.2.1), successive key poses are attached to the graph as a 'source' and sink' node with connections made to and from nodes in the motion graph

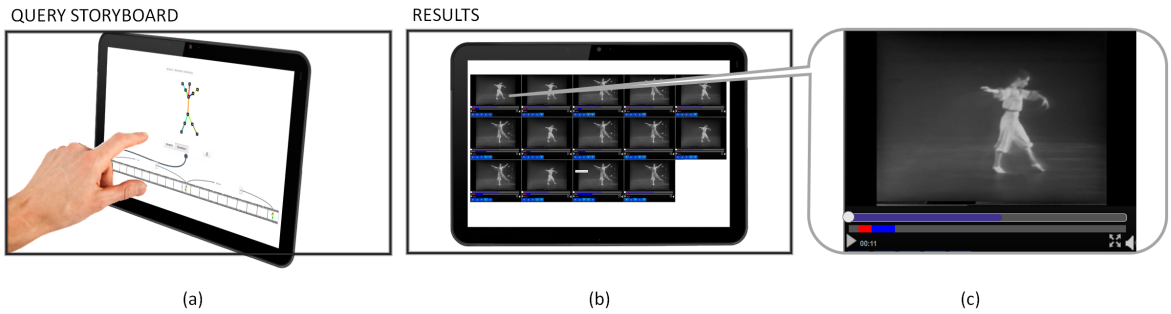



**Figure 5.13** Video Search using a VN query: (a) A sequence of poses interspersed with action constraints drives the search. Top-ranking results are displayed in a grid (b) and may be selected for playback and inspection (c). A blue highlight on the seek-bar indicates the contiguous subsection of a video in the database that was deemed relevant. Web interface visualised on a tablet

respectively. Duplicate copies of the motion graph are 'chained' together using these virtual nodes as before. Fig. 5.14 illustrates the topology of the graph.

The weights on the virtual nodes are set as per Sec. 5.2.1. Frame-to-frame edge weights (i. e. between non-virtual nodes) in the motion graph are computed via eq. 4.12, and sketch-frame weights (i. e. between virtual nodes and the motion graph) are computed via eq. 4.13.

The retrieval of is then reduced to the problem of finding the shortest path across the graph for the first to last virtual nodes. The optimisation cost is similar to eq. 5.1 within ReEnact, but is modified slightly as discussed in subsec. 5.4.2.

The large number of frames typically present within each video can cause unacceptable retrieval times, and so a short-cut is in made practice. Using the action tokenisation algorithm of Sec. 5.3.1 salient instants may be detected within the video at frames where the action classification of the performer changes. Since key poses specified by the user tend to coincide with such salient instants, we can significantly reduce the complexity of the path optimisation by connecting only these salient frames to the virtual nodes — instead of every frame to the virtual nodes.

### 5.4.2 Temporal Motion Graph Optimisation

We use the same path optimisation procedure and cost eq. 5.1, with the caveat of not having transition cost to consider (since the transition costs of temporally adjacent frames are likely to be low and can be disregarded).

Despite reducing the connectivity of the graph to salient frames, numerous near-duplicate results are returned in the top-ranked (i. e. shortest) paths discovered through the graphs. Whilst correct behaviour, the utility of a search system that returns many minor variants of

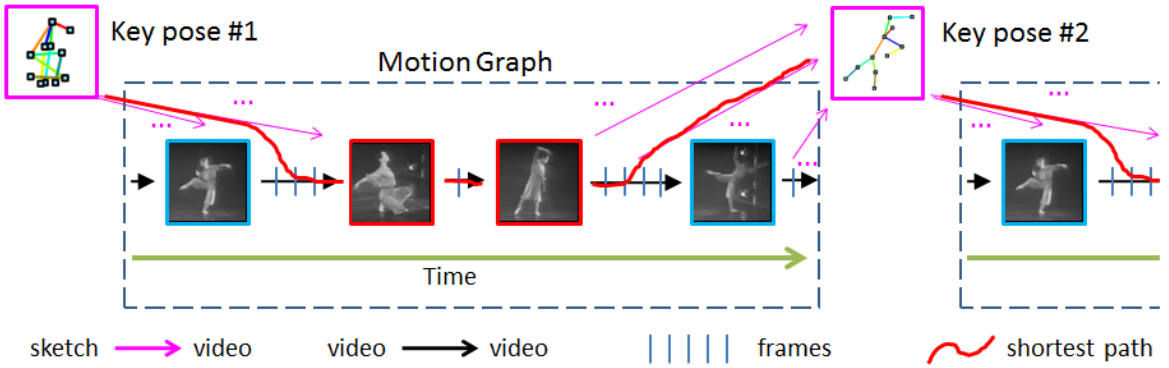



**Figure 5.14** Video Search via the graph representation. A directed graph is constructed from the video comprising sequential frames (blue marks on edges) linked at salient instants (blue nodes). Sketched key poses (magenta nodes) are added as virtual nodes that connect duplicate copies of the motion graph. The virtual nodes are linked to the nodes representing salient frames. The path across the graph — from first virtual source node to last virtual sink node — with lowest cost (red), by eq. 5.1, from first to last key pose yields the most relevant video sub-sequence.

essentially the same path, is limited in the context of video search. We therefore introduce an extra user parameter that constrains the level of temporal overlap allowable between results. We measure the temporal overlap of two sequences as a simple ratio; the count of frames both have in common (frame intersection), to the frame count of the union of both sequences (frame union).

The resulting normalised score is compared against parameter $\Gamma$ in a greedy strategy. The top (shortest) path result in is admitted to the results set $\mathcal{R}$. Next-best results are compared against those in $\mathcal{R}$ to ensure overlap is less than $\Gamma$ prior to inclusion in $\mathcal{R}$:

$$\forall i \in \mathcal{R} > 0 = \begin{cases} 1 & \text{if } \frac{\mathcal{R}_i \cap \mathcal{R}_{i-1}}{\mathcal{R}_i \cup \mathcal{R}_{i-1}} > \Gamma \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

where $\mathcal{R}$ is the set of results and $\mathcal{R}_i$ an individual result (contiguous video fragment). Increasing $\Gamma$ filters more results as the value increases. This can be demonstrated through an example set of results in Fig. 5.14 where the left represents the raw result set, and the other columns represent filtered result sets with increasing value of $\Gamma$.

By reducing $\Gamma$, fewer results are displayed to the user with greater diversity. In the case that only a few paths are originally identified this can result in only a single path being displayed, and such a result is common in practice as $\Gamma \mapsto 0.1$, which we therefore adopt as a lower limit on $\Gamma$, which is controlled via an interactive slider on the results page.

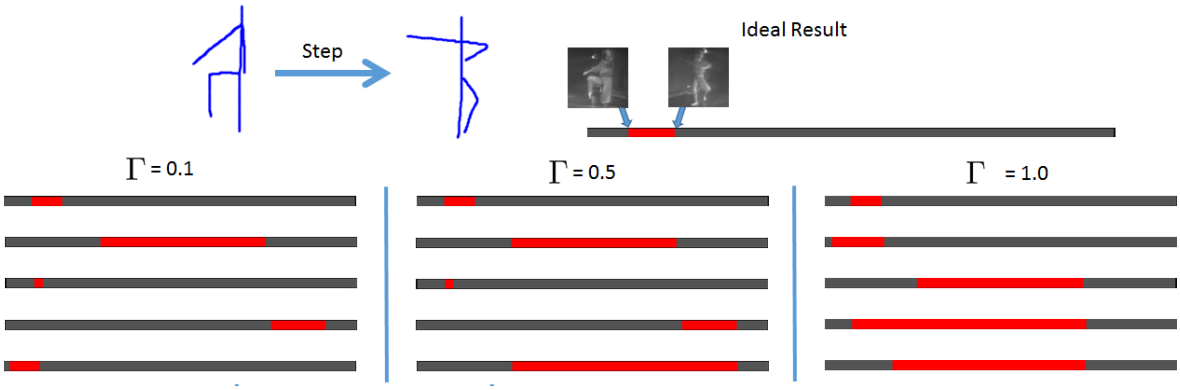



**Figure 5.15** Effect of Γ on constraining the displayed results, by increasing Γ fewer similar results are returned, reducing result count but increasing diversity.

### 5.4.3 Evaluation

We evaluate the dataset using three videos presented in sec. 4.1.1 – Blueprint, ThreeD and Expressive; evaluating results based on a variety of queries and VN lengths.

The nature of video sequence search presents a challenge to evaluation, in that the high specificity of a VN implies very few, if any, occurrences of that sequence in the video database. This can lead to near-binary precision scores which are not informative for analysis of system performance. We therefore adopt a 'softer' definition of relevance in which partial matches contribute fractionally to the precision score. We first explain our methodology (Sec. 5.4.3) then go on to perform a qualitative and quantitative evaluation of the video search system (Sec. 5.4.3).

#### Methodology

Similar to Collomosse et al. [53] we evaluate using a precision measure that awards value to partial matches. Each facet (*pose, action*) of the query contributes 1 to the rank result score, which is then normalised by the number of facets:

$$Score(V,Q) = \frac{1}{|Q|}\sum_{j}^{j=|Q|} relevance(V_j), \tag{5.9}$$

where Q is query VN, and V the returned is a clip. The Precision@k is the cumulative score over the range of the result set $R_i$ with $i = [1,k]$, therefore:

$$P_k(R,Q) = \frac{1}{k}\sum_{i}^{i=k} Score(R_i,Q), \tag{5.10}$$

where the Average Precision (AP) is:

$$AP = \frac{\sum_{k=1}^{n} P_k(R,Q)Score(R_k,Q)}{\sum_{k=1}^{n} Score(R_k,Q)} \tag{5.11}$$

i. e. the AP is the cumulative precision score over R segments normalised by the maximum attainable cumulative precision over those segments. The mean AP (MAP) is the average over these for all queries evaluated.

#### Evaluating Retrieval Accuracy

Fig. 5.5 shows 4 examples of VN queries demonstrating their respective top 3 returned clips as well as the matched poses and interspersed action distribution from within the returned clip; we also show a summary selection of frames highlighting facets of the result as correct or incorrect through green and red borders. For these results we fix $\Gamma = 1.0$, although adjusting this generates results with greater diversity (but lower accuracy). Generally the top match is correct throughout the queries, and as lower rank results are explored, one of the facets become less faithful to the query (Fig. 5.16d).

To perform a quantitative evaluation via the methodology of subsec. 5.4.3, we average performance over 5 queries of varying VN length (2-3 poses). Although it is possible to perform longer queries (e. g. demonstrated in Fig. 5.16d), matching this many facets within a video becomes very challenging as there are many ways in which content can deviate from query. Commonly when performing a VN query there will only be 1-2 exact pose matches within the video. The plot of $\Gamma$ versus precision shows a decrease in performance accompanied by an increase in result diversity (fig. 5.1).

Additionally to fig. 5.17 we compute MAP scores over the different values of $\Gamma$ and their respective videos in Table 5.1.

| | $\Gamma = 0.1$ | $\Gamma = 0.5$ | $\Gamma = 1.0$ |
|---|---|---|---|
| Blueprint | 87.6 | 78.2 | 77.0 |
| Expressive | 92.7 | 82.5 | 75.7 |
| ThreeD | 73.5 | 67.9 | 70.6 |

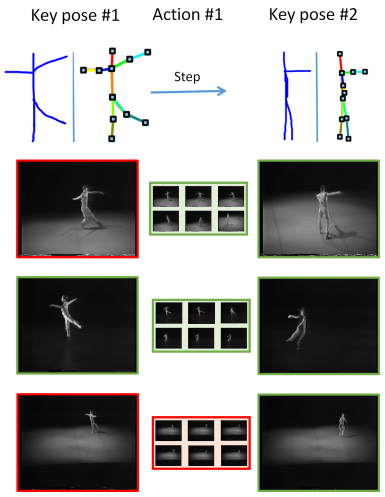
**Table 5.1** Comparison of the MAP performance of the system for various $\Gamma$ values over three videos.

## 5.5 Conclusion

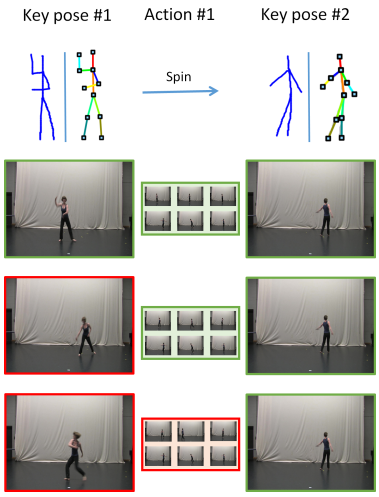
We have proposed a graph-based algorithm for video synthesis that uses a VN to specify the construction of the new video. Video sequences are synthesised by splicing segments of an archive video together under a concatenative synthesis framework. An adaptation of Kovars et al.'s motion graphs [123] is used in combination with key poses from the VN to model the space of all valid pose sequences. Routes through the graph are examined via a shortest-path optimisation to determine the most appropriate video sequence. New clips are generated in under 10 minutes with the performer from the archive footage being stitched onto a synthetic 'infinite' stage of uniform coloured background.

To enable the user to explore several promising routes through the motion graph we propose a modified video synthesis system that integrates the user into the system. Relevance feedback is sourced from the user, allowing them to interactively guide the creation of videos by providing positive and negative feedback on paths. An algorithm was contributed for clustering paths and representing them to the user in a concise way. Once paths have been scored by the user, the optimisation is re-run and the optimal path is then used to create the video. Ultimately this may result in a different action distribution to the VN query initially proposed, but is more relevant to the user and takes into consideration their relative preferences.

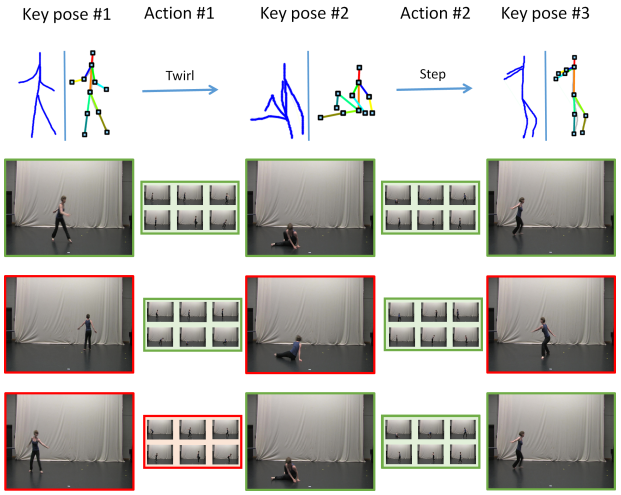
As an alternative to video synthesis we additionally demonstrate that a motion graph can be adapted to perform video sequence retrieval, identifying contiguous segments of video that match a query's poses and actions. We demonstrate an MAP of 88%, 93% and 74% over Blueprint, Expressive, ThreeD respectively. Interestingly we observe that as the VN length increases the complexity of accurately matching within the video often results in one component of the query being swapped for an alternative as opposed to completely different section of video. Although this may not be desirable in all scenarios, through user filtering we demonstrate that only the most relevant results are shown.



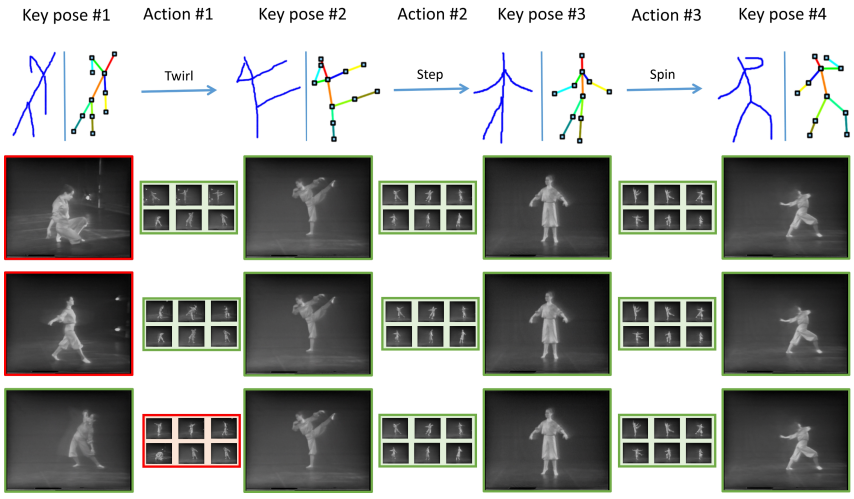

(a) 2 Part Query, from ThreeD




(b) 2 Part Query, from Expressive

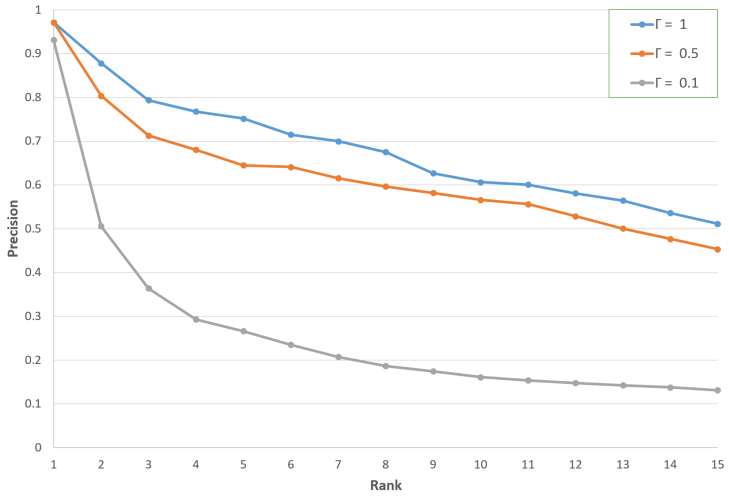



(c) 3 Part Query, from Expressive

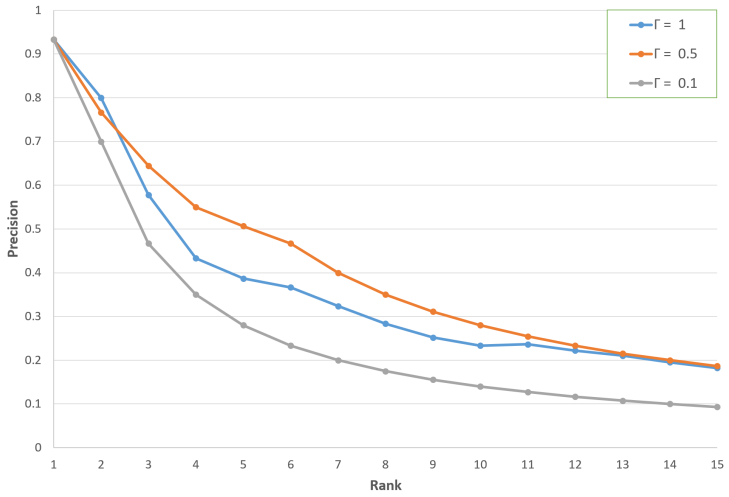



(d) 4 Part Query from Blueprint

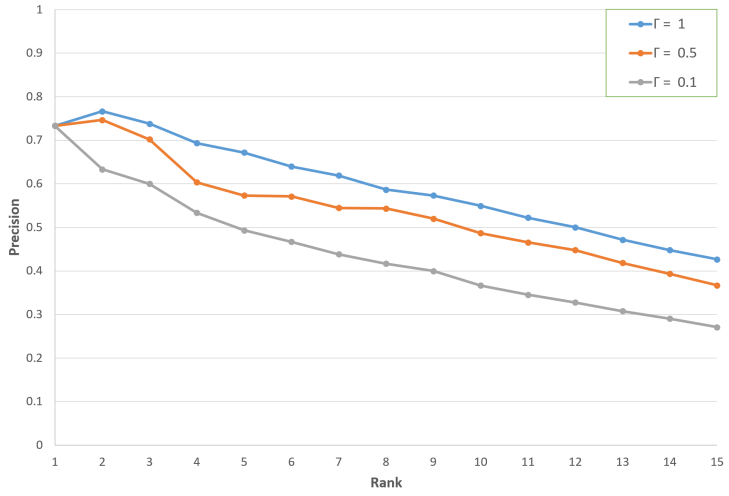
**Figure 5.16** Set of queries and their top 3 results, with correctly matched poses highlighted in green and incorrect ones in red. Additionally a subset of inter-key frames highlighting the action sequence.




(a) Blueprint




(b) Expressive




(c) ThreeD

**Figure 5.17** Variation of the effect of Γ showing Precision @K curve for Blueprint, Expressive and ThreeD respectively.

# Chapter 6

# Conclusion

This chapter summarises the contributions of this thesis, reflecting on the research hypotheses outlined in Chapter 1. Directions for future work are briefly outlined.

## 6.1 Summary of Contributions

This thesis explored visual narratives (VNs) as an interaction modality for searching and manipulating video. Visual narratives are pictorial representations of events comprising a sequence of one or more component sketches. The sketched components depict salient objects and their instantaneous actions. We have shown that VNs can be used to produce efficient and accurate visual search systems, so contributing to the fields of Computer Vision and Information Retrieval — most specifically to the sub-field of Sketch based Video Retrieval (SBVR). We have also shown that VNs can be used in conjunction with concatenative video synthesis algorithms, as a novel means for creating choreography, so contributing to Computer Graphics.

Several research hypotheses surrounding these applications of VNs were outlined in Chapter 1. We now summarise the technical contributions made in each chapter, concluding upon the evidence they provide to support these hypotheses.

### 6.1.1 Multi-modal Video Indexing for Sketch based Retrieval

Chapter 3 presented a novel spatio-temporal descriptor for SBVR. The descriptor encodes colour, shape, motion and semantic cues as well as the shape within the scene background. Little prior SBVR work has addressed multiple modalities within sketch, in particular fusing both appearance and semantic cues (so called 'hybrid SBVR'. Existing approaches that do,

adopt an inefficient 'model fitting' approach in which a sketch is treated as a model that is optimised to fit evidence within each clip in a database. Not only is the optimisation slow (taking many minutes to search a few hundred videos), but such approaches can not scale better than linearly with database size.

Uniquely, we encapsulated these multiple modalities within a video index representation resulting in a highly efficient SBVR system that can not only search several hundred videos in just tens of milliseconds, but also scales sub-linearly with the database size. The high performance of the proposed framework supports the hypothesis:

> [**H1**] Hybrid SBVR can scale to high performance large-scale video search using an efficient index representation.

Hybrid SBVR suffers from ambiguity not only in the sketch depiction, but in the relative priorities of the different modalities of cue. A sketch of a red car moving left contains information on colour, shape, semantics and motion yet the sketch contains no information on the relative importance of each of these attributes; in the absence of a perfect match would the user prefer to see red objects, or cars, or objects moving left? Commonly information retrieval deals with such prioritisations through relevance feedback (RF), the iterative refinement of results through feedback provided by a user 'in the loop' working alongside the system. Due to the slow retrieval times of prior work, RF for SBVR had not been explored previously. We showed that using an ensemble of linear SVMs, RF could be applied to SBVR in order to interactively tailor the results to the user's preference. We showed quantitative performance improvements are possible with our system (i. e. we exceeded the performance of the non-RF based SBVR state of the art), so satisfying hypothesis:

> [**H2**] SBVR accuracy can be improved using Relevance Feedback (RF)

The sub-linear scalability of our system at query time is due to the adoption of an efficient search structure (kd-tree). This structure is applicable because our approach is based on video descriptor matching, rather than expensive model fitting, at query time. However the scalability of our system over general footage is still limited in two main ways. First, video ingestion requires several pre-processing steps that limit the system to short clips, e. g. the detection of foreground and background is based on inter-frame homography estimation which becomes unreliable over long clips. Second, the reliance upon black-box semantic segmentation algorithm limits our approach to dealing with only a few tens of semantic classes. This limitation is likely to be relaxed as semantic classification technology improves. For example, significant advances in automated object classification are being reported very recently using Convolutional Neural Networks. Such technologies could trivially be dropped in as replacements as they mature.

### 6.1.2 Sketch based Pose Retrieval and Estimation

In Chapter 4 we presented an approach for retrieving images using a free-hand drawn stick figure. By learning a mapping between query (parsed skeletons) and descriptors of video frames, it was possible to robustly identify relevant poses. The manifold mapping technique underpinning the process was guided by a minimal amount of user training. We showed that training provided for a single video could be generalised across several other unseen videos using a transductive domain adaptation technique. The resulting system demonstrated, for the first time, sketch based human pose retrieval, supporting the hypothesis:

**[H4]** Human Pose can be depicted within a VN to enable performance search

Adopting archival dance footage as our research material provided rich, diverse pose variations that enabled development and evaluation of our matching algorithms. However the nature of this footage forced a number of engineering decisions, such as the use of silhouettes as a basis for the video pose description. This introduces ambiguity to the left/right orientation of a performer within the descriptor.

### 6.1.3 ReEnact: Graph Representation for Search and Synthesis

In Chapter 5 we presented a graph representation inspired by Motion Graphs [123] that we applied for both video search and video synthesis. We constructed the graph using the measure of pose similarity developed within Chapter 4, namely geodesic distance over the learned manifold of plausible pose within video.

Regarding video synthesis, we presented the ReEnact system demonstrating it possible to synthesise novel choreography from existing archival footage of dance performance. Choreography was designed using VN, with the transitions between each sketched component specified using a semantic action label (and a timing constraint). By identifying disjoint frames that are visually similar were are able to synthesise video by walking around the manifold via a route optimised to pass through each of the specified key poses and actions, and respecting the timing constraints provided by the user. ReEnact therefore supports the research hypothesis:

**[H5]** Sketch based concatenative synthesis may be facilitated using VN

Early user evaluations of ReEnact highlighted the difficulty of synthesising video in line with user expectation, based solely upon the VN. Again, ambiguity in the representation or the need to iterate on initial ideas presented a need for the user to be placed 'in the loop' during choreographic design. We therefore extended our automated synthesis method, to an interactive setup in which the user is able to select from multiple close-to-optimal paths to be

afforded greater control over the video synthesis. Paths were clustered and visualised to the user using a further feature of our manifold learning technique, the ability to run inference in reverse so as to perform pose estimation from video. The resulting interactive system is another example of RF for VN based video search, additionally satisfying the hypothesis also addressed in Chapter 3:

**[H2]** SBVR accuracy can be improved using Relevance Feedback (RF)

Regarding video retrieval, we presented in Chapter 5 a system leveraging the motion graph representation for visual search. Using a VN specified as per the ReEnact system, we showed that contiguous sections of video matching the VN can be identified. Since the VN may contain an arbitrary number of sketched actions, sequenced video events are shown possible to be retrieved, supporting the hypothesis:

**[H3]** Video event retrieval may be facilitated effectively using VNs

Evaluating video sequence retrieval using VNs was challenging, since a contiguous stretch of video described exactly by a user supplied VN often does not exist, or exists just once within the dataset. Again the issue of prioritising accuracy in one query modality (e. g. correctness of pose) over another (e. g. correctness of action) comes to bear in identifying 'second best' clips that partially match the VN. For VN sequence search this was enabled through interactive weights the user could set with sliders.

## 6.2 Future Work

Several high-level directions for future research have been identified through the contributions of this thesis. We outline these from the perspectives of the two core themes running through this work; Visual Narratives for Video Search (subsec. 6.2.1 and Synthesis (subsec. 6.2.2). Any detailed technical improvements suggested to the algorithms within this thesis, have already been discussed within the conclusions of their respective chapters.

### 6.2.1 Visual Narratives for Retrieval

The Creative Industries are moving increasingly to all-digital pipelines for film and broadcast production, resulting in significant volumes of production and archival video assets. The re-use of these expensive assets could significantly reduce the cost of future production, and visual asset management tools are emerging for this purpose. Visual Narratives are already used extensively during production planning, in the form of hand-sketched production storyboards. Could such sketches be used to search for potentially relevant content even at these early planning stages? The algorithms presented in Chapter 3 showed promising

scalability, with sub-linear search complexity and query times of milliseconds over hundreds of video clips. Yet pre-production footage is frequently in the form of lengthy rushes (pre-edit) footage rather than segmented into shots of a few seconds in length. The proposed descriptor is not amenable to very long clips given its spatio-temporal nature, and it would be interesting to explore how descriptor extraction over multiple temporal scales could be use to create an efficient index of longer clips in this context.

Within the field of sketch based interaction we are observing a trend toward assisted drawing systems, such as ShadowDraw [130] (Sec. 2.3.3). These present not only an interesting aid to artists, but an extreme form of relevance feedback in which each new stroke results in a new iteration of results being presented to the user as the query is being formed. It would be interesting to explore how a similar system might operate for SBVR, perhaps using an non-photorealistic rendering algorithm to summarise the video into a visual narrative in real-time, adopting a visual style similar to that being used to specify the search query. The ability to interactively guide the user toward indexed content, rather than have users blindly sketch a scene only to find it is not present within the database, could help reduce search time and promote discovery of diverse relevant content. Most significantly, assisted drawing can help mitigate a fundamental problem for sketch based retrieval systems — that of poor user artistic skill. Perhaps more so than algorithmic or technical matters, the inability of many users to sketch a visually descriptive scene with any fidelity presents a barrier to scaling over extremely large datasets. A users' sketch, due to its ambiguity, may closely resemble many unwanted objects within the dataset. An improved sketch constructed via an assisted drawing system could encourage high quality query sketches.

Other potential application domains for our work include surveillance, e. g. to aid identification of people performing actions and/or wearing certain clothing, and even as an accessibility tool for computer users with poor literacy. Despite relatively high levels of poverty, the use of mobile devices has exploded in Third World countries. In a village where few people can read or write, could the ability to search a 'visual Wikipedia' using sketch be of value in accessing online content?

### 6.2.2 Visual Narratives for Synthesis

Sketch is an abstraction of photorealistic content in which salient objects are identified and depicted in their essential form. This thesis has explored manifold mapping techniques to learn the relationship between sketches and videos of human pose (Chapter 4), and leveraged that mapping for both search (find a video given a sketch) and summarisation (create a sketch given a video). It would be interesting to experiment with the flexibility of emerging technologies such as Gaussian Processes [170] to see if subspace approaches could be used

to more robustly estimate this bi-directional mapping. Learning such a mapping would be of great interest from a scientific point of view, since it might tell us something about how we perceive structure in a visual scene. It might also find commercial application, e. g. enabling animators to transform from a sketched storyboard to a video or animation, and enabling visual narratives as a production aid beyond their basic, current use as a coarse storyboarding tool.

We explored the synthesis of dance choreography through the splicing and blending of of archival footage to produce a new piece of choreography. An alternative approach might be to synthesise new video from a sketched 'visual tapestry', extending the work of *Sketch2Photo* [80] and *PhotoSketch* [67] to the temporal domain and so allowing for multiple objects with visual, semantic and motion attributes to inform the generation of a photo-realistic shot.

# Bibliography


[1] Huda Abdulaali Abdulbaqi, Ghazali Sulong, and Soukaena Oukaena Hassan Hashem. Sketch based Image Retrieval: A review of literature. *Journal of Theoretical and Applied Information Technology*, 63(1), 2014.

[2] A. Agarwal and B. Triggs. 3D human pose from silhouettes by relevance vector regression. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 882–888. IEEE, 2004.

[3] Rakesh Agrawal. Mining Association Rules between Sets of Items in Large Databases. In *ACM*, volume 22, pages 207–216, 1993.

[4] A Amir, M Berg, SF Chang, and W Hsu. IBM research TRECVID-2003 video retrieval system. In *proceedings of TRECVID*, pages 1–18. NIST, 2003.

[5] Rk Rie Kubota Ando and Tong Zhang. A high-performance semi-supervised learning method for text chunking. In *proceedings of Association for computational linguistics*, pages 1–9, 2005.

[6] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Discriminative appearance models for pictorial structures. *International Journal of Computer Vision (IJCV)*, 99(3):259–280, Springer, 2012.

[7] D Anguelov and B Taskarf. Discriminative learning of markov random fields for segmentation of 3d scan data. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 169–176. IEEE, 2005.

[8] R Arandjelović and A Zisserman. Three things everyone should know to improve object retrieval. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2911–2918. IEEE, 2012.

[9] A. Arnold, R. Nallapati, and W.W. Cohen. A Comparative Study of Methods for Transductive Transfer Learning. In *proceedings of International Conference on Data Mining Workshops (ICDMW)*, volume 1, pages 77–82. IEEE, 2007.

[10] James Arvo and Kevin Novins. Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. In *proceedings of Symposium on User Interface Software and Technology*, volume 2, pages 73–80, 2000.

[11] Muhammad Nabeel Asghar, Fiaz Hussain, and Rob Manton. Video Indexing: A Survey. *International Journal of Computer and Information Technology*, 3(1):148–169, 2014.

[12] Jonathan Ashley, Myron Flickner, James Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. The query by image content (QBIC) system. *proceedings of ACM International Conference on Management of Data (SIGMOD)*, 24(2):475, ACM Press, 1995.

[13] S. Ayer and H.S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 777–784. IEEE, 1995.

[14] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching Words and Pictures. *Journal of Machine Learning Research (JMLR)*, 3(6):1107–1135, 2003.

[15] Kobus Barnard, Quanfu Fan, Ranjini Swaminathan, Anthony Hoogs, Roderic Collins, Pascale Rondot, and John Kaufhold. Evaluation of localized semantics: Data, methodology, and experiments. *International Journal of Computer Vision (IJCV)*, 77(1-3):199–217, 2008.

[16] Arslan Basharat, Yun Zhai, and Mubarak Shah. Content based video matching using spatiotemporal volumes. *Computer Vision and Image Understanding (CVIU)*, 110(3):360–377, Elsevier Science Inc., 2008.

[17] P. R. Beaudet. Rotationally invariant image operators. In *proceedings of the International Joint Conference on Pattern Recognition*, pages 579–583, Kyoto, Japan, 1978.

[18] Martin Bergtholdt, Jörg Kappes, Stefan Schmidt, and Christoph Schnörr. A Study of Parts-Based Object Class Detection Using Complete Graphs. *International Journal of Computer Vision (IJCV)*, 87(1-2):93–117, Springer, 2009.

[19] D. Besiris, N. Laskaris, F. Fotopoulou, and G. Economou. Key frame extraction in video sequences: a vantage points approach. In *proceedings of the Workshop on Multimedia Signal Processing*, pages 434–437. IEEE, 2007.

[20] Alberto Del Bimbo and Pietro Pala. Visual image retrieval by elastic deformation of object sketches. *Pattern Analysis and Machine Intelligence (PAMI)*, 19(2):121–132, IEEE, 1997.

[21] Li Bin, Sun Zhengxing, Liang Shuang, Zhang Yaoye, and Yuan Bo. Relevance feedback for sketch retrieval based on linear programming classification. In *proceedings of Multimedia Information Processing*, volume 4261, pages 201–210. Springer, 2006.

[22] L R Binford and Bertman J. Bone Frequencies and attritional process. *Theory Building In Archaeology*, pages 77–156, 1977.

[23] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, number July, pages 120–128. ACM, 2006.

[24] .JS. Boreczky and L.D. Wilcox. A hidden Markov model framework for video segmentation using audio and image features. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3741–3744. IEEE, 1998.

[25] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image Classification using Random Forests and Ferns. *proceedings of the International Conference on Computer Vision (ICCV)*, pages 1–8, IEEE, 2007.

[26] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *proceedings of the International Conference on Image and Video Retrieval (CIVR)*, pages 401–408. ACM, 2007.

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

[28] M. Brand. Shadow puppetry. In *proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1237 – 1244. IEEE, 1999.

[29] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):844–851, IEEE, 2000.

[30] Matthew Brand. Voice Puppetry. In *proceedings of SIGGRAPH*, pages 21–28. ACM, 1999.

[31] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video Rewrite: Driving Visual Speech with Audio. In *proceedings of SIGGRAPH*, pages 353–360. ACM, 1997.

[32] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High Accuracy Optical Flow Estimation Based on a Theory for Warping. In *proceedings of the European Conference on Computer Vision (ECCV)*, pages 25–36. Springer, 2004.

[33] J. Calic, N. Campbell, S. Dasiopoulou, and Y. Kompatsiaris. A Survey on Multimodal Video Representation for Semantic Retrieval. In *proceedings of the International Conference on Computer as a Tool (EUROCON)*, volume 1, pages 135–138. IEEE, 2005.

[34] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF : Binary Robust Independent Elementary Features. In *proceedings of the European Conference on Computer Vision (ECCV)*, pages 778–792. Springer, 2010.

[35] G Camara-Chavez and F Precioso. Shot boundary detection by a hierarchical supervised approach. In *proceedings of the European Association for Signal Processing Conference (EURASIP)*, pages 197–200. IEEE, 2007.

[36] Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang. Edgel index for large-scale sketch-based image search. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 761–768. IEEE, 2011.

[37] João Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic segmentation with second-order pooling. In *proceedings of the European Conference on Computer Vision (ECCV)*, volume 7578, pages 430–443. Springer, 2012.

[38] Z Cernekova, I Pitas, and C Nikou. Information theory-based shot cut/fade detection and video summarization. *Transactions on Circuits and Systems for Video Technology*, 16(1):82–91, IEEE, 2006.

[39] A. Chalechale, G. Naghdy, and A. Mertins. Sketch-Based Image Matching Using Angular Partitioning. *Transactions on Systems, Man, and Cybernetics*, 35(1):28–41, IEEE, 2005.

[40] Shih-fu Chang, Horace J Meng, and Di Zhong. VideoQ : An Automated Content Based Video Search System Using Visual Cues. In *Proceedings of the International Conference on Multimedia (MM)*, pages 313–324. ACM, 1997.

[41] T Chang and C.-C.J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Transactions on Image Processing*, pages 429–441, IEEE, 1993.

[42] Siripinyo Chantamunee. University of Sheffield at TRECVID 2007 : Shot Boundary Detection and Rushes Summarisation 1 Shot Boundary Detection. In *proceedings of TRECVID*, pages 1–6. NIST, 2009.

[43] James Charles, Tomas Pfister, Derek Magee, David Hogg, and Andrew Zisserman. Domain Adaptation for Upper Body Pose Tracking in Signed TV Broadcasts. In *proceedings of the British Machine Vision Conference (BMVC)*, pages 47.1–47.11, 2013.

[44] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *proceedings of the British Machine Vision Conference (BMVC)*, number 1, pages 76.1–76.12. British Machine Vision Association, 2011.

[45] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *proceedings of the British Machine Vision Conference (BMVC)*, pages 1–11, 2014.

[46] GC Chavez and F Precioso. Shot boundary detection at trecvid 2006. In *proceedings of TRECVID*. NIST, 2006.

[47] Liang-Hua Chen, Yu-Chun Lai, and Hong-Yuan Mark Liao. Movie scene segmentation using background information. *Pattern Recognition*, 41(3):1056–1065, Elsevier Science Inc., 2008.

[48] Ming-yu Chen, Huan Li, and Alexander Hauptmann. Informedia @ TRECVID 2009 : Analyzing Video Motions. In *proceedings of TRECVID*. NIST, 2009.

[49] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2Photo. In *proceedings of SIGGRAPH*, volume 28, pages 124:1–124:10. ACM, 2009.

[50] W. Chen and H. Sundaram. VideoQ: a fully automated video retrieval system using motion sketches. In *proceedings of the Workshop on Applications of Computer Vision (WACV)*, pages 270–271. IEEE, 1998.

[51] Tat Seng Chua Tat Seng Chua, Kian-Lee Tan Kian-Lee Tan, and Beng Chin Ooi Beng Chin Ooi. Fast signature-based color-spatial image retrieval. In *Proceedings of International Conference on Multimedia Computing and Systems*, pages 362–369. IEEE, 1997.

[52] J. P. Collomosse, G. McNeill, and L. Watts. Free-hand sketch grouping for video retrieval. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–4. IEEE, 2008.

[53] John Collomosse, Graham McNeill, and Yu Qian. Storyboard sketches for content based video retrieval. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 245–252. IEEE, 2009.

[54] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(5):603–619, IEEE, 2002.

[55] Matthew Cooper, Ting Liu, and Eleanor Rieffel. Video segmentation via temporal pattern classification. *Transactions on Multimedia*, 9(3):610–618, IEEE, 2007.

[56] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Co-clustering based classification for out-of-domain documents. In *proceedings of the International Conference on Knowledge Discovery and Data mining (KDD)*, pages 210–219. ACM, 2007.

[57] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. *proceedings of the International Conference on Machine Learning (ICML)*, pages 200–207, ACM, 2008.

[58] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *proceedings of the European Conference on Computer Vision (ECCV)*, volume 3952 LNCS, pages 428–441. Springer, 2006.

[59] Hal Daumé, III, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. In *proceedings of the Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 53–59, Stroudsburg, PA, USA, 2010. ACM.

[60] Henrik Stewénius David Nistér. Scalable Recognition with a Vocabulary Tree. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168. ACM, 2006.

[61] T. Dharani and I. Laurence Aroquiaraj. A survey on content based image retrieval. In *proceedings of International Conference on Pattern Recognition (ICPR)*, pages 485–490. IEEE, 2013.

[62] Daniel Dixon, Manoj Prasad, and Tracy Hammond. iCanDraw: using sketch recognition and corrective feedback to assist a user in drawing human faces. In *proceedings of International Conference of Computer Human Interaction (SIGCHI)*, pages 897–906. ACM, 2010.

[63] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *proceedings of the International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72. IEEE, 2005.

[64] David H. Douglas and Thomas K. Peucker. *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature*, pages 15–28. B. V. Gutsell, 1973.

[65] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2D Articulated Human Pose Estimation and Retrieval in (Almost) Unconstrained Still Images. *International Journal of Computer Vision (IJCV)*, 99(2):190–214, Springer, 2012.

[66] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. A descriptor for large scale image retrieval based on sketched feature lines. In *proceedings of the Eurographics Symposium on Sketch-Based Interfaces and Modeling (SBIM)*, page 29, New York, New York, USA, 2009. ACM.

[67] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. PhotoSketch : A Sketch Based Image Query and Compositing System. In *proceedings of SIGGRAPH*, pages 1–4. ACM, 2009.

[68] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-Based Image Retrieval: Benchmark and Bag-of-Features Descriptors. *Transactions on Visualization and Computer Graphics*, pages 1–14, IEEE, 2010.

[69] Mark Everingham, Luc Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, Springer, 2009.

[70] Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable videorealistic speech animation. In *proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 57–64. IEEE, 2004.

[71] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, Springer, 2005.

[72] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient matching of pictorial structures. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 66–73. IEEE, 2000.

[73] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, number figure 1, pages 1–8. IEEE, 2008.

[74] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2009.

[75] M.A. Fischler and R.A. Elschlager. The Representation and Matching of Pictorial Structures. *Transactions on Computers*, C-22(1):67–92, IEEE, 1973.

[76] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, ACM, 1981.

[77] Matthew Flagg, Atsushi Nakazawa, Qiushuang Zhang, Sing Bing Kang, Young Kee Ryu, Irfan Essa, and James M. Rehg. Human video textures. *proceedings of Symposium on Interactive 3D graphics and games (I3D)*, 1(212):199–206, ACM, 2009.

[78] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim a. Jorge. Content-based retrieval of technical drawings. *International Journal of Computer Applications in Technology*, 23(2/3/4):86–100, Springer, 2005.

[79] Manuel J Fonseca, César Pimentel, and Joaquim A Jorge. CALI : An Online Scribble Recognizer for Calligraphic Interfaces. In *Proceedings of the Symposium on Sketch Understanding*, pages 1–8. AAAI, 2002.

[80] David Gavilan, Suguru Saito, and Masayuki Nakajima. Sketch-to-collage. In *proceedings of SIGGRAPH*, number sap 0333, page 35. ACM, 2007.

[81] Tiezheng Ge, Qifa Ke, and Jian Sun. Sparse-Coded Features for Image Retrieval. In *proceedings of the British Machine Vision Conference (BMVC)*, pages 132.1–132.11. BMVC, 2013.

[82] Andrew Gilbert, John Illingworth, and Richard Bowden. Action Recognition using Mined Hierarchical Compound Features. *Transactions on Pattern Analysis and Machine Learning (PAMI)*, 33(5):883–897, IEEE, 2009.

[83] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, and U C Berkeley. Rich feature hierarchies for accurate object detection and semantic segmentation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, 2012.

[84] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *proceedings of the International Conference on Computer Vision (ICCV)*, pages 415–422. IEEE, 2011.

[85] Georgia Gkioxari, Pablo Arbelaez, Lubomir Bourdev, and Jitendra Malik. Articulated pose estimation using discriminative armlet classifiers. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 3342–3349. IEEE, 2013.

[86] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Unsupervised Adaptation Across Domain Shifts By Generating Intermediate Data Representations. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(11):2288–2302, IEEE, 2013.

[87] Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear Body Pose Estimation from Depth Images. *Pattern Recognition*, pages 285–292, Springer, 2005.

[88] V.N. Gudivada and V.V. Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, IEEE, 1995.

[89] Robert M Haralick, K Shanmugam, and Its'Hak Dinstein. Textural Features for Image Classification. *Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, IEEE, 1973.

[90] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 545–552. MIT Press, 2007.

[91] Chris Harris and Mike Stephens. A Combined Corner and Edge Detection. In *proceedings of Alvey Vision Conference*, pages 147–152. BMVA, 1988.

[92] Miki Haseyama, Takahiro Ogawa, and Nobuyuki Yagi. A Review of Video Retrieval Based on Image and Video Semantic Understanding. *Transactions on Media Technology and Applications*, 1(1):2–9, ITE, 2013.

[93] Adam Herout, Vitezslav Beran, Michal Hradiš, Igor Potúček, Pavel Zemčík, and Petr Chmela. TRECVID 2007 by the Brno Group. In *Proceedings of TRECVID*. NIST, 2007.

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

[95] Huy Tho Ho and Raghuraman Gopalan. Model-driven domain adaptation on product manifolds for unconstrained face recognition. *International Journal of Computer Vision (IJCV)*, 109(1-2):110–125, Springer, 2014.

[96] Sch Hoi, Lls Wong, and Albert Lyu. Chinese university of hongkong at trecvid 2006: Shot boundary detection and video search. In *Proceedings of TRECVID*, pages 1–11. NIST, 2006.

[97] Junwei Jun-wei Hsieh, Shang-li Yu, and Yung-sheng Chen. Motion-based video retrieval by trajectory matching. *Transactions on Circuits and Systems for Video Technology*, 16(3):396–409, IEEE, 2006.

[98] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Transactions on Information Theory (IRE)*, 8(2):179–187, IEEE, 1962.

[99] Rui Hu, Mark Barnard, and John Collomosse. Gradient field descriptor for sketch based retrieval and localization. *proceedings of International Conference on Image Processing (ICIP)*, pages 1025–1028, IEEE, 2010.

[100] Rui Hu and John Collomosse. Motion-sketch based Video Retrieval using a Trellis Levenshtein Distance. In *proceedings of International Conference on Pattern Recognition (ICPR)*, pages 121–124. IEEE, 2010.

[101] Rui Hu, Stuart James, and John Collomosse. Annotated Free-hand Sketches for Video Retrieval using Object Semantics and Motion. In *Proceedings of International Multimedia Modeling Conference (MMM)*, pages 473–484. Springer, 2012.

[102] Rui Hu, Stuart James, Tinghuai Wang, and John Collomosse. Markov random fields for sketch based video retrieval. *Proceedings of the International conference on multimedia retrieval (ICMR)*, pages 279–286, ACM, 2013.

[103] Weiming Hu, Nianhua Xie, and Li Li. A Survey on Visual Content-Based Video Indexing and Retrieval. *Transactions on Systems, Man, and Cybernetics*, 41(6):797–819, IEEE, 2011.

[104] Peng Huang. *Surface Motion Graphs for 3D Video-based Animation of People*. PhD thesis, 2009.

[105] Cisco Systems Inc. Global Mobile Data Traffic Forecast Update 2013–2018. *Cisco Visual Networking Index*, `http://www.gsma.com/spectrum/wp-content/uploads/2013/03/Cisco_VNI-global-mobile-data-traffic-forecast-update.pdf`, 2013.

[106] H.H.S. Ip, A.K.Y. Cheng, and W.Y.F. Wong. Affine-invariant sketch-based retrieval of images. In *proceedings of Computer Graphics International*, pages 55–61. IEEE, 2001.

[107] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *proceedings of SIGGRAPH*, pages 277–286, New York, New York, USA, 1995. ACM.

[108] Monika Jain and S K Singh. A Survey On: Content Based Image Retrieval Systems Using Clustering Techniques For Large Data sets. *International Journal of Managing Information Technology*, 3(4):23–39, 2011.

[109] Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and C. V. Jawahar. Video retrieval by mimicking poses. In *proceedings of International Conference on Multimedia Retrieval (ICMR)*, pages 34:1–34:8. ACM, 2012.

[110] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming Embedding and Weak Geometry Consistency for Large Scale Image Search. In *proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–317. ACM, 2008.

[111] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311. IEEE, 2010.

[112] Shuiwang Ji, Ying-Xin Li, Zhi-Hua Zhou, Sudhir Kumar, and Jieping Ye. A bag-of-words approach for Drosophila gene expression pattern annotation. *BMC bioinformatics*, 10:1–16, BioMed Central, 2009.

[113] Hao Jiang and David R. Martin. Global pose estimation using non-tree models. *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, IEEE, 2008.

[114] Jing Jiang. *A literature survey on domain adaptation of statistical classifiers.* PhD thesis, 2008.

[115] Yongsen Jiang. An HMM based Approach for Video Action Recognition Using Motion Trajectories. In *proceedings of Intelligent Control and Information Processing (ICICIP)*, number 1, pages 359–364. IEEE, 2010.

[116] Sam Johnson and Mark Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *proceedings of the British Machine Vision Conference (BMVC)*, number i, pages 12.1–12.11, 2010.

[117] Karen Sporck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, Taylor Graham Publishing, 1972.

[118] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In Paolo Remagnino, GraemeA. Jones, Nikos Paragios, and CarloS. Regazzoni, editors, *Video-Based Surveillance Systems*, pages 135–144. Springer, 2002.

[119] Yousun Kang, Hiroshi Nagahashi, and Akihiro Sugimoto. Semantic Segmentation and Object Recognition Using Scene-Context Scale. In *proceedings of the Pacific-Rim Symposium on Image and Video Technology*, pages 39–45. IEEE, 2010.

[120] L. Kaufman and P.J. Rousseeuw. *An Introduction to Cluster Analysis.* John Wiley & Sons, Inc., 1990.

[121] Y Ke, R. Sukthankar, and M. Hebert. Efficient Visual Event Detection Using Volumetric Features. In *proceedings of the International Conference on Computer Vision (ICCV)*, pages 166–173. IEEE, 2005.

[122] KC Ko, YM Cheon, Gye-young Kim, and HI Choi. Video shot boundary detection algorithm. In *Computer Vision, Graphics and Image Processing*, pages 388–396. Springer, 2006.

[123] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *proceedings of SIGGRAPH*, volume 21, pages 1–10. ACM, 2002.

[124] a Krizhevsky, I Sutskever, and Ge Hinton. Imagenet classification with deep convolutional neural networks. *proceedings of Neural Information Processing Systems (NIPS)*, pages 1097–1105, Curran Associates, Inc., 2012.

[125] L Ladický and P Kohli. Object-class Segmentation using Higher order {CRFs}. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 1–8. ACM, 2008.

[126] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, pages 432–439. IEEE, 2003.

[127] Ivan Laptev and Tony Lindeberg. Local descriptors for spatio-temporal recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 3667 LNCS, pages 91–103. Springer, 2006.

[128] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE, 2006.

[129] Jehee Lee, Jinxiang Chai, Paul S. a. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. In *proceedings of SIGGRAPH*, volume 21, pages 491–500. ACM, 2002.

[130] Yong Jae Lee, C Lawrence Zitnick, Michael F Cohen, T Haldankar, and A Malu. ShadowDraw: Real-Time User Guidance for Freehand Drawing. In *proceedings of SIGGRAPH*. ACM, 2009.

[131] Qi Li. Literature Survey: Domain Adaptation Algorithms for Natural Language Processing. *Technical Report*, pages 1–54, City University of New York, 2012.

[132] Xi Li, Weiming Hu, Zhongfei Zhang, Xiaoqin Zhang, and Guan Luo. Trajectory-Based Video Retrieval Using Dirichlet Process Mixture Models. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–10, 2008.

[133] S Liang and Z Sun. Sketch retrieval and relevance feedback with biased SVM classification. *Pattern Recognition Letters*, 29(12):1733–1741, Elsevier Science Inc., 2008.

[134] Cailiang Liu, Dong Wang, Xiaobing Liu, Changhu Wang, and Bo Zhang. Robust Semantic Sketch Based Specific Image Retrieval. In *proceedings of the International Conference on Multimedia and Expo (ICME)*, pages 30–35. IEEE, 2010.

[135] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, Springer, 2004.

[136] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: cue integration in image segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 918–925. IEEE, 1999.

[137] Anna Margolis. A literature review of domain adaptation with unlabeled data. *Technical Report*, (March 2008):1–42, University of Washington, 2011.

[138] Smit Marvaniya, Sreyasee Bhattacharjee, Venkatesh Manickavasagam, and Anurag Mittal. Drawing an Automatic Sketch of Deformable Objects Using Only a Few Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 63–72. Springer, 2012.

[139] Stanislaw Matusiak, Mohamed Daoudi, Thierry Blu, and Olivier Avaro. *Sketch-Based Images Database Retrieval*. Springer, 1998.

[140] Krystian Mikolajczyk. Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86, Springer, 2004.

[141] Krystian Mikolajczyk and Cordelia Schmid. Performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–30, IEEE, 2005.

[142] Krystian Mikolajczyk and H Uemura. Action Recognition with Appearance-Motion Features and Fast Search Trees. *Computer Vision and Image Understanding (CVIU)*, pages 1–34, Elsevier Science Inc., 2010.

[143] George A. Miller. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41, ACM, November 1995.

[144] Yogita Mistry and D T Ingole. Survey on Content Based Image Retrieval Systems. *International Journal of Innovative Research in Computer and Communication Engineering*, pages 1827–1836, Foundation of Computer Science, 2013.

[145] Thomas B. Moeslund, Adrian Hilton, Volker Kruger, and Leonid Sigal. *Visual Analysis of Humans: Looking at People*. Springer, 2011.

[146] Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast Discriminative Visual Codebooks using Randomized Clustering Forests. In *proceedings of Neural Information Processing Systems (NIPS)*. MIT Press, 2007.

[147] Greg Mori and Jitendra Malik. Estimating human body configurations using shape context matching. In *proceedings of European Conference on Computer Vision (ECCV)*, pages 666–680. Springer, 2002.

[148] DP Mukherjee, SK Das, and Subhra Saha. Key Frame Estimation in Video Using Randomness Measure of Feature Point Pattern. *Transactions on Circuits and Systems for Video Technology*, 17(5):612–620, IEEE, 2007.

[149] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 692–699. IEEE, 2013.

[150] Gosuke Ohashi and Yoshifumi Shimodaira. Query-by-sketch image retrieval using relevance feedback. In *proceedings of Optomechatronic Machine Vision*, volume 6051, pages 60510Z–60510Z–9. SPIE, 2005.

[151] Dumebi Okwechime. *Computational Models of Socially Interactive Animation*. PhD thesis, 2011.

[152] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724. IEEE, 2014.

[153] Sinno Jialin Pan, James T. Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *proceedings of the Conference On Artificial Intelligence*. AAAI, 2008.

[154] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *Transactions on Neural Networks*, 22(2):199–210, IEEE, 2011.

[155] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning, 2010.

[156] B V Patel. Content Based Video Retrieval Systems. *International Journal of UbiComp*, 3(2):13–30, 2012.

[157] Vishal M Patel, Raghuraman Gopalan, and Ruonan Li. Visual Domain Adaptation : An Overview of Recent Advances. *Signal Processing Magazine*, 02138:1–34, IEEE, 2014.

[158] C Vishal Patil, R Kavin Kumar, and R Aarthi. Sketch Based Image Retrieval ? A Short Survey. *International Journal of Engineering Research & Technology*, 3(2):2475–2477, IJERT, 2014.

[159] Shilpa Pawar and Sonali Tidke. Survey on Sketch Based Image Retrieval System. *International Journal of Emerging Technology and Advanced Engineering*, 4(8):418–423, 2014.

[160] P Pérez, M Gangnet, and A Blake. Poisson image editing. In *proceedings of SIGGRAPH*, pages 313–318. ACM, 2003.

[161] Xavier Perez-Sala, Sergio Escalera, Cecilio Angulo, and Jordi Gonzàlez. A survey on model based approaches for 2D and 3D visual human pose recovery. *Sensors*, 14:4189–210, 2014.

[162] Florent Perronnin, Yan Liu, and S Jorge. Large-scale image retrieval with compressed Fisher vectors. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391. IEEE, 2010.

[163] Thomas Petersen. A Comparison of 2D-3D Pose Estimation Methods. Master's thesis, 2008.

[164] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, IEEE, 2007.

[165] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[166] Christian Plagemann and Daphne Koller. Real-time Identification and Localization of Body parts from depth images. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, pages 3108 – 3113. IEEE, 2010.

[167] John C Platt. Probabilities for SV Machines. In *Advances in large margin classifiers*, pages 61–74. MIT Press, 1999.

[168] Qiang Qiu, Vishal M Patel, Pavan Turaga, and Rama Chellappa. Domain adaptive dictionary learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 631–645. Springer, 2012.

[169] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught Learning : Transfer Learning from Unlabeled Data. *Proceedings of the International Conference on Machine Learning (ICML)*, pages 759–766, ACM, 2007.

[170] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 573–586. Springer, 2012.

[171] Deva Ramanan. Learning to parse images of articulated bodies. In *proceedings of Neural Information Processing Systems (NIPS)*. MIT Press, 2006.

[172] Deva Ramanan. Part-based models for finding people and estimating their pose. *Visual Analysis of Humans*, pages 199–223, Springer, 2011.

[173] Ananth Ranganathan. Semantic Scene Segmentation using Random Multinomial Logit. In *proceedings of the British Machine Vision Conference (BMVC)*, pages 1–12, 2009.

[174] Ren Reede and John Collomosse. Visual Sentences for Pose Retrieval Over Low-Resolution Cross-Media Dance Collections. *Transactions on Multimedia*, 14(6):1652–1661, IEEE, 2012.

[175] Neil Robertson and Ian Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding (CVIU)*, 104(2-3):232–248, Elsevier Science Inc., 2006.

[176] Remi Ronfard, Cordelia Schmid, and Bill Triggs. Learning to parse pictures of people. In *proceedings of the European Conference on Computer Vision (ECCV)*, volume 2353, pages 700–714. Springer, 2002.

[177] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *proceedings of SIGGRAPH*, pages 309–314, New York, NY, USA, 2004. ACM.

[178] E Rublee, V Rabaud, K Konolige, and G Bradski. ORB: An efficient alternative to SIFT or SURF. In *proceedings of the International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011.

[179] Mathieu Salzmann and Raquel Urtasun. Implicitly constrained gaussian process regression for monocular non-rigid pose estimation. In *proceedings of Neural Information Processing Systems (NIPS)*, pages 1–9. MIT Press, 2010.

[180] Arno Schödl and Irfan A Essa. Controlled animation of video sprites. In *proceedings of the Symposium on Computer animation (SCA)*, page 121. ACM, 2002.

[181] Arno Schödl, Richard Szeliski, David H. Salesin, Irfan Essa, and Arno Sch. Video textures. In *proceedings of SIGGRAPH*, pages 489–498, New York, New York, USA, 2000. ACM.

[182] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36. IEEE, 2004.

[183] E. Sciascio, M. Mongiello, E. Di Sciascio, G. Mingolla, and M. Mongiello. Content-Based Image Retrieval over the Web Using Query by Sketch and Relevance Feedback. In ArnoldW.M. Huijsmans, DionysiusP. and Smeulders, editor, *proceedings of Visual Information and Information Systems (VISUAL)*, pages 123–130. Springer, 1999.

[184] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *proceedings of the ACM International Conference on Multimedia (MM)*, number c, page 357, New York, New York, USA, 2007. ACM.

[185] Tevfik Metin Sezgin and Randall Davis. Sketch Interpretation Using Multiscale Models of Temporal Patterns. *Computer Graphics and Applications*, (February):28–37, IEEE, 2007.

[186] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 2003.

[187] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, Wiley Periodicals Inc., 2005.

[188] Eli Shechtman and Michal Irani. Matching Local Self-Similarities across Images and Videos. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 8. IEEE, 2007.

[189] Choon-bo Shim and Jae-woo Chang. *Efficient Similar Trajectory-Based Retrieval for Moving Objects in Video Databases*. Springer, 2003.

[190] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, Elsevier, 2000.

[191] J Shotton, J Winn, C Rother, and A Criminisi. TextonBoost : Joint Appearance , Shape and Context Modeling for Multi-Class Object Recognition and Segmentation.

In *proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2006.

[192] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, and Alex Kipman. Efficient human pose estimation from single depth images. *Decision Forests for Computer Vision and Medical Image Analysis*, pages 175–192, Springer, 2013.

[193] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[194] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *International Journal of Computer Vision (IJCV)*, 81(1):2–23, Springer, 2007.

[195] Matheen Siddiqui and Gerard Medioni. Human pose estimation from a single view point, real-time range sensor. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2010.

[196] Leonid Sigal and Michael J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2041–2048. IEEE, 2006.

[197] Nidhi Singhai and Shishir K. Shandilya. A Survey On: Content Based Image Retrieval Systems. *International Journal of Computer Applications*, 4(2):22–26, 2010.

[198] Josef Sivic and Andrew Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477. IEEE, 2003.

[199] Josef Sivic and Andrew Zisserman. Video Google: Efficient Visual Search of Videos. *Toward Category-Level Object Recognition*, 4170:127–144, Springer, 2006.

[200] Alan F. Smeaton, Paul Over, and Aiden R. Doherty. Video shot boundary detection: Seven years of TRECVid activity. *Computer Vision and Image Understanding (CVIU)*, 114(4):411–418, Elsevier Science Inc., 2010.

[201] J Smith and Shih-fu Chang. VisualSEEk: A fully automated content based image query system. In *proceedings of Multimedia*, pages 87–98. ACM, 1996.

[202] John R Smith and Shih-fu Chang. Transform features for texture classification and discrimination in large image databases. In *proceedings of International Conference on Image Processing (ICIP)*, volume 3, pages 407–411, Austin, TX, 1994. IEEE.

[203] John R Smith and Shih-fu Chang. Tools and Techniques for Color Image Retrieval. In *proceedings of the Storage and Retrieval for Image and Video Databases*, volume 2670, pages 2–7. SPIE, 1996.

[204] J.R. Smith and Shih-Fu Chang Shih-Fu Chang. Single color extraction and image query. *proceedings of the International Conference on Image Processing*, 3:1–4, 1995.

[205] R. Socher. ImageNet: A large-scale hierarchical image database. *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, IEEE, 2009.

[206] Pedro Sousa and Manuel J. Fonseca. Sketch-based retrieval of drawings using spatial proximity. *Journal of Visual Languages & Computing*, 21(2):69–80, Elsevier, 2010.

[207] Markus Stricker and Markus Orengo. Similarity of Color Images. In *SPIE*, pages 381–392, 1995.

[208] Chih-Wen Su, Hong-Yuan Mark Liao, Hsiao-Rong Tyan, Chia-Wen Lin, Duan-Yu Chen, and Kuo-Chin Fan. Motion Flow-Based Video Retrieval. *Transactions on Multimedia*, 9(6):1193–1201, IEEE, 2007.

[209] Min Sun and Silvio Savarese. Articulated Part-based Model for Joint Object Detection and Pose Estimation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 723–730. IEEE, 2011.

[210] Xinghai Sun, Changhu Wang, Chao Xu, and Lei Zhang. Indexing billions of images for sketch-based retrieval. *Proceedings of the International Conference On Multimedia (MM)*, pages 233–242, ACM, 2013.

[211] Hari Sundaram and SF Chang. Video scene segmentation using video and audio features. In *Proceedings of International Conference on Multimedia and Expo (ICME)*, volume 00, pages 1145–1148. IEEE, 2000.

[212] Kin-wai Sze, Kin-man Lam, and Guoping Qiu. A New Key Frame Representation. *Transactions on Circuits and Systems for Video Technology*, 15(9):1148–1155, IEEE, 2005.

[213] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural Features Corresponding to Visual Perception. *Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, IEEE, 1978.

[214] Yap-peng Tan and Hong Lu. Model-based clustering and analysis of video scenes. In *Proceedings of International Conference on Multimedia and Expo (ICME)*, volume 1, pages 617–620. IEEE, 2002.

[215] C.J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE, 2000.

[216] Sikora Thomas. The mpeg-7 visual standard for content description - an overview. *Transactions on Circuits and Systems for Video Technology*, pages 696–702, IEEE, 2001.

[217] Y Tian, Cl Zitnick, and Sg Narasimhan. Exploring the spatial hierarchy of mixture models for human pose estimation. In *proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2012.

[218] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1653–1660. IEEE, 2014.

[219] Endel Tulving. *Elements of Episodic Memory*. Oxford University Press, USA, 1985.

[220] Jack Valmadre and Simon Lucey. Deterministic 3D Human Pose Estimation Using Rigid Structure. In *proceedings of the European Conference on Computer Vision (ECCV)*, volume 6313, pages 467–480. Springer, 2010.

[221] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1582–1596, IEEE, 2010.

[222] Remco C. Veltkamp and Mirela Tanase. Content-based image retrieval systems: A survey. Technical report, 2000.

[223] R Vezanni, D Baltieri, and R Cucchiara. HMM based action recognition with projection histogram features. In *proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 6388, pages 290–297. Springer, 2010.

[224] A. Vezhnevets, J. M. Buhmann, and V. Ferrari. Active learning for semantic segmentation with expected change. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 3162–3169. IEEE, 2012.

[225] a. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 845–852. IEEE, 2012.

[226] Alexander Vezhnevets, Vittorio Ferrari, and Joachim M. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *proceedings of the International Conference on Computer Vision (ICCV)*, pages 643–650. IEEE, 2011.

[227] Ji Wan, Dayong Wang, Steven C H Hoi, and Pengcheng Wu. Deep Learning for Content-Based Image Retrieval : A Comprehensive Study. In *proceedings of the International Conference on Multimedia (MM)*, pages 157–166. ACM, 2014.

[228] Changhu Wang. MindFinder : Image Search by Interactive Sketching and Tagging. In *proceedings of the International Conference on World Wide Web*, pages 1309–1312. ACM, 2010.

[229] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, and Thomas Huang. Locality-constrained Linear Coding for Image Classification. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 27–30. IEEE, 2014.

[230] Zheng Wang, Yangqiu Song, and Changshui Zhang. Transferred dimensionality reduction. In *proceedings of the European Conference on Machine Learning (ECML)*, volume 5212, pages 550–565. Springer, 2008.

[231] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In *proceedings of the European Conference on Computer Vision (ECCV)*, pages 650–663. Springer, 2008.

[232] Willowgarage. OpenCV. http://opencv.org, 2015.

[233] X. Wu, Pong C. Yuen, C. Liu, and J. Huang. Shot Boundary Detection: An Information Saliency Approach. In *proceedings of the Congress on Image and Signal Processing*, pages 808–812. IEEE, 2008.

[234] G. Xue, G.R. Xue, W. Dai, W. Dai, Q. Yang, Q. Yang, Y. Yu, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *proceedings of SIGIR*, pages 627–634. ACM, 2008.

[235] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392. IEEE, 2011.

[236] Yang Yongsheng and Lin Ming. A Survey on Content based video retrieval. Technical report, 1999.

[237] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang. A Formal Study of Shot Boundary Detection. *Transactions on Circuits and Systems for Video Technology*, 17(2):168–186, IEEE, 2007.

[238] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *proceedings of International conference on Machine learning (ICML)*, page 114. ACM, 2004.

[239] Harris Zellig. Distributional structure. *Word*, 10(2-3):14, 1954.

[240] Kun Zhang, Krikamol Muandet, Zhikun Wang, and Others. Domain adaptation under target and conditional shift. In *proceedings of the International Conference on Machine Learning (ICML)*, volume 28, pages 819–827, 2013.

[241] Xiao Zhang, Zhiwei Li, Lei Zhang, Wei Ying Ma, and Heung Yeung Shum. Efficient indexing for large scale visual search. *proceedings of the International Conference on Computer Vision (ICCV)*, pages 1103–1110, IEEE, 2009.

[242] Zhi-Cheng Zhao and An-ni Cai. Shot boundary detection algorithm in compressed domain based on adaboost and fuzzy theory. In *Proceedings of the International Conference on Computing, Networking and Communications (ICNC)*, pages 617–626. Springer, 2006.

[243] Youding Zhu and Kikuo Fujimura. Constrained Optimization for Human Pose Estimation from Depth Sequences. In *proceedings of Asian Conference on Computer Vision (ACCV)*, volume 4843, pages 408–418. Springer, 2007.

[244] S. Zinger, C. Millet, B. Mathieu, G. Grefenstette, P. Hède, and P.-a. Moëllic. Extracting an Ontology of Portrayable Objects from WordNet. In *proceedings of MUSCLE/ImageCLEF workshop on Image and Video retrieval evaluation*, pages 17–23, 2005.

[245] S. Zuffi, O. Freifeld, and M. J. Black. From Pictorial Structures to deformable structures. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3553, IEEE, 2012.