UNIVERSITY OF SOUTHAMPTON

DOCTORAL THESIS

Mobile Image Parsing for Visual Clothing Search, Augmented Reality Mirror, and Person Identification

Author:
George A. CUSHEN

Supervisor:
Prof. Mark S. NIXON

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Vision, Learning and Control Research Group
School of Electronics and Computer Science

February 2016

Declaration of Authorship

I, George A. CUSHEN, declare that this thesis titled, 'Mobile Image Parsing for Visual Clothing Search, Augmented Reality Mirror, and Person Identification' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

“Far better it is to dare mighty things, to win glorious triumphs, even though checkered by failure, than to take rank with those poor spirits who neither enjoy much nor suffer much, because they live in the gray twilight that knows not victory nor defeat."

Teddy Roosevelt

University of Southampton

Abstract

Faculty of Physical and Applied Sciences
School of Electronics and Computer Science

Doctor of Philosophy

Mobile Image Parsing for Visual Clothing Search, Augmented Reality Mirror, and Person Identification

by George A. CUSHEN

With the emergence and growing popularity of online social networks, depth sensors (such as Kinect), smart phones/tablets, wearable devices, and augmented reality (such as Google Glass and Google Cardboard), the way in which people interact with digital media has been completely transformed. Globally, the apparel market is expected to grow at a compound annual growth rate of 5 between 2012 and 2025. Due to the huge impact for ecommerce applications, there is a growing interest in methods for clothing retrieval and outfit recommendation, especially efficient ones suitable for mobile apps. To this end, we propose a practical and efficient method for mobile visual clothing search and implement it as a smart phone app that enables the user to capture a photo of clothing of interest with their smart phone and retrieve similar clothing products that are available at nearby retailers. Furthermore, we propose an extended method where soft biometric clothing attributes are combined with anthropometrics computed from depth data for person identification and surveillance applications. This addresses the increased terrorist threat in recent years that has driven the need for non-intrusive person identification that can operate at a distance without a subject's knowledge or collaboration. We implement the method in a wearable mobile augmented reality application based on a smart phone with Google Cardboard in order to demonstrate how a security guard could have their vision augmented to automatically identify a suspect in their field of vision. Lastly, we consider that a significant proportion of photos shared online and via apps are selfies and of dressed

people in general. Hence, it is important both for consumers and for industry that systems are developed to understand the visual content in the vast datasets of networked content to aid management and perform smart analysis. To this end, this dissertation introduces an efficient technique to segment clothing in photos and recognize clothing attributes. We demonstrate with respect to the emerging augmented reality field by implementing an augmented reality mirror app for mobile tablet devices that can segment a user's clothing in real-time and enable them to realistically see themselves in the mirror wearing variations of the clothing with different colours or graphics rendered. Empirical results show promising segmentation, recognition, and augmented reality performance.

Acknowledgements

The decision to pursue a PhD was simultaneously the most exciting and most frightening of my career. It transpires to have been the best decision I could have made. I have thoroughly enjoyed my time at Southampton University and am very thankful for the opportunity I had to study here.

I would like to express my appreciation and thanks to my supervisor, Prof. Mark Nixon. As an undergraduate, his engaging and inspiring lectures fostered my passion in computer vision. In our first conversation when I began the PhD, he presented a list of suitable research topics and a clear path to overcoming obstacles to achieving the PhD. However, when I presented some alternative challenging ideas of my own, he had faith in me and gave me the freedom to develop them. In the years since, I am very appreciative of his continued enthusiasm, encouragement and guidance.

Thanks also go to my mini-thesis examiner, Dr. John Carter, for his insightful questions and invaluable feedback.

Additionally, thanks to all my friends and colleagues in the VLC research group who made this such an exciting and inspirational environment to study and conceive innovations.

Finally, I am especially grateful to my parents for laying the groundwork years ago by encouraging me to think creatively as a child and supporting me in many ways during the PhD.

Contents

Dedicated to my parents: Julie and Peter. . .

Chapter 1

Introduction

Ecommerce is an exponentially growing market. Global retail sales totaled $22.5 trillion in 2014, with $1.316 trillion of these sales occurring online. By 2018, ecommerce retail spending is projected to increase to nearly $2.5 trillion [1]. A significant proportion of this spending is on clothing items. Between 2012 and 2025, India and China are predicted to achieve clothing market compound annual growth rates (CAGR) of 12 and 10 respectively. Globally, this market is expected to grow at a CAGR of 5 to $2.1 trillion in 2025 [2]. Therefore, there is a high commercial importance in the global clothing ecommerce market, and in particular, the Indian and Chinese markets. However, finding precisely the clothing you want from online shops is still not a solved problem, and generally relies on text based searching and navigating around complex websites crammed with a massive choice of products. Due to the huge impact for ecommerce applications, there is a growing interest in methods for clothing retrieval and outfit recommendation, especially efficient ones suitable for mobile apps.

Many people now possess a smart phone with integrated camera and regularly use the camera with various apps. More than 1.8 billion photos are being uploaded to Flickr, SnapChat, Instagram, Facebook and WhatsApp every day [3]. The rapidly accelerating growth of online visual platforms such as these is transforming how people interact with photos and videos. It is clear from browsing social networks that the subject of a significant proportion of photos are “selfies” and of people in general. These large

social network datasets of people can be exploited by computer vision researchers. Particularly, computer vision algorithms to recognise clothing in photos of people may benefit content based image retrieval [4–6], person identification[7, 8], surveillance[9], computer graphics[10], intelligent fitting rooms[11], pose estimation[5], gender classification[12], and customer profile analysis.

With respect to the popularity of mobile devices, Steve Jobs made a key prediction at the D8 conference after the iPad's release in 2010 that tablet sales would eventually overtake personal computers. An interesting excerpt from his presentation which infers this prediction is as follows:

> *"We were an agrarian nation, all cars were trucks because that's what you needed on the farm. But as vehicles started to be used in the urban centers and America started to move into those urban and suburban centers, cars got more popular. PCs are going to be like trucks. They're still going to be around. They're still going to have a lot of value, but they're going to be used by one out of X people."*

According to Gartner, after five years of dramatic tablet growth and slowly declining sales of PCs (consisting of both desktops and laptops), 2015 will be the year that Jobs' vision is realized. A total of 320 million tablet sales and 316 million PC sales are predicted for 2015. Considering smart phones, Gartner is predicting 1.95 billion mobile phone sales for 2015, approximately 70% of which will be smart phones. In terms of mobile OS, Android is expected to lead with 53% of the market (1.37/2.59 billion units total) [20]. Thus, this shows the need for computer vision algorithms capable of running efficiently on (primarily Android) mobile devices, especially since it can be seen that many people use their mobile device as their camera.

Recognition algorithms which infer semantic attributes of objects in a scene have been rapidly gaining research attention. They allow for a more detailed description of objects over the traditional tasks of object matching and classification. Not only can objects be recognised by using the predicted attributes, but also unfamiliar objects can be described.

State of the art clothing labelling has been performed by image parsing. Image parsing attempts to find localised semantic labels, such as pixel-wise, rather than traditional recognition approaches which can only provide image-wise labels. Clothing parsing tackles the problem of explaining an image at a more general scale by unifying image segmentation, object detection, and recognition. This can allow for a wealth of possible applications including customer profile analysis, augmented reality and image retrieval.

Fashionistas on Chictopia, a popular fashion social network, often describe clothing primarily in terms of colours and patterns. However, 3D information is also an important cue when humans attempt to recognize and semantically describe dressed clothing because the dressed clothing may not have consistent color and texture but must inhabit an integrated region in space. Depth images have several advantages over 2D intensity images such as they are robust to changes in color and illumination, and they are simple representations of 3D structure. With the popular emergence of depth sensors, such as that found in the Microsoft Kinect, another important cue has become widely available for computer vision algorithms to exploit.

Imagine now to be able to take a photo on your smart phone of an item of clothing that you like, visually retrieve similar products from ecommerce stores, and be able to try on those items of clothing in a virtual or augmented reality without visiting a physical store. Virtual reality (VR) and augmented reality (AR) have the potential to be two of the most disruptive technologies for a decade [13]. Virtual reality puts the user in an entirely computer-generated world whereas augmented reality superimposes computer-generated images over the user's real world environment. As of 2013, there were 1.2 billion gamers in the world and none of them had the hardware or software needed to play VR or AR games [14, 15]. With the emergence in 2014 of Google Cardboard and Facebook's $2B acquisition of Oculus Rift, VR and AR have been thrust into the mainstream and the first devices have been made available to consumers. According to CCS Insight [13], hardware shipments of AR devices will increase 16-fold over the next 3 years, from 300,000 in 2015 to 4m by 2018. Hence, augmented reality is an important emerging technology to consider.

Due to recent terrorist attacks and the increasing threat, there is a need for non-intrusive person identification that can operate at a distance without a subject's knowledge or collaboration. Analysts forecast in 2014 that the global biometrics market will grow from $8.7 billion in 2013 to approximately $27.5 billion by 2019 with a five year compound annual growth rate of 19.8% between 2014 and 2019 [16]. One of the key drivers of this predicted expansion is the covert identification of individuals[16, 17]. At present, covert identification is primarily achieved through facial recognition software, however, there can be major problems with covert facial recognition such as insufficient resolution provided by the camera, shading, and occlusion or misalignment of a subject's face with the camera[18, 19]. Hence, this dissertation acknowledges the importance of the trend by investigating utilizing covertly captured traits of a subject's clothing and anthropometrics for this purpose.

1.1 Objective

Based on these findings and the associated gaps in literature, an investigation into semantic parsing (segmentation/classification) and retrieval of clothing given colour and/or depth images of individuals is the primary goal of this thesis. Specifically, the problems of *visual clothing search*, *augmented reality try-on of clothing*, and *person re-identification using wearable augmented reality devices* are considered.

The massive continued growth and popularity of social networks and their associated image datasets, augmented reality, and smart phone/tablet/wearable devices (with electronics consumers shifting towards buying and using mobile devices and away from PCs) has created a huge demand for *fast*, *efficient*, and *scalable* image analysis solutions. To this end, these three characteristics underpin the investigations and proposals. Hence, very computationally intensive models, such as those based on deep neural networks will not be presented.

1.2 Contributions

The main contributions of this thesis are as follows:

- A novel mobile client-server framework for automatic visual clothes searching featuring a new dominant colour descriptor for the efficient and compact representation of clothing is presented. The approach is evaluated on query images from a fashion social network dataset along with a clothing product dataset for results, showing promising retrieval results with a relatively fast response time. Thus this contribution resides in a mobile system for automated clothes search with proven capability. Figure 1.1 gives a brief preview of this contribution.
- An efficient method for semantic parsing of predominantly uniformly coloured clothing is proposed. The clothing is segmented using chromatic (colour) and achromatic (intensity) histograms, and high level clothing attributes such as clothing brand are classified by heuristic and random forest techniques. The method is quantitatively evaluated and demonstrated with application to augmented reality clothing try-on.
- An extension to our aforementioned augmented reality clothing parsing method. A dynamic multi-resolution approach is proposed for 3D shape reconstruction of highly deformable surfaces (such as cloth) in a photorealistic retexturing framework for augmented reality from monocular vision in a non-laboratory environment. We show that this approach can be used to realistically retexture graphics on clothing.
- A novel RGB-D based soft biometric clothing parsing framework for automatic person re-identification and retrieval is presented. The system extracts low and high level visual features as well as estimating core anthropometric features from captured RGB-D frames. K-nearest neighbours is then utilized to retrieve the closest matches and clothing parsing is performed to yield a semantically labelled person identification result. We apply the approach to the emerging field of wearable mobile augmented reality, enabling a security

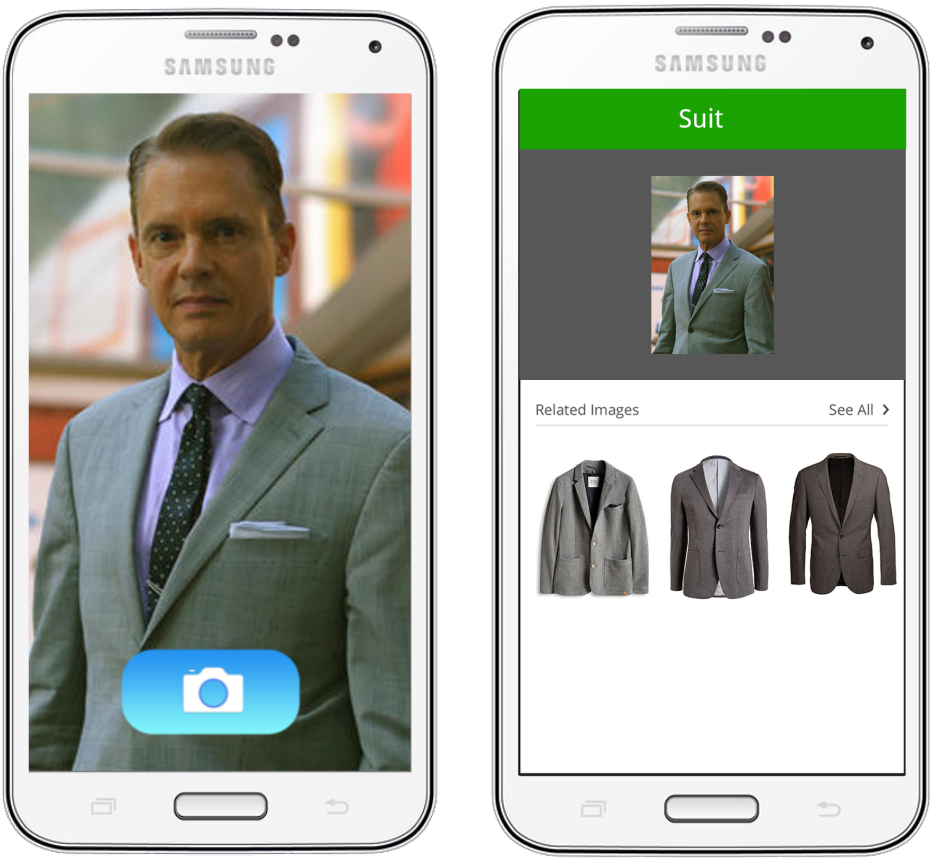


FIGURE 1.1: Our mobile visual clothing search application.

guard's vision to be augmented with the identify of a suspect in their field of view.

Currently, the following papers have arisen from this research:

- G. A. Cushen and M. S. Nixon. Markerless Real-Time Garment Retexturing From Monocular 3D Reconstruction. In *IEEE ICSIPA*, pages 88–93, Malaysia, November 2011
- G. A. Cushen and M. S. Nixon. Real-Time Semantic Clothing Segmentation. In *ISVC*, pages 272–281. Springer, 2012
- G. A. Cushen and M. S. Nixon. Mobile Visual Clothing Search. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*. IEEE, 2013
- G. A. Cushen. A Person Re-Identification System For Mobile Devices. In *Signal Image Technology & Internet Systems (SITIS)*. IEEE, 2015

These publications form the foundation of the dissertation and have been extended and improved upon. Most notably, they have been cited and discussed by Kalantidis et al. [24], Jammalamadaka et al. [25], Jia-Lin Chen et al. [26], Yamaguchi et al. [27, 28] and Traumann et al. [29].

1.3 Organisation

The thesis is structured as follows:

CHAPTER 2: Background gives a survey of the related state of the art work in the relevent research fields within computer vision, machine learning and computer graphics. Secondly, some of the core tools and algorithms that are built on throughout the dissertation are introduced.

CHAPTER 3: Product Retrieval describes a visual clothing search system for mobile devices such as smart phones which uses an efficient feature extraction pipeline whilst achieving reasonable accuracy on the results retrieved from real clothing e-commerce stores.

CHAPTER 4: Augmented Reality Mirror proposes efficient image segmentation and semantic parsing for augmented reality try-on of upper body clothing and fashion analysis. The approach is demonstrated in the form of an app for tablets and smart phones.

CHAPTER 5: Augmented Reality Re-texturing extends the work in the previous chapter with a monocular surface reconstruction approach for retexturing graphics on the deformable clothing surface. For example, this can enable different graphical T-shirt designs to be realistically augmented on the user.

CHAPTER 6: Person Identification proposes a solution to the problem of person identification on mobile devices by combining soft biometric cues from clothing and anthropometrics.

CHAPTER 7: Conclusions draws concluding remarks and discusses avenues for potential future work in the field.

Chapter 2

Background

Clothing parsing, mobile augmented reality, mobile visual search, and mobile person re-identification are complicated problems that build upon an in depth and wide breadth of research in computer vision, machine learning and computer graphics. In this chapter, the related work in the relevent research fields is reviewed and analysed.

Later in the chapter we introduce some of the core tools and algorithms we build on throughout our work. They include local feature descriptors, encoders, and classification algorithms.

2.1 Related Work

2.1.1 Clothing Segmentation and Parsing

Clothing can be considered to be one of the core cues of human appearance and segmentation is one of the most critical tasks in image processing and computer vision. Clothing parsing is a domain specific application of the more general image parsing. The goal of image parsing in computer vision is to provide a semantic label to each pixel in a given image. In this chapter the problem of automatic and efficient labelling of clothing, textures (such as logos) on the clothing, and the background is addressed. Clothing and texture labels are considered as they can be important cues for augmented reality and person identification applications.

There has been a rapidly accelerating interest in clothing segmentation/-parsing problems due to the important benefits of image retrieval for e-commerce [4, 24, 30] (refer to chapter 3), virtual and augmented reality [21] (refer to chapter 4), recognition for re-identification [8] (refer to chapter 6), human detection [31], pose estimation [32], and fashion analysis [33].

For background reading on segmentation techniques, the reader is referred to the literature survey of Zhang et al. [34]. From this survey, it is clear that a popular representation of images uses graph theory and in particular GrabCuts, however, we find this can be too computationally intensive for responsive mobile phone applications.

A review of the literature yields a number of approaches to solving general image segmentation and parsing problems [35–41]. One of the most popular approaches for the more focussed area of clothing segmentation involves a Markov Random Field (MRF) framework based on graph cuts [8, 42, 43]. Although these approaches have robustness to a diverse range of clothing, they can suffer in accuracy, producing very crude segmentation. This is especially true in cases of occlusions and difficult poses. The MRF has since been reformulated to deal with groups of people [44]. More recently, Wang and Ai [45] introduced a clothing shape model which is learned using Random Forests and self-similarity features, with a blocking model to address person-wise occlusions.

State of the art work focusses on offline clothes parsing with deep semantic classification [5, 12, 24, 25, 27, 46–49], Kinect based segmentation [29] (also refer to the method presented in chapter 6), and semantic 3D clothing segmentation [50]. In chapter 4, the 2D approaches in related work are of primary interest as an efficient application for mobile devices is considered and publicly available mobile devices are currently only capable of capturing 2D images, so it's the most practical.

Chen et al. [12] model clothing by a conditional random field (CRF) with classification predictions from individual attribute classifiers. They propose an application to predict the dressing style of a person by analysing a group of photos. Yamaguchi et al. [5] begin with superpixels and articulated pose estimation in order to predict and detect the clothing classes

present in a real-world image. Their paper can be considered one of the most notable early works on clothing parsing. Kalantidis et al. [24] present a method similar to that of Yamaguchi et al. [5] but for an application of cross-scenario retrieval. They start from an articulated pose estimation, segmenting the person and clustering image regions in order to detect the clothing classes present in a query image. Manfredi et al. [49] design a clothing retrieval system where clothing is segmented by Gaussian Mixture Models (GMMs) and graph cuts. Their approach is offline and is evaluated against a very simple dataset captured in a controlled lab environment. Yamaguchi et al. [27] employ a retrieval based approach to tackle the problem of clothing parsing. Given a query image, similar styles from a large database of tagged fashion images are found and used as examples to recognise clothing items in the query. The approach combines global, local, and transferred parse-masks from retrieved examples. They show significant improvements over previous state-of-the-art for both localization (clothing parsing given weak supervision in the form of tags) and detection (general clothing parsing). They also find that the pose estimation problem can benefit from the results of clothing parsing.

The most notable state of the art two dimensional work on clothing parsing is by Yamaguchi et al. [5, 27, 46–48]. They achieve good performance at parsing a diverse range of clothing classes on real world photos from an online fashion social network, although their methods, particularly [5], are offline. The methods are very computationally intensive and require a fully body pose (rather than upper body, such as in many of the "selfies" photos that are shared on more general social networks).

Although the field has recently been gaining much attention, a real-time clothing parsing system remains challenging. This is primarily due to the wide diversity of clothing designs, uncontrolled scene lighting, dynamic backgrounds, variation in human pose, and self and third-party occlusions. Secondly, difficult sub-problems such as face detection are usually involved to initialize the parsing procedure. The fastest approaches in literature are the approaches presented in this chapter: Cushen and Nixon [22] (2.0ms per detected person in the query image), Cushen and Nixon [21] (3.7ms for the overall application including segmentation), as well as

the region growing approach presented in [9] (16.5ms per detected person). Although [9] extracts the person including skin pixels, the approach is fast, with the authors reporting 16.5ms per detected person for segmenting clothing and 10fps overall (including face detection and a classification application). Their private dataset was captured in a controlled lab setup, featuring a predominantly white background. In [51], an RGB-colorspace approach is briefly described for segmenting T-shirt cloth and texture in a manner which to some degree removes these drawbacks. However, their method requires major *a prioris* such as cloth color and a simple rectangular texture, and they only show subjective visual results for a specific T-shirt.

The papers [21] and [22] resulted from the two approaches presented in chapter 4. Even though research in the field of clothing segmentation and parsing has rapidly accelerated in the last few years, with papers such as [24, 25, 29] citing the above, our approaches remain unique as they are specifically targeted towards practical applications for mobile devices and real-time broadcasting where very high efficiency is required. Furthermore, the related work exhibits a number of constraints and can be less suitable and practical for the considered applications.

Face detection is employed in many existing approaches to initialize points on the cloth. It should be noted that due to frontal face detection, the segmentation approaches are limited to frontal poses. The initial cloth points are located by applying a (scaled) distance from the bottom of the detected face. In the case of clothing with deep neck lines, such as vests and many female tops, these methods can segment the skin rather than the cloth. In contrast, we design a more complex initialization scheme which attempts to avoid this.

The majority of related work focusses on segmenting a single image of a single person offline. Contrary to these methods, it is attempted to simultaneously process multiple persons and maintain reasonable accuracy whilst increasing computational efficiency to enable real-time image/video processing. The efficiency of the proposed methods makes them practical for use on mobile devices. Furthermore, the author is not aware of any existing fast approaches in literature to specifically yield a semantically

labelled segmentation which includes contours for any textured regions, such as logos, on the cloth.

2.1.2 Clothing Retrieval

Based on the clothing and mobile device statistics and trends mentioned in chapter 1, an efficient mobile application to automatically recognize clothing in photos of people and retrieve similar clothing items that are available for sale from retailers could transform the way we shop whilst giving retailers a great potential for commercial gain. Tightly connected to this, is the potential for an efficient clothing retrieval system to be employed for the purpose of highly targeted mobile advertising which learns what clothing a person may wish to purchase given their social networking photos.

The problem of efficient and practical mobile clothing search appears relatively unexplored in literature [4], although there is a growing number of papers related to the more general case of clothing recognition for retrieval and recommendation applications [24, 26, 27, 30, 33, 49, 52, 53]. This growth has been triggered by the evolution of automatic clothing parsing and recognition methods which can enable natural semantic image searching (refer to subsection 2.1.1).

One early scenario of clothing recognition for retrieval is presented in [6, 11]. Si Liu et al. [30] address the problem of cross-scenario clothing retrieval where the query image is a real world image and results are retrieved from an online shopping dataset with simple poses and clean backgrounds. The authors of the paper consider a human detector to locate human parts and a calculation on the parts is used to obtain one-to-many similarities between the query photo and online shopping photos. A mapping is created between real world and e-commerce images with a sparsely coded transfer matrix so that the difference between these two distributions does not adversely affect the quality of retrieval. Note that their approach has offline elements and is designed for a PC - it does not consider a mobile framework for retrieval, unlike the first photo in the paper suggests. Kalantidis et al. [24] take a similar cross-scenario retrieval

approach, employing clothing parsing to represent each clothing item explicitly and utilizing Yamagutchi's Fashionista dataset to learn clothing classes.

State of the art work focusses primarily on clothes parsing based on retrieval results [27], detection based deep semantic classification [53], random forests [49], and interactive clothing retrieval [26]. The method of Yamaguchi et al. [27] is discussed in subsection 2.1.1. Chen et al. [53] do not segment the images, such as in clothing parsing methods like Yamagutchi et al. and instead propose a double-path deep domain adaptation network to model the data from the two domains of constrained (as in typical e-commerce photos) and unconstrained (as in real world photos) images jointly. Alignment cost layers are placed to ensure the consistency of the two domain features and the feasibility to predict unseen attribute categories in one of the domains. Both of these related works achieve good performance, but they are computationally intensive and unsuitable for fast searching on a mobile infrastructure. In the third state of the art work mentioned, Manfredi et al. [49] model the background with Gaussian Mixed Models (GMMs) in order to segment the image and classifies clothing attributes using a random forest. Their method is perhaps faster than the previous approaches, however they only consider a very simple dataset which appears to have been captured in a controlled lab environment. Jia-Lin Chen et al. [26] design an interactive clothing retrieval system on Yamaguchi's Fashionista dataset. However, the focus in chapter 4 is on automatic approaches.

Current mobile image retrieval systems include Google Goggles[1], Kooaba, and LookTel. However, these systems are developed for image retrieval on general objects in a scene. When these systems are applied to clothes search, they can provide visually and categorically less relevant results than our method for retrieving products based on a dressed person and can have significantly longer response times than our method.

[1] google.com/mobile/goggles, kooaba.com, looktel.com

2.1.3 Person Identification

The increased terrorist threat in recent years has driven the need for non-intrusive subject identification that can operate at a distance. Person identification and retrieval are a critical tasks in surveillance for recognising a person across spatially disjoint sensors. The emerging field of soft biometrics consists of physical, behavioural or adhered human characteristics which can be utilized for this task. Besides the rapidly growing number of published papers on person identification, the importance of this field is recognised in the recent survey on re-identification methods by Vezzani et al. [54], published in the book on "Person Re-Identification" edited by Gong et al. [55]. The reader is referred to this publication for a comprehensive background in person identification.

A number of soft biometric approaches have been presented recently [56–59]. Some general re-identification applications have also been proposed [60–62], as well as some more specifically related to clothing[7, 8, 63]. Yang and Yu [9] develop an application for a surveillance scenario albeit with some major constraints. Baltieri et al. [64] introduced 3D anthropometric information to the re-identification problem where a coarse and rigid 3D body model was fitted to different pedestrians. Barbosa et al. [65] considering the use of depth sensors for re-identification and create an RGB-D dataset for this purpose. Jaha and Nixon [63, 66] show how clothing traits can be exploited for identification purposes, exploring the validity and usability of a set of proposed semantic attributes. Baltieri et al. [67] exploit non-articulated 3D body models to spatially map appearance descriptors into the vertices of a regularly sampled 3D body surface.

Real world image sequences from CCTV cameras often feature a person of interest in the foreground and a cluttered background. Some recent approaches for general clothing retrieval [24, 30] formulate the problem as one of cross-scenario. For cross-scenario retrieval, the query image for searching is a real world image while the search results are often returned from a dataset captured in a controlled lab environment with simple backgrounds. The approach proposed in this chapter returns real-world images of subjects with similar clothing and anthropometrics to those in a real-world query image of a subject. Thus, the approach can more challenging

and attempts to be more practical in terms of application than many more general related works on clothing retrieval.

Related work tends to focus on low-level visual appearance based cues and pays relatively little attention to predicting clothing attributes or incorporating RGB-D sensors, although these topics are rapidly gaining attention. Here, a soft biometrics method for person identification which predicts high level clothing attributes (e.g. shirt) and exploits anthropometrics (e.g. height) from RGB-D sensors is presented. By combining these cues, our approach attempts to provide robustness to noise and minor variations in clothing, occlusions, and illumination conditions, achieving promising intermediate results for identification without requiring facial features (i.e. the image does not need to be very high resolution). Furthermore, by utilizing anthropometrics in the approach it can provide robustness to a potentially large number of subjects wearing similar clothes, where traditional appearance based cues for identification may be of little use.

Figure 2.1 shows a real-life need for the proposed soft biometrics approach based on clothing attributes during the 2011 London Riots. The large image highlights a marked suspect with a covered face[2] and in the bottom right corner is a separate image of a suspect in a different location who appears to wear the same distinct clothes[3]. This example also demonstrates that in some cases, semantic clothing attributes can be the only soft biometric cue available for exploitation.

Table 2.1 lists a comparison of some common soft biometric traits. Clothing attributes and anthropometrics such as height are employed in this thesis as they are both reasonably distinct cues that are commonly used by humans to describe each other, can be captured non-intrusively at a distance and it is hypothesized that the increased permanence of the height cue can help improve the performance over using clothing only cues. The proposed system can also be used to supplement other biometrics such as gait.

[2] http://www.adelaidenow.com.au/.../story-e6frea8l-1226111443592
[3] http://www.bbc.co.uk/news/uk-england-london-16171972

FIGURE 2.1: A usecase for soft biometrics based on clothing and anthropometrics.

Soft biometric cue	Permanence	Distinctiveness	Variable
Weight	Medium	Medium	Continuous
Height	High	Medium	Continuous
Semantic Clothing Attributes	Low	Medium/High	Discrete
Gender	High	Low	Binary
Gait	Medium	High	Continuous

TABLE 2.1: Comparison of soft biometric traits

2.1.4 Augmented Reality

Augmented reality systems conventionally operate in real-time, allowing the user to interact with computer-generated objects in a real scene. For a comprehensive survey of traditional augmented reality literature, the reader is referred to [68, 69]. The previously described work by Bradley et al. [70] and Hilsmann and Eisert [51] can be considered the closest to our augmented reality framework.

2.1.5 Clothing Surface Reconstruction

Our methods are primarily related to work from the following four areas of research: (1) garment segmentation, (2) geometric recovery of flexible surfaces, (3) cloth modeling, and (4) real-time augmented reality. This section describes the most relevant work from each of these categories.

The paper [21] resulted from the research presented in this chapter. Even though our work has since been extended by [29], the approach in this chapter remains unique as it does not require manual interaction or depth images.

2D Surface Recovery An area of literature which is generally considered to be solved is the *2D* recovery of a non-rigid surface from *monocular vision* [71–75]. Recently, Bradley et al. [70] and Hilsmann and Eisert [51] have extended the work on this topic for the purpose of AR applications. Bradley et al. use markers printed onto a white T-shirt for 2D cloth recovery and optical flow is used to track the marker positions over a video sequence for retexturing the markered region with a new image. Hilsmann and Eisert extend this optical flow approach with a deformable model for self-occlusion handling and consider a specific T-shirt with a rectangular texture rather than markers.

3D Surface Recovery Multiple view geometry is a very popular field in Computer Vision for solving problems such as human pose estimation and 3D shape reconstruction. A survey of multiple view reconstruction literature is presented in [76]. The 3D shape of non-rigid surfaces can be reconstructed in a similar way to human vision: by constraining the depth based on multiple view geometry and establishing point correspondences across the views. Given sufficient views, this process is over-constrained for a moderately deformable surface. However, for the case of a highly non-rigid surface, such as cloth, the cloth can exhibit many self-occlusions which make simple point correspondence techniques fail. White et al. [77] addressed this with a marker-based approach where markers are printed on the garments for successful cloth surface reconstruction. Recently, Bradley et al. [78] improved upon this by presenting a markerless approach.

Recovering the *3D* geometric layout of deformable self-occluding surfaces from *monocular vision* remains an open and challenging problem in computer vision due to the fact that it is severely under-constrained. The problem can be overcome to some extent by introducing deformation models which are either physically-based [79–84] or learned from training data

[85–88]. The model parameters in all these methods are initialized and refined by the minimization of an image based objective function. Good initialization is important but very difficult to achieve because this function generally has many local minima. Recent work [89–92] addresses this issue by proposing constraints of a reference image in which the shape is known, and pixel correspondences between the input and reference images which are also known. These methods assume that the surface is inextensible and hence the geodesic distances between neighboring surface points must remain constant.

Alternatively, Shape from Shading (SfS) techniques can be used to reconstruct 3D shape from shading information in a single image with a fully calibrated camera. This is especially suitable for relatively untextured objects, with few reliable features. Traditionally, shape from shading techniques were designed within the context of Lambertian surfaces of an unknown albedo, with a distant single point light source [93]. Since then, many variations [94] have been proposed, but shape from shading techniques continue to suffer from a number of limitations because they depend on very restrictive assumptions. Also, shape from shading is known to suffer from the Bas-Relief Ambiguity [95]. This refers to the fact that there is an implicit ambiguity in determining 3D shape from an unknown object with Lambertian reflectance which is viewed orthographically. These drawbacks reduce the applicability of shape from shading methods significantly. Recently, more accurate models based on non-Lambertian surfaces have been used to replace the Lambertian assumptions [96].

Cloth Modeling Cloth modeling is a popular field in computer graphics which attempts to mimic the dynamic three-dimensional characteristics of real deforming cloth. The wide range of literature [97, 98] can be grouped into the following categories:

- Geometrical techniques tend to focus on methods such as curve fitting, sub-division, relaxation, interpolation, and user interaction.
- Physical techniques tend to focus on methods such as Newtonian dynamics, elasticity theory, and deformable models. Physical techniques can be categorised into the following approaches:

Energy-based approaches calculate the entire cloth energy from a set of equations and then carry out energy minimisation. Generally best suited for computing static cloth simulations.

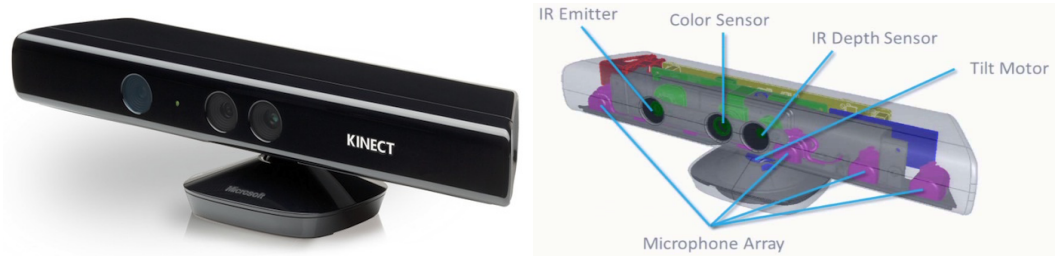
Force-based approaches represent the forces at nodes as differential equations and then perform numerical integration at each time step, to obtain the node coordinates. Generally best suited for computing dynamic cloth simulations.

- Hybrid techniques exploit the advantages of geometrical and physical models in an attempt to achieve superior results. Generally, a geometrical technique is employed to compute a rough representation of the cloth, and then a physical technique is employed to refine the cloth structure.

Early computer graphics cloth simulation techniques were developed some time ago and examples include Terzopoulos et al. [99], Terzopoulos and Fleischer [100], Weil [101], Thingvold and Cohen [102], Carignan et al. [103], Okabe et al. [104], Breen et al. [105].

In recent years, there has been a strong focus on high resolution, offline cloth simulation [106], with one of the popular applications being photo-realistic cloth simulation in films [107–109].

Mass spring systems are frequently used in the simulation of deformable objects due to their high computational efficiency, conceptual simplicity, and reasonable results [110, 111]. This allows for real-time processing. It can be noted that results by Bridson et al. [108], Etzmuss et al. [112], Irving et al. [113], Selle et al. [114] show that the more complex finite elements models behave similarly to mass-spring models.



(A) Microsoft Kinect (B) Components

FIGURE 2.2: Microsoft Kinect sensor

2.2 Devices

2.2.1 Microsoft Kinect

The Microsoft Kinect sensor (Figure 2.2) consists of an infrared (IR) laser emitter, an infrared camera and an RGB camera. Unlike traditional cameras, the device allows computers to accurately view the captured scene in three dimensions. We choose to use the Microsoft Kinect for three dimensional data capture since it is the most popular consumer depth sensor (over 29 million units of the Kinect models sold [115]), is relatively affordable, achieves good performance, and is widely used for the purpose of academic research.

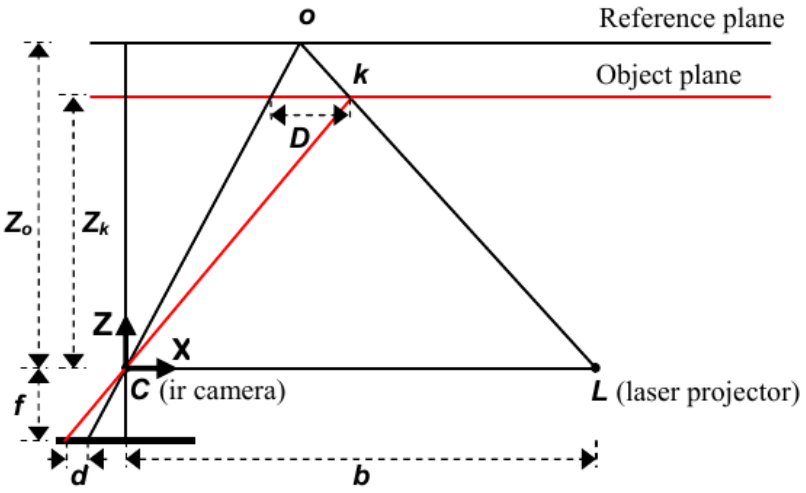


FIGURE 2.3: Diagram of depth-disparity relationship

The IR laser emits a beam which is split into multiple beams by a diffraction grating to create a constant pattern of dots projected onto the scene.

Objects in the scene cause the pattern to appear deformed. An IR camera captures the resulting pattern and it is correlated against a reference pattern to obtain a disparity map. For each pixel in the disparity map, the distance between the object in the scene and the sensor can be calculated. Figure 2.3 depicts the relationship between the distance of a point k and the measured disparity d. Note that the coordinate system has the Z axis perpendicular to the image plane, X axis orthogonal to Z along the baseline b, and Y axis orthogonal to Z and X. Assuming that there is an object on the reference plane at distance Z_0 from the sensor and a corresponding disparity d in image space, then by the similarity of triangles:

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \tag{2.1}$$

and

$$\frac{d}{f} = \frac{D}{Z_k} \tag{2.2}$$

where Z_k is the depth distance of the point k in object space that we want to measure, D is the displacement of point k, b and f are the base length and focal length of the IR camera, respectively. Substituting D from Equation 2.2 into Equation 2.1 we can obtain the depth Z_k:

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f_b} \cdot d} \tag{2.3}$$

The limitations of the Kinect are as follows [116]:

- field of vision is $57.8°$
- range is from 0.6m up to 5m
- frames are captured at $640 \times 480 \times 30$fps (with 3 bytes of color and 2 bytes of depth data)
- density of points decreases with increasing distance from the sensor; depth resolution is very low (7cm) at the maximum distance (5m)

- random error of depth measurements increases quadratically with increasing distance from the sensor, reaching 4cm at the maximum range of 5m
- indoor use only, avoiding dusty environments, IR light, and variable lighting

Ideally, data should be acquired within 1 to 3 metres of the sensor. Otherwise, at larger distances, the data quality becomes significantly degraded by noise and the low resolution of the depth measurements.

2.2.2 Google Cardboard

Google Cardboard is a virtual reality platform developed by Google for use with a smart phone in a head mount made from folded cardboard, as seen in Figure 2.4. It was conceptualized by David Coz and Damien Henry and introduced in 2014 at the *Google I/O* developers conference. It makes immersive virtual (or augmented) reality accessible to everyone in possession of a smart phone in a simple, fun, and affordable way. The head mount can be purchased ready-made or can be easily constructed from cardboard, lenses, magnets, velcro and a rubber band (Figure 2.5) by following Google's instructions.

We choose to use Cardboard as our wearable virtual/augmented reality device since it is currently the only popular product of its kind available that we can develop for. Related products include Google Glass, Oculus Rift, and HTC Vive (Steam VR). However, Google Glass was in an experimental stage which ended in it becoming discontinued in January 2015, whilst Google work to improve the technology. Whereas, Oculus Rift and HTC Vive (Steam VR) are yet to be released and designed for gamers in a home environment.

The Cardboard SDK can be utilized to build apps for Android and Apple iOS smart phones that display 3D scenes with binocular rendering, track and react to head movements, and interact with apps through trigger input.

FIGURE 2.4: The Google Cardboard virtual reality (VR) platform.

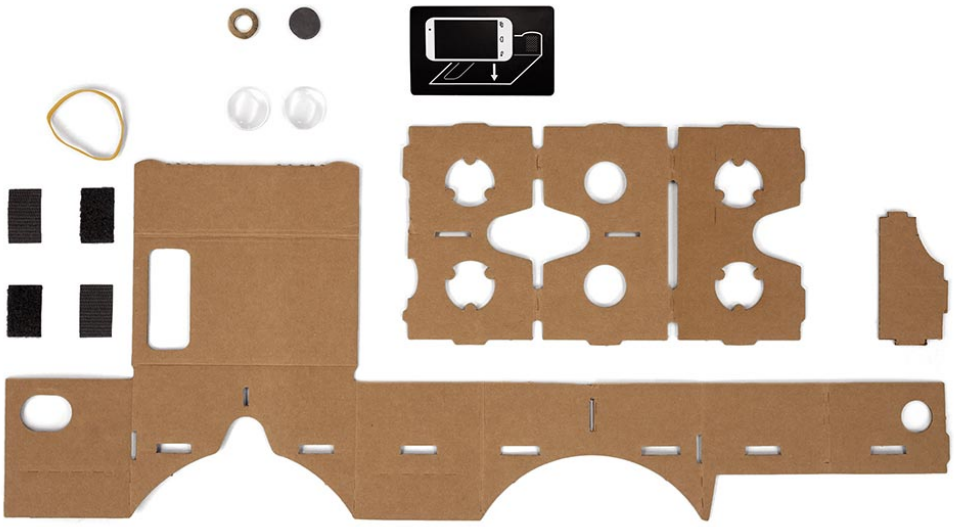


FIGURE 2.5: Constructing a Google Cardboard.

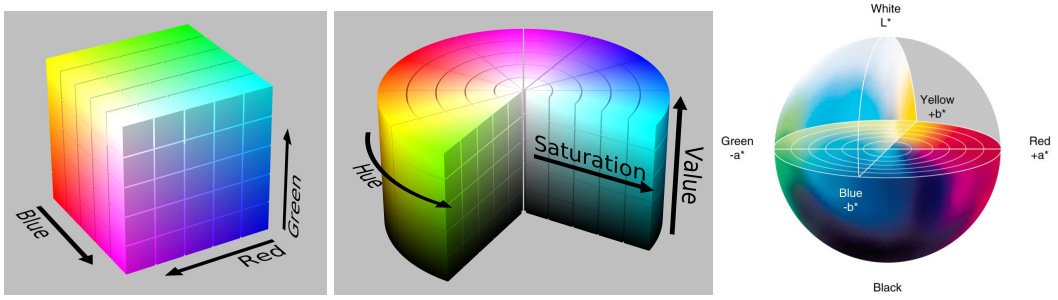
2.3 Feature Extraction

Feature extraction is a process to efficiently represent interesting parts of an image as a compact numerical feature vector, suitable for machine learning.

Note that global features such as colour histograms do not take spatial information into consideration which can result in very disimilar images

being matched due to their similar colour distribution. This problem can be solved by including other descriptors for spatial information or by partitioning an image, computing the colour descriptor on each partition and then concatenating to unify the sub-descriptors.

Colour Colour can be considered one of the most important features of clothing, especially for the purpose of image retrieval where it is the most extensively used feature. Furthermore, fashionistas commonly dress up or purchase new clothing items based primarily on colour (related to our application in chapter 3) and law enforcement commonly ask the public for help identifying a subject based on attributes including clothing colour (related to our application in chapter 6). Unlike humans, a computer sees colour just as numeric pixel values of a colour space. Thus, colour space must first be determined prior to selecting an appropriate colour descriptor for our algorithms. Three of the most important and relevent colour spaces, as shown in Figure 2.6, are now described.



(A) RGB (B) HSV (C) Lab

FIGURE 2.6: Colour spaces

The RGB space is composed of three colour components, red, green, blue, and it is by far the most popular colour space for image capture and display. A colour in RGB space is defined by adding together values of its components, thus it can be referred to as *additive primaries*. While the RGB space is easy to understand, being perceptually non-uniform it fails to mimic how humans perceive colour and it is also device-dependent.

The CIE Lab space consists of a lightness L component and two chromatic components a and b. A major advantage of Lab is that it is device independent and considered to be perceptually uniform.

HSV space also consists of two chromatic components for hue *H* and saturation *S*, as well as a lightness component called value *V*. An advantage with HSV is that it is very intuitive for computer vision applications such as image retrieval where we wish to retrieve an image based on illumination invariant hue. Less computation is required for the conversion of RGB to HSV than to Lab.

There is little agreement on which is the best all round color space. The literature on image retrieval, and its parent field of computer vision, employs a wide range of different color spaces, often with little or no reasoning behind the selection. State of the art considers HSV or Lab to generally achieve the best singular descriptor performance for image classification and retrieval tasks, and combining them together for better performance across a diverse dataset of real world images [117]. Thus, we choose to consider HSV and/or Lab colour descriptors in this dissertation.

Histograms are used to give an efficient estimate of the probability density of an underlying function, which in the case of this dissertation is the probability P of a pixel color C occurring in an image I.

Each pixel in an image can be described by the three components of the HSV or Lab colour space, so a histogram can be defined for each component. The main parameter for a 3-dimensional histogram is the number of bins in each dimension of the colour space. There is an important trade-off with this parameter. Too few bins and the histogram will be unable to disambiguate between images with significantly different color distributions. Likewise, with too many bins, images with very similar distributions may be regarded as not similar even though in reality they are, so we would like to match them. The number of bins for a colour histogram can be dependent on the size of the dataset and how similar the color distributions of the images in the dataset are. Larger datasets containing a wide range of colours will require a larger, more discriminative histograms. An iterative, experimental approach can be used to optimize the parameter.

Note that clustering can be used to determine k dominant colours for a particular region of interest over a training dataset. Then each of these dominant colours is taken as a histogram bin to minimize the number

of bins required and eliminate small noisy bins. Thus reducing computational cost in cases which would otherwise require a large number of bins. However, clustering dominant colours in this way assumes images in the test dataset will have the same dominant colours as the training dataset, and it is an extra computational step that is not needed if the bin size is already set relatively low.

2.4 Feature Encoding

2.4.1 Compressed Fisher Vectors

The Bag of Visual words (BoV) feature encoding technique is commonly adopted in the image classification and retrieval literature. The technique uses k-means clustering on a set of local features extracted from a large training set of images to generate a codebook of visual words. The local features of an image are each associated with their nearest visual word and the overall image is represented by a histogram of visual words.

The Fisher Vector (FV) [118] can be viewed as a generalization of the BoV framework. It is a powerful state of the art encoding technique that groups a variable number of local features into a single fixed size vector using the Fisher kernel. Compared to the Bag of Visual words, the Fisher Vector gives a more complete representation of the samples as it encodes not only the probabilistic count of occurrences but also higher order statistics based on its distribution with respect to the words in the vocabulary. This leads to a more efficient representation, since the vocabulary size to achieve a given performance is greatly reduced. The FV has been shown to outperform BoV and other encoding methods on a number of challenging datasets in terms of both classification accuracy and efficiency [119, 120].

Although the Fisher Vector is an ideal image representation for small to medium scale problems, it becomes impractical for large scale applications due to the storage requirements for the very high-dimensional and dense representation. For example, 4 byte floating point FV representations with 512K dimensions will require 2MB of storage per signature.

Storing the signatures for the 14M images of the ImageNet dataset would require around 27TBs. Handling TBs of data also has other side effects such as making experimentation very difficult and costly (particularly for cloud based machine learning systems), or in the worst case, impractical.

In addition, dense feature vectors residing in a high-dimensional space are not suitable for fast retrieval in a mobile visual search application. Ideally, we want a low bit rate image representation in order to satisfy the time and resource constraints of mobile visual search systems.

Perronnin et al. [121] address these problems by using Product Quantization (PQ), as an efficient and effective approach to perform lossy compression of FVs which enables balancing accuracy, CPU cost, and memory usage. This can be referred to as Compressed Fisher Vectors (CFV).

2.4.2 Pooling

A major challenge of object recognition is to generate feature representations that are robust to appearance variations. Pooling addresses this problem by transforming the overall feature representation into a more usable one that preserves the important discriminative information inherent to the set of encoded features while discarding irrelevant information. In particular, Spatial Pooling plays a key role in achieving these invariance properties by grouping local features within spatial neighborhoods. The Spatial Pyramid (SP) is a spatial pooling model introduced by [122] which has been shown to be effective for object recognition with Fisher Vectors [120]. It consists of subdividing the image into a set of regions and aggregating descriptor statistics over these regions, such that one FV is computed per image region and then the resulting FVs are concatenated to form one fixed length feature vector.

2.4.3 Normalization

We now describe two normalization steps which have been shown to benefit image retrieval and classification, especially when FVs are combined with a linear classifier.

Power normalization The power law or Signed Square Root (SSR) is a simple form of normalization that has recently gained popularity. The Euclidean distance is often used to compare low level features or higher level encoded features. This distance measure can easily become dominated by large feature values in a dimension of the feature vector, thus reducing the accuracy of image retrieval and classification. Unfortunately, it has been shown that large values often occur for low level descriptors [123] and BoV/FV encoding [121, 124–126].

However, by performing a power normalization of the form:

$$x \longleftarrow \text{sign}(x)|x|^{\rho} \qquad 0 < \rho \leq 1 \tag{2.4}$$

to each dimension of the FV or low level feature, this undesired effect caused by large values can be alleviated. Generally in literature, ρ is defined as $\rho = 0.5$ and this specific case is referred to as the Signed Square Root (SSR). Alternatively, a validation set can be used to tune the value of ρ.

ℓ^2 normalization Feature scaling/normalization is a widely used technique in machine learning to improve image retrieval and classification by standardizing the range of feature values. If a feature has a wide range of values, a classifier's distance metric will be governed by this particular feature. Therefore, by normalizing the range of all features, each feature should contribute approximately proportionately to the overall distance between vectors.

ℓ^p normalization, in particularly ℓ^1 and ℓ^2, is very commonly applied to all kinds of feature. We utilize ℓ^2 for this purpose of normalizing high-dimensional feature vectors and also due to its characteristic on FVs of cancelling out the fact that different images contain different amounts of background information [121].

2.5 Machine Learning

This dissertation is about learning from data. Generally we have a categorical outcome (such as clothing brand) that we wish to predict based on a set of features extracted from an image. Given a training dataset in which we observe the outcomes $Y = y_1, \ldots, y_n$ for various features $X = x_1, \ldots, x_n$, we can build a prediction model to predict the outcome for features in new unseen images. This kind of machine learning is referred to as *supervised* learning since the algorithm is guided or 'supervised' by the available outcome data. In this section, we briefly explore the background for some of the main machine learning techniques utilized later on in the dissertation.

2.5.1 k-Nearest Neighbours

The k-nearest neighbour classifier (kNN) finds the k closest features in the training set X to a query feature x and returns the majority vote of their labels. The principle is based on the intuitive concept that features of the same class should be closer together in the feature space. In the simplest case of $k = 1$, there is no voting and the predicted class $\hat{y}$ is directly given by:

$$\hat{y}(x) = y_{n^*} \tag{2.5}$$

where

$$n^* = \arg\min_{n \in X} \mathtt{d}(x, x_n) \tag{2.6}$$

The distance metric $\mathtt{d}()$ is used to calculate similarity between features. For real-valued feature vectors of dimension d, Euclidean distance is commonly used:

$$\mathtt{d}(\mathbf{x}, \mathbf{x}_n) = \|\mathbf{x} - \mathbf{x}_n\|^2 = (\mathbf{x} - \mathbf{x}_n)^T(\mathbf{x} - \mathbf{x}_n) = \sum_{i=1}^{d}(x(i) - x_n(i))^2 \tag{2.7}$$

An example of kNN classification is given in Figure 2.7. The green circle represents the test sample that should be classified as either the blue or red class. If $k = 3$ (solid line circle) it is assigned to the red class because there are two red classes and only one blue class inside the inner circle. Whereas, if $k = 5$ (dashed line circle) it is assigned to the blue class since there are more blues than reds inside the outer circle.

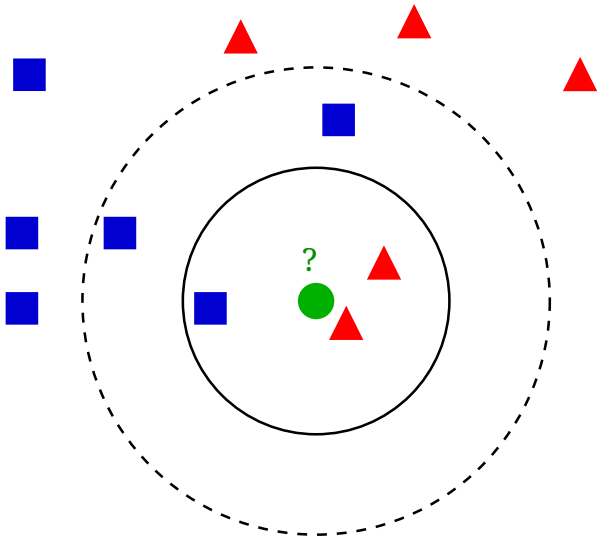


FIGURE 2.7: k-nearest neighbour classification example. By Antti Ajanki, licensed under CC-BY-SA-3.0.

The value of model parameter k is an important choice. As k increases, the model averages over more data yielding smoother predictions, otherwise for small values of k, the result will be more highly influenced by noise. A large value makes the model very computationally expensive.

k-dimensional tree (kd tree) is a useful space-partitioning data structure that addresses the computational inefficiencies of the naive brute-force approach for finding nearest neighbours. The feature vectors in the training dataset are recursively split across a binary tree (similar to Figure 2.8, discussed further in the next section) by thresholding the vector dimensions. The threshold is generally chosen at the dimension with the maximum variance. Finally, the inner tree nodes terminate to leaf nodes which store the features. A single leaf node representing the nearest neighbour of a query feature is found by traversing the tree with the partitioned query feature, comparing the values at each inner node in the tree.

For n samples in d dimensions, the kd tree approach can dramatically reduce the computational cost of a nearest neighbors search from $O(dn)$ to $O(d\log(n))$ for small $d < 20$. This comes at the expense of increased training time and complexity. When the dimension is very high, performance and speed can significantly decrease. To rectify this, improvements have been proposed such as creating multiple trees [127] or using fast approximate k-NN search by locality sensitive hashing (a randomized technique) [128]. Furthermore, the dimension can be reduced in a pre-processing step, by using principal component analysis (PCA) for example.

2.5.2 Random Forests

Random Forests (RF) combine the concept of bagging, a technique for reducing variance of a prediction function, with the random selection of features in order to construct a collection of de-correlated decision trees with controlled variance.

A decision tree is a hierarchical structure of simple binary decision functions containing two types of node. As depicted in Figure 2.8, the single root and multiple inner nodes function as binary *splitting* nodes which are eventually *terminated* into leaf nodes. For classification purposes, each leaf stores the probability distribution of class labels of the training samples that reached it. During testing, a splitting function $f(x) \rightarrow \{0, 1\}$ is evaluated on sample x at the root and each of the inner nodes.

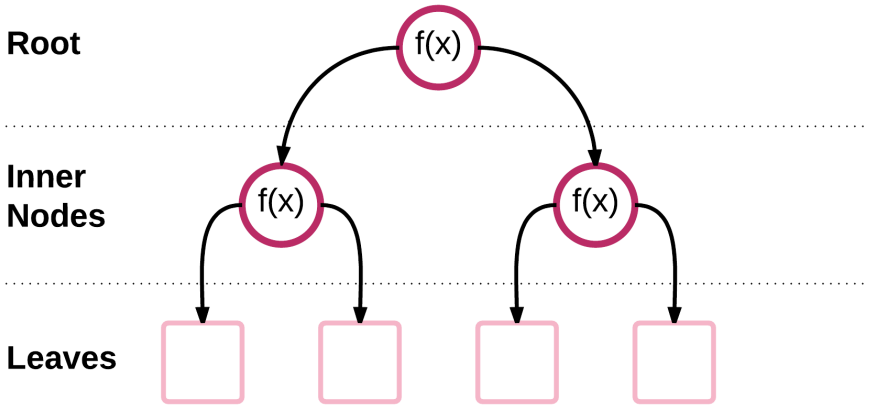


FIGURE 2.8: Binary decision tree structure.

A major problem is that a single tree significantly over fits the training data. This led to the random forest proposition of combining several decision trees with randomized splitting functions and training each tree on a random subset of the training dataset [129]. To classify a new image from an input feature vector, the input vector is put down each of the trees in the forest. Each tree votes for a class. The forest chooses the classification having the most votes over all the trees in the forest. Each tree b of B trees in the random forest $T = \{T_b\}$ is grown as follows:

For $b = 1, ..., B$:

1. If the training set has a total of N samples, choose the sample of the training set for growing this particular tree by sampling N cases at random but with replacement.

2. If there are M input features, a constant $m << M$ is chosen for the forest growing such that at each node, m features are selected at random out of total M and the best split on these m is used to split the node.

3. Each tree is grown until a stopping criterion, such as minimum node size, is met. Then, each leaf l stores the distribution $p(y|l)$.

The forest error rate depends on two things: the correlation between any two trees in the forest and the strength of each tree in the forest, where a tree with a low error rate is known as a strong classifier. The overall error rate increases when the inter-tree correlation increases or when the strength of individual trees decreases.

Chapter 3

Product Retrieval

Mobile visual clothing product search is an important task due to the continued growth of the global clothing industry, increasing popularity of e-commerce, and dramatic growth in tablet and smart phone sales. The goal is that a user can simply take a photo of someone wearing a product of interest or select a social networking photo on a mobile device and an app will quickly identify the product and/or retrieve visually similar products from retailers.

Although there are now a number of related works, the majority of these methods do not consider the discrepancy between the user-provided query photos and clothing product images from e-commerce websites. They also tend to extract features without prior clothing segmentation and thus there can be inaccuracies from extracting features on background pixels within the expected clothing region.

The main contributions of this chapter of the dissertation are as follows: (1) we present a novel mobile client-server framework for automatic visual clothes searching; (2) we propose a dominant colour descriptor for the efficient and compact representation of clothing; and (3) we have evaluated our approach on query images from a fashion social network dataset along with a clothing product dataset for results and shown promising retrieval results with a relatively fast response time. The contributions in this chapter thus reside in a mobile system for automated clothes search with proven capability.

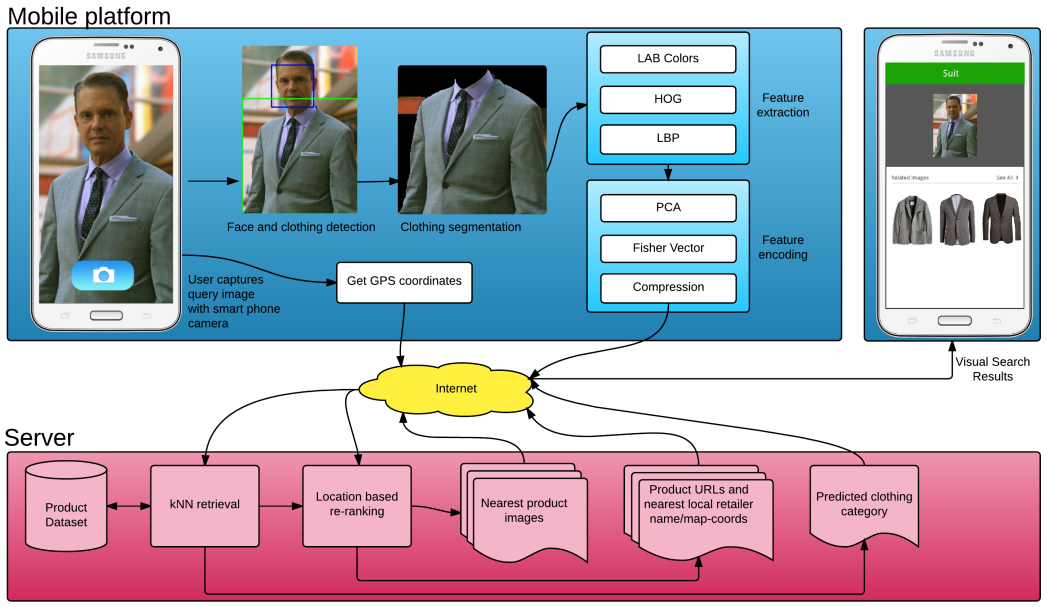


FIGURE 3.1: Overview of the mobile visual clothing search pipeline.

The pipeline for our mobile visual clothing search system to retrieve similar clothing products in nearby retail stores is shown in Figure 3.1. A smart phone user can either capture a photo of a person wearing clothing of interest or choose an existing photo, such as from a social network. The person is then detected in the image and clothing segmentation is performed to attempt to select only the clothing pixels for feature extraction. Note that we only consider searching upper body clothing since the images in our social networking dataset indicate that many people just take upper body fashion photos or *selfies*. The segmented upper body clothing image is divided up into non-overlapping patches and features (dominant Lab colours, HoG, LBP) are extracted to describe the shape, texture and colour of the clothing. To enable large scale retrieval on mobile devices, PCA reduces the feature dimensions, Fisher Vectors encode the features, and the encodings are compressed by product quantization. The compressed features are sent to the server alongside the phone's GPS coordinates. Similar products are retrieved from the database, re-ranked by retailer location and the resulting top product images (including product URLs) are downloaded to the smart phone. The user can then view a map detailing where they can locally purchase the products or click their associated URLs for purchasing online.

It is impractical to store large scale databases of clothing products from various retailers on the mobile phone client. Thus, a client-server architecture is conceived for our mobile visual clothing search, rather than being fully offline with no server infrastructure.

Our system is designed to be efficient with short response times and offer an interactive graphical user experience after results are retrieved. The client communicates with the server using compressed feature information rather than directly uploading a large query image. This is especially important since the majority of smart phones now have high megapixel cameras which can capture images requiring over 12MB of storage each. Hence, uploading an image this size (or even if resized) would be slow over a mobile network. Our approach allows for fast transmission on typical 3 or 4G mobile networks and has the additional benefit of distributing processing load between client and server so that the server may handle more simultaneous search requests. Our contributions are described in the following sections.

3.1 Datasets

The goal of our retrieval problem is to find the product images in the online shopping domain that correspond to a given query photo in the *street* domain that is uploaded by the user. Therefore, two different kinds of data are required.

Several clothing datasets exist but none of them appear to be suitable to evaluate our clothing retrieval task. Datasets mentioned in the current literature either do not solely contain frontal poses [5], do not feature a large range of clothing and people, are for the classification task, or are private.

For the query dataset (DQ), we collected a subset of 914 images from Chen's Clothing Attribute Dataset [12]. First, we discard all the images where the Mechanical Turk users were unable to agree on the main clothing category present in each image. The category labels for these undecided cases were stored as NaNs, so discarding them allows us to quantitatively evaluate

TABLE 3.1: Query dataset classes and number of images per class.

Category	Images
Shirt	112
Sweater	75
T-shirt	102
Outerwear	171
Suit	196
Tank Top	53
Dress	205

every image using our retrieval approach. Additionally, we discard all images where a face cannot be detected by ViolaJones in the top half of each image. This step is required since our approach is reliant on fast face detection rather than slow pose estimation for initialisation. Table 3.1 lists the clothing categories present in our dataset and the number of images per category class.

For the shopping dataset (DP), we consider real e-commerce images from esprit.co.uk. For this dataset, we collected 994 images equally divided between the 7 categories used in our query dataset, such that there are 142 per class. We also store their associated product URLs (so visual retrieval results can link to their product page for details and purchasing).

Furthermore, we select a small subset of query images from the Fashionista dataset [5] that exhibit frontal poses suitable for our Viola-Jones face detector. We use these images for the purpose of qualitatively demonstrating the cross-scenario retrieval scenario.

3.2 Pre-Processing

Real-world images captured on a smart phone or downloaded from online social networks provide a number of challenges to computer vision algorithms such as a wide variety of lighting/shading in the scene (referred to as non-uniform illumination) and image dimensions. To address these issues, the image is normalized to have a maximum side length of 600 pixels and the illumination colour channel is normalized in HSV colour space.

In order to locate and process clothing in the photo, first we must locate a person in the query image who is wearing the clothing of interest. The Viola-Jones face detector is used to estimate the face size and location which are fed as parameters to initialise a simple human detector. Our human detector [4] yields an approximate bounding box for the person's full body pose excluding head, ROI_p, and a smaller upper body only region, ROI_u. For efficiency, we constrain the face detector to the top half of the image and find this to be a valid assumption for our dataset.

Note that an alternative is to estimate the full pose of the person in order to more precisely localize the clothing features, but we found the popular state of the art pose estimaters such as [130] to be unsuitable for real-time use, particularly within a fast mobile framework. This is confirmed by Yamaguchi et al. [48] who state that full body pose estimation is one of their main bottlenecks. We explore a more accurate yet efficient solution to this in chapter 6 by utilizing depth information. Whereas in this chapter, we focus on using currently available technology that is built into tablets and smart phones.

3.3 Clothing Segmentation

Query Image Due to the inherent nature of using an approximate bounding box, some pixels belonging to the background will appear in the box as well as the pixels belonging to the foreground clothing that we are interested in. If features are extracted directly from this bounding box, it will lead to inaccurate classification since parts of the background will be considered as clothing. To address this, we attempt to automatically segment the clothing from the person and background within the larger bounding box ROI_p by using the efficient yet high performance DenseCut algorithm [131]. A probability mask is created to initialize the segmentation. It is created such that it labels the pixels within the middle half of the upper body box ROI_u as *foreground* and the pixels from the top of the image to the centre of the detected face as *background*.

We further attempt to eliminate the skin from the segmented person by employing an efficient thresholding method. Chai and Ngan [132] reported

that skin pixels on the face can be identified by the presence of a certain set of chrominance values in the YCrCb colour space and utilized for face detection purposes. Based on this work, we propose a thresholding method for the purpose of clothing segmentation that takes into account other skin pixels on the body. This can be more challenging as we find illumination on the face tends to be more uniform. Consider R_r and R_b as ranges of the respective Cr and Cb values that correspond to the colour of skin pixels. For a random sample of our social networking dataset, we found ranges of $R_r = [140\,165]$ and $R_b = [105\,135]$ to be optimal. In our experiments, these ranges prove to provide a good compromise between robustness against different types of skin colour and attempting to preserve clothing pixels of similar chrominance to the skin. Thus, we have the following equation:

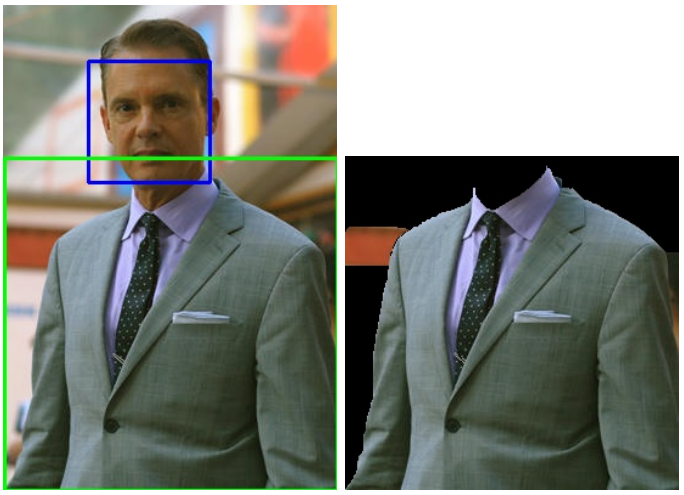
$$\text{skin}(x,y) = \begin{cases} 1 & \text{if Cr}(x,y) \in R_r \cap \text{Cb}(x,y) \in R_b \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where x and y are pixels in ROI_p. Morphological opening is then performed on the binary mask $\text{skin}(x,y)$ to reduce noise.

Finally, the segmented full body clothing is cropped to the upper body region ROI_u and normalised in size. The area of segmented clothing is compared to the area of the ROI_u. If the percentage of clothing pixels is less than an empirically defined threshold τ_a, we perform the next stage (feature extraction) on the DenseCut image rather than the skin elimination result. This final step can increase overall robustness of the system to the special case where the clothing and skin are of a very similar colour. The resulting upper body clothing image is denoted I_c.

Figure 3.2 depicts the clothing segmentation process on a challenging real-world image.

Database Product Images E-commerce clothing websites typically display product images against a clean solid background. A colour histogram is computed on each product image in the database and we can assume that the peak of each image's histogram identifies its background colour. The product mask can be obtained by comparing each pixel in the image against the estimated background colour. Figure 3.3 depicts this process.



(A) Query image (B) Segmented

FIGURE 3.2: Clothing segmentation of a query image given face detection (blue) and estimated upper body bounding box (green).

(A) Product image (B) Segmented

FIGURE 3.3: Clothing segmentation for product images.

3.4 Clothing Features

Within the upper body bounding box that was detected in the previous step, low-level clothing features are densely sampled on a grid. These features include HOG [133], LBP [134], and our dominant Lab colour descriptor. The grid is constructed such that the upper body clothing image I_c is divided up into a regular grid of 4×5 cells.

We propose an efficient method to compactly describe dominant clothing colours based on the MPEG-7 descriptor [135]. We denote each column of

the grid as ROI_c^k where $k = 1 \dots 5$ and we propose providing robustness to layered clothing (e.g. jacket and top) by computing the dominate colours for each column and concatenating.

A 3D histogram is computed on $I_c \in \text{ROI}_c^k$ in HSV colour space. For clothing, hue quantization requires the most attention. We find a quantization of the hue circle at 20° steps sufficiently separates the hues such that the red, green, blue, yellow, magenta and cyan are each represented with three sub-divisions. Also, saturation and illumination are each quantized to three sub-divisions. Hence the colour is compactly represented with a vector of size $18 \times 3 \times 3 = 162$.

The quantized colour of each colour bin is selected as its centroid. If we let C_i represent the quantized colour for bin i, $X = (X^H, X^S, H^V)$ represent the pixel colour, and n_i be the number of pixels in bin i, we can calculate the mean of the bin's colour distribution as follows:

$$C_i = \bar{X}_i = \frac{1}{n_i} \sum_{j}^{n_i} X_{i,j} \quad , \quad 1 \leq i \leq 162 \tag{3.2}$$

Ideally, the dominant colours would be given by bins with the greatest percentage of image pixels. However, in practice, due to factors such as uncontrolled illumination, bins of similar quantized colours often exist per perceived clothing colour. Therefore, the mutual polar distance between adjacent bin centres is iteratively calculated and compared with a threshold, τ_d, and similar colour bins are merged using weighted average agglomerative clustering. Considering X_1 and X_2 in the adjacent bins, we let P_E represent the pixel percentage of the colour component E and perform the following equation for each colour component, substituting E for the H, S, and V components respectively:

$$X^E = X_1^E \left(\frac{P_1^E}{P_1^E + P_2^E} \right) + X_2^E \left(\frac{P_2^E}{P_1^E + P_2^E} \right) \tag{3.3}$$

Bins with a pixel percentage less than τ_p are considered insignificant colours and merged to their closest neighbour bin. Since each set of worn upper body clothing in our product dataset is humanly perceived to generally have less than 3 dominant colours per ROI_c^k, thresholds τ_d and τ_p are empirically defined to yield approximately this amount of dominant

colours. For the purpose of our similarity stage, we convert the polar HSV colours to the Euclidean LAB space and the represent the dominant colours F_k^c as:

$$F_k^c = \{(C_1^L, C_1^A, C_1^B, P_1), \ldots, (C_n^L, C_n^A, C_n^B, P_n)\} \tag{3.4}$$

where (C_1^L, C_1^A, C_1^B) is a vector of LAB dominant colour, the corresponding percentage of that colour in the clothing is given by P_1 and $0 > n \leq 3$ is the number of dominant colours on the clothing. For our application, we generate $F^c = \{F_k^c\}$ (padding each F_k^c if necessary) to yield total dimensions of $4 \times 3 \times 5 = 60D$.

Texture/shape features based on histogram of oriented gradient (HoG) and local binary pattern (LBP) are computed in each cell on I_c. HoG gradient orientations are quantized to every $45°$, thus there are 8 direction bins. The local histograms of the cells are then concatenated together to form the $8 \times 20 = 160D$ HoG feature F^h. For LBP, we extract a 26D feature on each cell using the uniform LBP implementation which offers improved rotation and grayscale invariance. Thus, we have an overall $26 \times 20 = 520D$ LBP feature F^l

3.5 Clothing Similarity

Once the low-level features sets have been extracted, we use them to construct a signature to describe the image.

First, PCA is performed on the feature descriptors to simplify the representation and reduce redundancy in the data. By reducing the dimensions to 64 with PCA, this helps to speed up the next step involving Fisher Vectors (FVs) since the FV size scales linearly with feature dimension. Additionally, it helps satisfy the FV diagonal covariance matrix assumption.

We use Fisher Vectors [118] to construct the overall image signature since they have been found to be the most effective in a recent evaluation study of feature pooling techniques for object recognition and retrieval [119, 120].

The FV is a generalization of the very popular Bag Of Visual words (BoV) representation. In the case of BoV, the local feature descriptor is quantized according to k-means clustering on a large dataset of image descriptors. Whereas, for each quantization cell of FV, not only are the number of assigned descriptors stored, but also their corresponding mean and variance in each dimension. For K quantization cells and D dimensional descriptors, this yields a signature with dimensions $K(2D+1)$. Since more data is stored per cell, less quantization cells can be used than for BoV, making the FV signature more compact and faster to compute. Unlike k-means clustering in BoV, the FV representation uses Gaussian mixture clustering. For training the GMM, a random subset of features are extracted over all the images in the dataset.

We compute FVs for the PCA representation of the low-level features. Then power and L2 normalizations are applied on the FV, which has been shown to significantly improve the performance. Finally, the FV is compressed by Product Quantization (PQ) [121] in order to reduce CPU and memory/bandwidth requirements.

For retrieval of similar product images to a given query image, we utilize k-Nearest Neighbours (kNN). For efficiency and fast searching of large scale databases, we implement a kd-tree to index the clothing images. The similarity measure that we compute for the kNN is the fast and popular L2 distance metric. For cross-scenario retrieval, let H_j be the Compressed FV (CFV) representing the query image and H_j be the CFV for the j^{th} image in the product dataset. Thus, the index of the top result is given by:

$$\hat{j} = \arg\min_{j} \texttt{dist}(\mathbf{H_q}, \mathbf{H_j}) \tag{3.5}$$

where

$$\texttt{dist}(\mathbf{H}_q, \mathbf{H}_j) = \|\mathbf{H}_q - \mathbf{H}_j\|^2 = (\mathbf{H}_q - \mathbf{H}_j)^T(\mathbf{H}_q - \mathbf{H}_j) = \sum_{i=1}^{d}(h_q(i) - h_j(i))^2 \tag{3.6}$$

where d is the dimension of the CFV.

A further background on FV feature encoding, compression, kNN, and kd-tree based retrieval can be found in chapter 2.

3.6 Experimental Results

We consider evaluation with respect to within-scenario and cross-scenario, by using the DQ query set by itself and alongside the DP product set, respectively.

3.6.1 Quantitative

For quantitative evaluation, we focus on the within-scenario for the street (DQ) dataset since this poses the most challenging images for both querying and retrieving compared to the product dataset (DP).

Given a query image I_q, the retrieval procedure can rank all n images in a dataset by similarity. If we denote $\mathrm{Rel}(i)$ as the groundtruth relevance between q and the i^{th} ranked image, we can use a precision to evaluate the ranking of the top k retrieved images with respect to the query I_q:

$$Precision@k = \frac{\sum_i^k \mathrm{Rel}(i)}{N} \tag{3.7}$$

In order to ensure that correct ranking results in a precision score of 1, we denote N as a normalization constant.

Our street (DQ) dataset labels the single most dominating upper body clothing category in each image. Therefore, $\mathrm{Rel}(i)$ will yield a binary value as we only consider one attribute of the query image.

Considering within-scenario for the street dataset, we achieve a Precision@5 of 0.61. This result is roughly comparable to the state of the art approaches which also use challenging real-world street datasets (such as [52]), but our approach also has increased efficiency and is designed for a mobile infrastructure. We focus on $k = 5$ since the top results are the most important, especially for our mobile application where screen size is very limited for displaying visual results. Note that when k is large enough, the precision will ultimately increase to near 1.

3.6.2 Qualitative

Our retrieval results are reported qualitatively in Figures 3.5 and 3.6 for the within and cross scenario, respectively. We can see that the results appear promising with relevant clothing results of a similar colour and shape/texture to the query image generally retrieved.

The upper body clothing detection and segmentation stages are critical since they initialize the system. If they are inaccurate, significant errors can be propagated forward to the rest of the system.

Inaccuracies in the dimensions of the bounding box generated by upper body clothing detection can occur when the face is at an angle and not directly aligned with the camera sensor, or in other cases where ViolaJones face detection can be inaccurate. To address this, a more comprehensive upper body person detector could be employed alongside non-maximum suppression to select the best bounding box candidate, but this is challenging to implement on mobile, and whilst it can be less computationally intensive than full body pose estimation, it is still a relatively intensive operation with respect to our overall approach.

Segmentation inaccuracies appear to generally be caused by inherent issues such as when scenes contain a garment that is a very similar colour to the background or skin, or there is poor illumination present, or significant clothing occlusions, such as long hair. However, when skin segmentation fails and our algorithm decides to instead use the initial segmentation result to establish features, such as in Figure 3.6i, we see that the results can still be reasonably relevant although may not be the most accurate.

3.6.3 Implementation

The server stage is implemented in Python and C++ and deployed on a 2.93GHz CPU. A graphical user application is designed for the client side which is implemented in Java and C++ using the Android SDK, NDK, and OpenCV library. Hence, the demonstration application is intended for Android smart phones - specifically, we consider the popular Samsung Note 4 (Quad-core 1.3 GHz Cortex-A53 and Quad-core 1.9 GHz Cortex-A57) for

(A) (B) (C) (D)

FIGURE 3.4: Mobile application: (a) home, (b) search query, (c) search results, (d) local retailer location for top product result.

demonstration and timing analysis. For demonstration, we design features such as photo querying, viewing top search results, product information (by linking to the relevent product URL), and displaying similar products from nearby local retailers on a map. Screenshots for these different aspects of the app can be seen in Figure 3.4. For evaluation purposes, we set all products to one retail location so that only the more important visual relevance is evaluated and not location relevence. Otherwise, the map coordinates of the retailer's local stores can be used and compared by the system to the GPS location of the smart phone user. Note that the server also sends the clothing class label belonging to the top retrieved result to the mobile client as a rough classification that can be displayed in the app.

3.6.4 Computational Time

Our system takes on average approximately 1 seconds for client processing. Although, we do not fully investigate transmission timing, our system can achieve a total response time of 3 seconds to retrieve results from the server across a 4G data network with excellent smart phone reception. Table 3.2 lists the computational times of the various stages of the system performed on the client and server. For reliability, the average timings consider a random sample of 10 images with each image in the sample being processed 10 times. These results show that the feature extraction is

TABLE 3.2: Computational Time

Client	Time (s)
Person Detection	0.16
Clothing Segmentation	0.21
Feature Extraction	0.43
Feature Encoding	0.05
Server	Time (s)
Search and re-ranking	0.22

our biggest bottleneck. This can be improved by optimizing the code. Our approach is slower than the real time work of [9], however their approach is for a different application, is not implemented in a mobile framework and their dataset is captured on a clean white background. Our approach is much faster than the work by [5] which works offline, requiring $2-3$GB of memory.

3.7 Summary

In this chapter, a complete novel mobile client-server system is presented for automatic visual clothes searching of challenging real-world images. A smart phone user can capture a photo (or select a social networking photo) of somebody wearing clothing they like and retrieve similar clothing products that are available at nearby retailers. Our system first identifies the clothing region in the image and segments it from the background using a fast DenseCut implementation. HoG, LBP and novel Lab features are extracted to describe the clothing shape, texture and colour. To enable large scale retrieval on mobile devices, PCA reduces the feature dimensions, Fisher Vectors encode the features, and the encodings are compressed by product quantization. The compressed features are sent to the server alongside the phone's GPS coordinates. Similar products are retrieved from the database, re-ranked by retailer location and the resulting top product images (including product URLs) are downloaded to the smart phone. The user can then view a map detailing where they can locally purchase the products or click their associated URLs for purchasing online.

Parts of the research in this chapter have been published in [4] and notably cited by Yamaguchi et al. [27] and Jia-Lin Chen et al. [26]. Our work remains novel as our paper has been improved upon in this dissertation and it also targets the specific application of mobile retrieval with a highly practical, efficient and unique mobile client-server framework. An avenue for future work is to collect our own large scale dataset to address the lack of such datasets in the retrieval field and perform a more comprehensive evaluation.

FIGURE 3.5: Within-scenario results: a sample from each clothing class is queried and the top 3 upper body clothing photos are displayed. Retrieved images that do not match the query clothing category are highlighted.

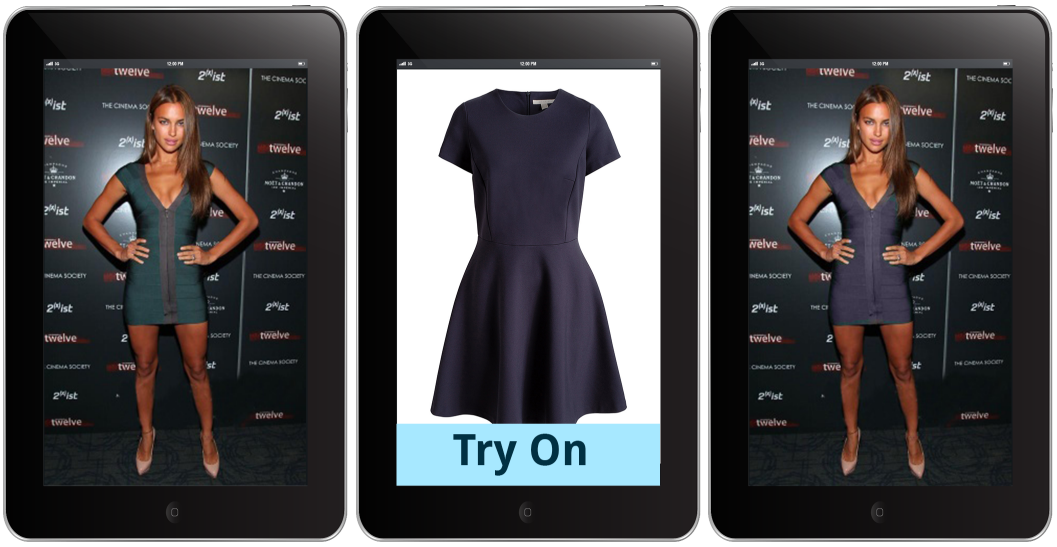
FIGURE 3.6: Cross-scenario results: examples of top retrieval candidates. Note that whilst the worn product images are depicted on the right in each query/result pair, their corresponding unworn product images are used for feature extraction.

Chapter 4

Augmented Reality Mirror

In this chapter, a new highly efficient approach is proposed for clothing parsing on mobile devices, based on the discussion of clothing and mobile devices in chapter 1. Related approaches are also discussed in chapter 3 and chapter 6 for use in product retrieval and person identification applications respectively. Our new approach leads to a demonstration of recoloring and retexturing upper body clothing in an augmented reality mirror application for mobile tablet devices. A tablet user places the tablet in its stand on a desk, selects an ecommerce product with the desired colour/-texture to try on, and then the tablet acts similarly to a mirror, displaying a live fullscreen video stream from the integrated camera augmented with the rendered product colour/texture as shown in Figure 4.1. The system also predicts semantic attributes for clothing colour, neckline, sleeve length and brand, which may be useful in applications such as fashion analysis, customer profiling, and clothing retrieval.

Augmented Reality (AR) is the concept of adding virtual elements to real world scenes. Virtual Try On (VTO) is the concept of allowing a user to try on garments virtually to check the fit and look of garments on a textured body model of themselves usually via an internet connected PC without needing to visit the retailer. In contrast to AR, the VTO concept is traditionally a virtual reality environment in which the real world is entirely replaced with a simulated one where the scene, user and clothing are modeled and computer generated.



(A) Real-time mirror (B) Product selection (C) Re-coloured dress

FIGURE 4.1: Our mobile augmented reality mirror application.

Virtual try on is revolutionizing the clothing retail and fashion industries [158]. However, existing VTO methods can be complex to implement and are computationally intensive requiring an offline setup with manual user interaction for 3D body modeling and garment design. It is also challenging to model and texture a 3D representation of the user which is both photorealistic and anthropometrically accurate given a practical non-invasive consumer driven input such as a single 2D photo. *Augmented Try On (ATO)* can be achieved by fusing the concept of AR with VTO, allowing for systems which are more real-time and photorealistic to be designed with a focus on visualization of the look/style of the retextured garment worn by the user rather than on checking the fit. This fusion may be particularly well suited for visualization of printed textures on common garments whose designs do not vary as much as other types of garment and for which users will likely already know their size. It is designed to be practical for a consumer and unlike most related work which utilizes high-end computer vision cameras and special lighting, no expensive high-end hardware is required or used for obtaining results.

Segmentation of the textured garment is required for our augmented reality framework in order to find the flexible textured surface which we wish to reconstruct and also for allowing the garment to be recolored during rendering. We consider the segmentation of loose T-shirts since they exhibit highly non-rigid properties and often feature a printed texture,

which is required for our reconstruction and retexturing methods. The segmentation of clothing worn on a subject is challenging due to the wide diversity of clothing designs, the complexity of scene lighting, dynamic backgrounds, and self/third-party occlusions.

The main previous real-time work by Hilsmann and Eisert [51], described in Section 2.1.1, has major aforementioned *a prioris* such as T-shirt color and a simple rectangular texture, and the authors only show subjective visual results for a specific T-shirt. They do not take advantage of HSV-colorspace which features improved hue invariance to lighting over RGB-colorspace by separating the illumination channel from the color. Our method is perhaps more closely related to the hue analysis part of the CamShift algorithm [136].

We describe a clothing segmentation method for single images and video which can yield semantic per pixel labels for upper body clothing of persons in real-time, as summarized in Figure 4.2. Our approach is primarily designed to benefit mobile devices and emerging augmented reality applications [21]. These augmented applications include computer gaming and augmenting localized adverts or statistics onto players' shirts for close-up shots in live TV/internet broadcasting. Shirts worn by sports teams, such as in major league basketball, are often uniformly coloured with text and logos to indicate the player, team, and sponsors. For this reason, we focus on the case of predominantly monochromatic tops, and attempt to additionally segment any textures on them which can be useful for the purpose of retexturing.

Thus, the main contributions can be summarized as:

1. An efficient automated method for accurate semantic segmentation of persons wearing primarily uniformly coloured upper body clothing which may contain textured regions. Semantic attributes for clothing colour, neckline, sleeve length and brand are predicted. Spatial priors are employed and each set of resulting cloth and texture contours are semantically labelled as such and associated to a face. Unlike most previous work which evaluates visually or with respect to applications (such as recognition), we evaluate the segmentation directly and quantitatively against a dataset of 100 people.

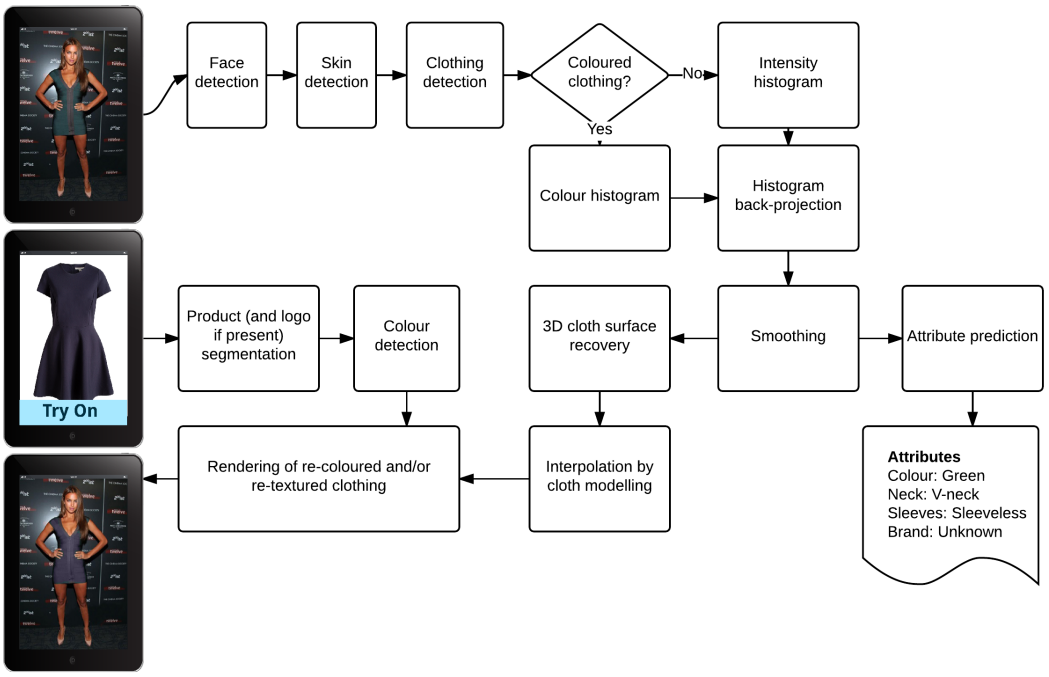


FIGURE 4.2: Clothing parsing and surface reconstruction for our mobile augmented reality mirror application.

2. An initialization scheme where initial points on the cloth are located by estimating skin colour and employing an iterative colour similarity metric to locate the clothing. This can prevent initializing cloth points on the skin in the case of clothing with deep neck lines such as vests and many female tops.

3. A mobile augmented reality mirror application to demonstrate the method.

4.1 Datasets

A clothing dataset consisting of people in frontal poses wearing predominantly uniformly coloured upper body clothing is required to evaluate our model. For this purpose, we utilize the Images of Groups [157] dataset. The dataset features challenging real-world Flickr photos of groups of people, enabling our method to detect and parse multiple persons per image.

A testing subset of 100 persons is formed from images featuring predominantly uniformly coloured upper body clothing and does not feature

groups where clothing of adjacent persons is of a very similar colour and in direct contact. A training subset of 50 persons is randomly selected from remaining images in the dataset for use in Sections 4.2.2 and 4.2.3.

For demonstration only purposes, we also collect several Creative Commons licensed images from Flickr. This enables us to show a wider variety of challenging clothing parses.

For the evaluation of clothing logo recognition, the *FlickrLogos-32* is a popular dataset, but unfortunately the classes are too general (not clothing focussed) and the class set does not consist of the logo classes found in the aforementioned clothing dataset. Also, we are interested in clothing textures where the texture is not necessarily a logo or solely consisting of a logo, such that the brand can be inferred if the texture is a unique design. Thus FlickrLogos is unsuitable for this task. In this case, we collect a small dataset of 75 clothing texture samples exhibiting 10 logo classes, plus an extra *unknown* class containing 15 images of other logos and cloth. To collect the images, we design a script to crawl Google Images using keywords such as *Hollister T-shirt* and manually ensure that the dataset includes instances where the cloth is deformed or partially self-occludes the logo. We keep the dataset small to enable fast research and prototyping and suggest it can be easily extended as future work.

4.2 Clothing Parsing

4.2.1 Pre-Processing and Initialization

For pre-processing, the single image or video frame is converted from RGB to the more intuitive and perceptually relevant HSV colour-space. The corresponding illumination channel and image dimensions are then normalized, giving image N. This helps to alleviate, to some extent, the non-uniform effects of uncontrolled scene lighting. Additionally, a 3×3 box blur is performed as a simple denoising measure, yielding image I. We let the H, S, and V channels correspond to I_0, I_1, and I_2 respectively and use the OpenCV HSV intervals $I_0 = [0, 180]$ and $I_{1,2} = [0, 255]$. For our image notation, we also refer to the origin as the top left of the image.

A chromatic/achromatic mask is defined where achromatic pixels are those with illumination extremes or low saturations:

$$\texttt{chrome}(I) = 0 \leq I_0 \leq 180 \wedge 26 \leq I_1 \leq 255 \wedge 26 \leq I_2 \leq 230 \tag{4.1}$$

Viola-Jones face detection is performed on image N as a prerequisite for our segmentation approach. This technique is based on a cascade architecture for reasonably fast and accurate classification with OpenCV's popular frontal face trained classifier cascade. We limit the region of interest for object detection to the top half of the image in order to further increase efficiency. For each face detected, the segmentation procedure in the following sections is performed.

4.2.2 Spatial Priors

To increase robustness against hues/intensities in the background which are similar to those on the clothing, and to increase computational efficiency, we constrain segmentation of each person to a region of interest (ROI). The size of this region is determined by detecting faces in our training dataset (see § 4.4) and studying the upper body clothing bounds, given by anatomy and pose, relative to the detected face size and position. As a result of these studies, spatial priors are defined as 5 times the detected face height and 4.5 times the face width and positioned as follows:

$$\begin{aligned}\texttt{crop}(I) = \texttt{Rect}(&\texttt{Point}(F_x - 1.75F_{width}, F_y + 0.75F_{height}),\\ &\texttt{Point}(F_x + 2.75F_{width}, F_y + 0.75F_{height} + 5F_{height}))\end{aligned} \tag{4.2}$$

where the F vector for each person is output by face detection. The bounds of the ROI are also clipped to within the image dimensions.

4.2.3 Locating Points on the Clothing

Points on the clothing are required in order to initialise segmentation. Previous work often employs a scaled distance from a detected face to achieve this. However, this approach is susceptible to initialising clothing

points on the skin in the case of clothing with deep neck lines such as vests and many female tops, and hence the segmentation has reduced accuracy. We propose a solution to this problem. The faces detected on the training dataset in the previous section are scaled to within 80×80 pixels, whilst maintaining their aspect ratios. We study the average face and define a region which tends to primarily be skin pixels and avoids occlusion by long hair:

$$\texttt{FSkin}(I) = \texttt{Rect}(\texttt{Point}(F_x + 15s, F_y + 36s), \texttt{Point}(F_x + 65s, F_y + 56s)) \quad (4.3)$$

where the scale factor $s = F_{width}/80$. The skin colour α is estimated by computing the mean of the pixels in the $\texttt{FSkin}(I)$ region.

A sparse iterative procedure is established across the $x = [F_x, F_x + F_{width}]$ and $y = [F_y + F_{height}, F_y + 2F_{height}]$ intervals, shifting a 5×5 pixel window. During each iteration, the mean colour β of the window is computed. The HSV colour similarity between the window's mean β and the estimated skin colour α is calculated. The two cylindrical HSV colour vectors are transformed to Euclidean space using the following formulae:

$$x = \cos(2I_0) \cdot I_1/255 \cdot I_2/255, \;\; y = \sin(2I_0) \cdot I_1/255 \cdot I_2/255, \;\; z = I_2/255 \quad (4.4)$$

The Euclidean distance d is then computed between the 3D colour points. If $d \leq 0.35$, we assume the window primarily contains skin pixels. The bottom of the clothing's neck, $Neck_y$, is located as the lowest 'skin window' within the aforementioned x and y intervals. Note that in the case that the subject is wearing clothing which is so similar in colour to their skin that the colour similarity distance remains below the threshold, we establish a cloth sampling window located around the x-coordinate of the face centre at the end of the y-interval. Otherwise, in the typical case, the cloth sampling window is located beneath the garment's neck at:

$$\begin{aligned} \texttt{sample}(I) = &\texttt{Rect}(\texttt{Point}(F_x + 0.25F_{width}, Neck_y + 1.5\gamma), \\ &\texttt{Point}(F_x + 0.75F_{width}, Neck_y + 1.5\gamma + 0.25F_{height})) \end{aligned} \quad (4.5)$$

where γ refers to the aforementioned window size of 5 pixels.

TABLE 4.1: Segmentation Parameters

Parameter	Segmentation Plane S	
	Hue I_0	Intensity I_2
q	16	15
λ	50	3

4.2.4 Chromatic vs Achromatic

We consider an approach to segment both chromatic (coloured) clothing and achromatic (black, white, or grey) clothing. The segmentation of achromatic clothing can pose a much greater challenge since it is more sensitive to the illumination present in the scene. We design a histogram based approach because this is very efficient and can have a high accuracy on segmenting clothing which is primarily monochromatic (i.e. plain and not multi-coloured). In such cases, it can also be suitable for semantic segmentation of printed/stitched textures within the clothing. First, we determine the chromatic ratio of the clothing which is estimated by taking the mean of the binary image $\texttt{chrome}(I)$ (see Equation 4.1) with the sampling ROI of Equation 4.5 applied:

$$\text{Chromatic Ratio} = r = \frac{1}{0.5F_{width} \cdot 0.25F_{height}} \sum_{x,y \in \texttt{sample}(I)} \texttt{chrome}(I(x,y)) \quad (4.6)$$

Second, the image plane for segmentation is determined based on whether the clothing is primarily achromatic or chromatic:

$$\text{Segmentation Plane} = S = \begin{cases} I_0 & \text{if } r \geq 0.5 \\ I_2 & \text{otherwise} \end{cases} \quad (4.7)$$

Based on these two cases, we empirically define some segmentation parameters in Table 4.1.

4.2.5 Clothing Segmentation

This section describes our histogram based segmentation routine. A histogram $\{g\}_{i=1\ldots q}$ is computed for image plane S with the ROI $\texttt{sample}(I)$ applied:

$$g_i = \sum_{x,y \in \texttt{sample}(I)} \delta[b(x,y) - i]. \tag{4.8}$$

where δ is the Dirac delta function and let $b\colon \mathbb{R}^2 \to \{1 \ldots q\}$ be the function which maps the pixel at location $S(x,y)$ to the histogram bin index $b(x,y)$. We empirically choose to quantize to q bins as this provides a good compromise between under-segmentation (due to variation in cloth hue/intensity caused by lighting) and over-segmentation (due to objects with similar hues/intensities which are in direct contact with the clothing). Quantization reduces the computational and space complexity for analysis, clustering similar color values together. The histogram is then normalized to the discrete range of image intensities:

$$h_i = \min\left(\frac{255}{\max(g)} \cdot g_i, 255\right), \forall i \in 1 \ldots q \tag{4.9}$$

where h is the normalized histogram, g is the initial histogram, and subscripts denote the bin index.

Image S is back-projected to associate the pixel values in the image with the value of the corresponding histogram bin, generating a probability distribution image P where the value of each pixel characterizes the likelihood of it belonging to the clothing (i.e. histogram h). The resulting probability image is thresholded to create a binary image:

$$P(x,y) = \begin{cases} 255 & \text{if } P(x,y) \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

Scene conditions such as illumination can alter the perceived hue/intensity of the cloth, so we empirically set the λ threshold relatively low (see Table 4.1).

We further constrain P by considering $\texttt{chrome}(I)$, the computed chromatic mask. If $S = I_0$, we let $P = P \wedge \texttt{chrome}(I)$. Otherwise, if $S = I_2$ and $r \leq 0.05$, we constrain with the achromatic mask, letting $P = P \wedge (255 - \texttt{chrome}(I))$.

(A) Detection (B) Chromatic mask (C) Backprojection

(D) Segmentation (E) Attributes

FIGURE 4.3: Our segmentation and parsing pipeline.

In the unlikely case that the sampled clothing pixels are mostly achromatic but not entirely (i.e. for $0.05 < r < 50$), we do not constrain P with the achromatic mask as it can exhibit significant holes at the location of coloured cloth pixels.

Morphological closing with a kernel size of 3×3 is employed to remove small holes, followed by opening, with the same kernel, to remove small objects. Experimentally, this has been found to fill small holes in the edges of the clothing caused by harsh lighting, and remove small objects of a similar hue/intensity to the cloth which are in contact with it from the camera's perspective.

Suzuki-Abe border tracing is employed to extract a set T of contours with corresponding tree hierarchy H. We choose to limit the hierarchy to 3 levels deep as this can provide sufficient information for the clothing contour, potential contours for a printed/stitched texture within, and potential holes within the texture contour(s). The top level of the hierarchy is iterated over, computing the bounding box area of each contour. The area is approximated to that of the bounding box for efficiency and experimentally this appears to be acceptable. We define the largest contour T_{max} as the clothing. Therefore, there is robustness to objects in the scene which have a similar hue/intensity as the clothing but are not in contact with it from the camera's perspective. An initial clothing segmentation mask $\tilde{M}_c$ can be defined by filling T_{max}.

The initial clothing segment $\tilde{M}_c$ can suffer in accuracy in cases of harsh illumination or patterned clothing. Robustness to these cases can be increased by sequentially performing morphological closing and opening with a large kernel size. The reason for not employing this larger kernel on the first iteration of morphological operations is that this can decrease accuracy if it is not just the clothing segment of interest present, but also other large objects of similar hue/intensity in the background. Since the morphological processes can create additional contours, border tracing is computed again to extract the clothing as one segment M_c.

4.2.6 Logo/Graphics Segmentation

Existing clothing segmentation methods do not purposely attempt to semantically segment printed/stitched textures within clothing masks. Segmentation of any potential printed designs on clothing can be used to make the clothing cue more informative. We hypothesize that this could be useful for the purpose of re-texturing in emerging augmented reality clothing applications.

We iterate through the contours T in the second level of the contour hierarchy H, computing their areas. Unlike the area computation for the cloth contour, we do not approximate by bounding boxes here because textures

can have more variation in shape, which may result in inaccurate area estimations. We consider contours with areas above a dynamic empirically defined threshold of $0.25F_{width}F_{height}$ to belong to a printed texture on the clothing. If no contour above the threshold is found, then we assume that there is no texture. Otherwise the extracted contours are filled and the regions of their corresponding hole contours in the third level of the hierarchy are subtracted (if they exist) from this result, yielding the texture mask M_t.

4.2.7 Clothing Attributes

Given a query image, the method has so far yielded per pixel labelling for clothing segmentation M_c, texture segmentation M_t, and background.

A final parsing stage is presented to assign clothing colour, neckline, sleeve length, and brand attributes as these attributes can be important for applications such as product retrieval (refer to chapter 3), subject identification (refer to chapter 6), and fashion analysis.

Color Name The English language contains eleven main colour terms: 'black', 'white', 'red', 'green', 'yellow', 'blue', 'brown', 'orange', 'pink', 'purple', and 'gray'. The estimated clothing colour attribute will be chosen from this set. For a background of these main universal colour terms that are used across many different cultures, the reader is referred to [137] and [138].

The mean clothing colour in Lab representation (L_c, a_c, b_c) is computed over the clothing segmentation M_c. In order to convert this numeric HSV representation of clothing colour to one that is more meaningful to humans, the popular CSS3 colour names by the World Wide Web Consortium [139] are considered. CSS3 is the latest evolution of the Cascading Style Sheets language, a language for describing the rendering of structured documents (such as HTML and XML) on various media such as screen or paper. CSS3 colour names are represented in a file which maps certain colour strings to RGB colour values. The mappings for the eleven main colours mentioned above are extracted from this file and converted to Lab

colur space, allowing the mean clothing colour value to be matched to the closest value in the list, giving a semantic attribute for clothing colour. The Lab colour space is used for the matching process since it is perceptually uniform, ensuring that the difference between two colors as perceived by the human eye is proportional to the Euclidian distance ΔE_{ab} within the color space. The distance between (L_c, a_c, b_c) and a colour (L_i, a_i, b_i) in the colour name set can be calculated according to the Lab *CIE76* formula:

$$\Delta E_{ab}(i) = \sqrt{(L_i - L_c)^2 + (a_i - a_c)^2 + (b_i - b_c)^2} \quad 0 \leq i \leq 10 \tag{4.11}$$

where Δ denotes difference, E is the German word for sensation, *Empfindung*, and $_{ab}$ identifies this as the *CIE76* formula. Minimizing the distance across the colour name set yields the semantic colour name of the clothing:

$$\hat{i} = \arg\min_i \Delta E_{ab}(i) \tag{4.12}$$

Neckline In order to predict the neckline of the garment, we make the assumption that the centre of the garment's neckline is located at the same x coordinate as the centre of the detected face. We find this assumption to be valid for the majority of cases where the subject of the photo is standing planar to the camera. Next, distances to the garment contour T are calculated to locate the closest point p on the contour that lies directly beneath this x coordinate. To calculate the gradient of one side of the neckline, we iterate up the garment contour from point p until the gradient approaches zero. The gradient is then extracted at 5 keypoints along this section of the contour. A simple sanity check is carried out on the estimated gradients of both sides, in case for example there is any major anomoly caused by incorrect estimation of p due to an unusual face position. The same procedure is repeated for the other side of the neckline, and the gradient key points are averaged.

Samples of images containing crew-necks and v-necks are used to calculate an average garment gradient vector for each respective neckline class. The neckline feature vectors are weighted to favour the elements closest to the centre point p. Finally, k-Nearest Neighbors is employed to classify the neckline.

Sleeve Length Related work tends to use pose estimation to locate the position of the arms and then applies a learning algorithm on the low level features of the arms. Without performing computationally intensive pose estimation and low level feature learning, this becomes an even more challenging task. To simplify the task, we assume that the arms usually appear in a specific area of the image (such as by the sides of the body). For each image in the training dataset, a bounding box is manually specified for each of the arms (from the shoulder to the hand) and the result is labelled as either 'sleeveless', 'short sleeves', or 'long sleeves'. From the manually specified arm bounding boxes in the training set and the ratio to their face bounding box dimensions, dynamic arm bounding boxes can be established in a new image. The amount of skin pixels in the mean of the arm bounding boxes in an image can then be classified based on learned thresholds.

Brand Logo recognition is an active area of research that has been receiving increased attention in the field of computer vision [140–152] as it can be used to benefit the recognition of products and organizations. There has been much focus on printed logo retrieval where the logo is in a simple image under ideal conditions and does not need to be located first [150]. Whereas, detecting logos in real world scenes has not received as much attention [140–149, 151–153] and is generally more challenging, especially as logos must be located in the photo prior to performing the recognition task. Various logo detection methods have been proposed for real world photos [141–149, 151, 153], videos [140, 152], and documents [154].

These approaches for real world photos can be grouped into the categories of query based [141, 142, 144–146, 153] and model based [143, 147–149, 151]. The main difference being that in a model based approach, a model is trained on a set of logos for each logo class, rather than just using one query logo for detection. Hence, the model based approach can be more accurate.

Logos on clothing exhibit non-rigid deformation and perspective tilt (based on camera and clothing positions) that cause tough challenges for detection. SIFT matching [155] is comonly used for logo detection, but by design it does not consider these challenges. Anti-distortion affine scale invariant

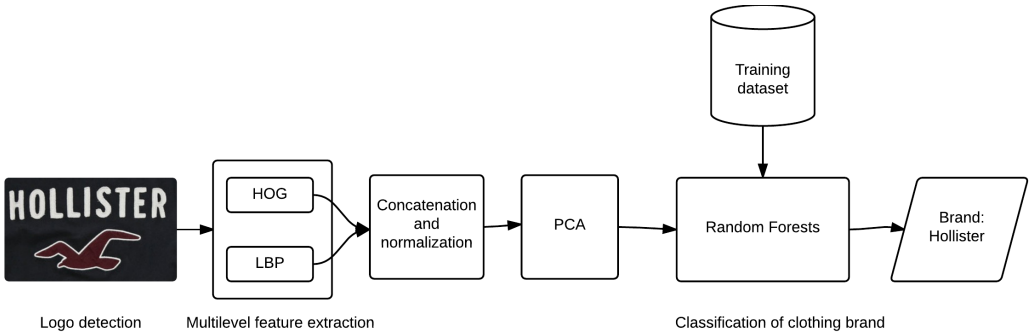


FIGURE 4.4: Clothing brand classification using random forests.

feature transform (ASIFT) matching [156] can be used to overcome these shortfalls. Both SIFT and ASIFT are computationally intensive, but ASIFT is even more so. Thus for offline high performance detection, ASIFT could be employed. However, the focus in this chapter is on highly efficient algorithms for mobile devices.

Therefore, this investigation proposes a model based approach by extracting multilevel Histograms of Oriented Gradients (HOG) and multilevel uniform Local Binary Pattern (LBP) on the detected clothing texture (if present). HOG and LBP are employed for their high efficiency and performance, as discussed in chapters 3 and 6. Principal Components Analysis (PCA) is then employed to further improve the performance by reducing the dimension of the multilevel HOG and LBP feature vectors to 64D.

Random forests (RF) are employed for the purpose of classifying the clothing brand based on the PCA feature vector for the texture M_t since they are efficient multi-class classifiers that can handle high dimensional features and are robust to noise and imbalanced datasets. A random forest classifier consists of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \ldots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$. A background on random forests is presented in § 2.5. Figure 4.4 depicts the clothing brand classification process.

4.3 Augmented Reality Framework

Figure 4.2 gives an overview of our clothing parsing and augmented reality pipeline. The augmented reality framework consists of the following steps:

1. Initialization of segmentation and reconstruction (refer to earlier in this chapter and § 5.1 respectively).
2. The mobile tablet device user downloads the image of a clothing product that they wish to try on from an ecommerce store. The product should be predominantly uniformly coloured and similar to the clothing they are wearing but in a different colour or art work variation. Hence, the method can work best for items such as plain/printed/embroidered T-shirts and tank-tops.
3. The dominant colour and any texture (such as graphic art work or logo) are extracted from the product image by using the simple method described in § 3.3. The product colour is then taken as the mean of the segmented product image and the product texture as the largest contours (if any) within the segmentated clothing contour. The product image is hereafter referred to as the virtual clothing.
4. The user's upper body clothing is automatically segmentated according to the method presented earlier in this chapter.
5. Point correspondences are tracked (refer to Chapter 5.1).
6. Sparse 3D points are reconstructed (refer to Chapter 5.1).
7. The rectangular cloth surface containing the clothing texture is geometrically reconstructed (refer to Chapter 5.2).
8. Illumination is recovered from the textured part of the cloth (described in this section).
9. The augmented scene is rendered and displayed (described in this chapter). The user can then choose to recolor and retexture the garment with different product designs.

4.3.1 Illumination Recovery

The application of illumination recovery within our framework is two-fold. Rather than simply overlaying a new colour or texture, we wish to remove the existing texture (if there is graphic art work or a logo present on the clothing) and then apply a new colour or texture of a potentially different shape/size. We recovered the dominant clothing color during the segmentation initialization stage, so this leaves the illumination to be recovered in order to reconstruct the appearance of parts of the cloth which are occluded by any real texture. Secondly, we modulate the alpha channel of the virtual texture that is discussed in chapter 5 with the recovered illumination to increase realism of the augmentation within the real scene. Given the segmentation mask of the real texture which we have obtained, we can treat the mask as unknown pixel values and interpolate to reconstruct the illumination values, yielding recovery of the cloth with any graphic art work or logos removed. Inpainting is an advanced interpolation scheme used to reconstruct unknown or lost parts of an image.

Traditionally, inpainting methods have focussed on recovering small cracks in paintings. Inpainting on large areas remains a challenging task, and even more so in real-time. Popular inpainting methods [166–168] are generally intended for offline use or for near-online use interpolating very small unknown areas. Although slow for online use, Telea's approach is significantly faster than the others mentioned.

We employ the fast inpainting approach presented by Telea. However, to support the online nature of our framework, we must inpaint at a reduced resolution based on the characteristic $\rho_r \propto 1/\rho_a$ where ρ_r is the resolution and ρ_a is the area of the texture mask. For the purpose of demonstration, we could limit results like previous work to textured clothing where the texture consists of small to medium contours. However, our focus is on a practical method with robustness to different clothing designs, and many printed T-shirts in our random sample of the population feature large area designs, so we employ a larger window for inpainting than previous work.

The columns in Figure 4.5 show the illumination channels before and after illumination recovery by inpainting. The first row shows the best case

where the high frequencies across the texture are smoothed out. The second row shows the worst case where there are significant folds present within the textured region. These folds exibit harsh intensity changes (high frequency data) which when inpainted, are overly-smoothed, reducing the realism of the folds. There is a compromise here because reducing the window size can improve some regions of this result, however a small window can also produce artifacts when interpolating across wide unknown regions. The retexturing results in Figure 5.8 show that when the texture is alpha-modulated by the recovered illumination, the artifacts around the folds are not noticeable and the lighting is convincing. Thus this is a reasonable result for interpolating across a large unknown area. Sharp texture edges are filtered out and although there are some noticeable artifacts inherent to interpolating large areas (especially in real-time), smooth intensity changes are generally preserved.

FIGURE 4.5: Illumination Recovery: columns show before and after whereas rows show best case and worst case respectively.

4.3.2 Rendering

OpenGL is employed to render the output according to the following algorithm:

1. A frame is captured from the integrated camera sensor of the tablet device.
2. The captured frame is rendered fullscreen in the mobile app, such that the user can see themselves in real-time like a mirror.
3. The virtual clothing colour fills the segmentation mask of the user's clothing.
4. If present, the virtual clothing texture (such as graphic art work or logo) is mapped to the recovered 3D surface mesh.
5. The segmentation mask of the user's clothing is applied to the recovered illumination (i.e. the interpolated illumination channel for an untextured garment) and used to modulate the alpha channel of the virtual clothing color and texture, hence increasing the realism of the augmentation within the real scene.

4.4 Experimental Results

In order to analyse the accuracy and robustness of our segmentation approach, we experiment with various printed T-shirts (different T-shirt colors and printed textures) worn on various subjects against various backgrounds. We study the quality of the clothing segmentation, robustness to noise, qualitative evaluation of predicted clothing attributes, and the computational timing. These results are reported in Table 4.2, alongside a comparison with the state of the art. Augmented reality results are mainly discussed in the next chapter, once the re-texturing method has been presented.

To compare the computational efficiency of our approach to the closest state of the art, we similarly consider static image regions for each person with resolution 200×300. Note that the timing result excludes the texture based clothing brand labelling stage in subsection 4.2.7 as it may be unrepresentative using random forests on a small dataset. Our approach, excluding pre-processing (Section 4.2.1), achieves on average 2ms per person using a 2.93GHz CPU core and also runs in real-time when tested on a

	Proposed Method	Yang and Yu [9]	Yamaguchi et al. [27]	Wang and Ai [45]	Gallagher and Chen [8]
Timing	2.0**ms pp.** 2.9GHz core	16.5ms pp. 3.2GHz core	20-40s pp	Offline	Offline
Seg. Accuracy	0.97 **F-score**	N/A	84.7% (full pose only)	92.8%	89.4%
Parses Logo	✓	✗	✗	✗	✗
Semantic Attributes	✓ **colour, brand, neckline, sleeve length**	✗	✓**clothing categories**	✗	✗

TABLE 4.2: Indicative only - different datasets. Legend: pp = per person.

Samsung Galaxy Tab S 10.5 tablet with a Exynos 5 Octa CPU (1.9Ghz Quadcore + 1.3 Ghz Quadcore). Thus our method is over 88% more efficient than results reported by [9] under similar conditions, with the exception that they employ a faster CPU core (3.16GHz). Our overall system, including pre-processing (face detection and simple denoising), is fast, achieving results at an average rate of 25fps (frames per second) for segmenting one person given an input resolution of 480×640 pixels. The equivalent computation time is dissected as 38ms pre-processing per image and 2ms clothing segmentation per person. Face detection is our biggest computational bottleneck, so for high resolutions, the input to face detection could be downscaled. The segmentation procedure could easily be parallelized for each face detected, if using a multi-core CPU and the average number of persons to segment justifies the threading overhead. The mobile version of the application has been implemented for Android using the Android SDK, NDK, and OpenCV library.

Segmentation accuracy is reported using the best F-score criterion: $F = 2RP/(P + R)$, where P and R are the precision and recall of pixels in the cloth segment relative to our manually segmented ground truth. We achieve an average F-score over the entire testing dataset of 0.97. Since the F-score reaches its best value at 1 and worst at 0, our approach shows good accuracy. Additionally, by visual inspection of Figure 4.7, we can see that

our approach can semantically segment clothing of persons in various difficult uncontrolled scenes with some robustness to minor occlusions (Figure 4.7c) and minor patterns (Figure 4.7b). Clothing segmentation literature tends to report accuracy with regards to applications (such as recognition or classification) rather than directly on segmentation. Although not directly comparable, the performance is higher than that reported in [45], using mostly images from the same dataset.

We also consider robustness to one of the most common forms of noise: additive white Gaussian noise. This is caused by random fluctuations in the pixels. Naturally, this could be easily filtered but our aim is to demonstrate robustness. If the input image is represented by I_{input}, and the Gaussian noise by Z, then we can model a noisy image by simply adding the noise: $I_{noisy} = I_{input} + Z$. Z consists of 3 planes which correspond to the RGB planes of I_{input}, and is drawn from a zero-mean normal distribution with standard deviation σ. We study the effects of noise on a randomly selected image of a single person. Figure 4.8a depicts a graph of accuracy (F-score) versus the noise standard deviation (255σ) which shows our approach can handle significant noise. Note that we multiply σ by 255 since we consider integer images, and there is no data point plotted for $\sigma = 0.9$ because face detection mistakenly detects two faces and thus there are two results. Noise can positively affect our segmentation, for example at $\sigma = 0.2$, if noise pixels with hues similar to the cloth are established in dark clothing regions which originally had many unstable hues. At $\sigma = 1.0$, depicted by Figure 4.8b, the face detection accuracy continues to decrease; however, the corresponding clothing segmentation in Figure 4.8c remains reasonably accurate. The segmentation fails entirely at $\sigma = 1.1$ since the prerequisite of face detection fails to detect any faces.

Preliminary results for clothing attribute labelling are promising, as seen in Figure 4.3. Furthermore, Figure 4.6 presents examples of logo classification for the *Hollister* class. We can see that when the usual style of Hollister bird logo or text is classified, it often results in a correct classification. However, when the graphics are in a more unusual style, the classification fails mainly due to the lack of training data for such a style. For future work, a larger clothing dataset can be collected for a comprehensive quantitative analysis of the semantic attributes generated by the

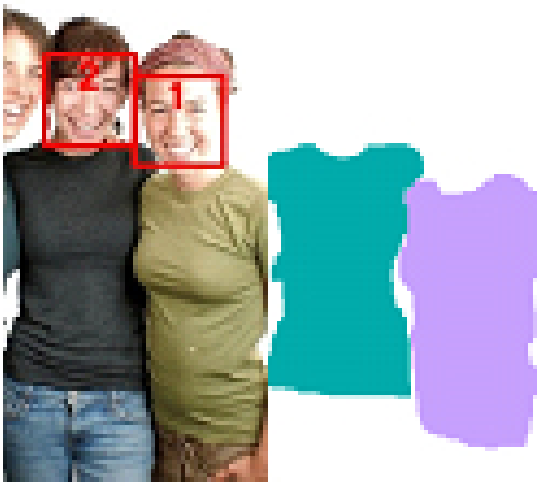
FIGURE 4.6: Examples of logo classification for the *Hollister* class.

heuristic and random forest classifiers.

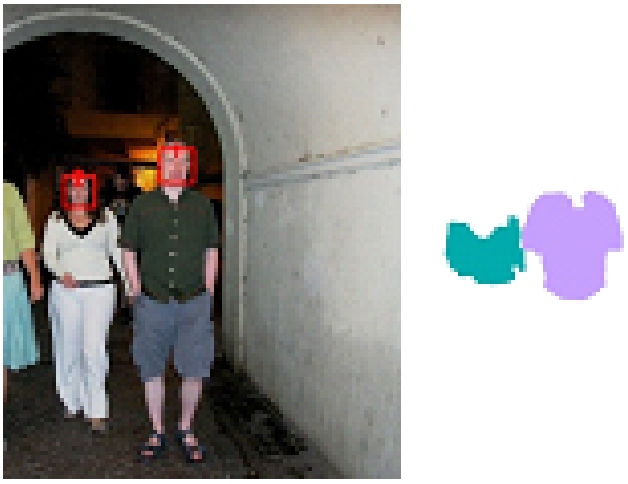
Our approach is subject to some limitations. We assume that there are no significant objects of a similar hue/intensity to the chromatic/achromatic clothing which are in direct contact with it from the camera's perspective, and the clothing is predominantly uniformly coloured (i.e. it is not significantly patterned). These limitations should not significantly affect the suggested computer games and broadcasting applications for the purpose of augmented reality.

4.5 Summary

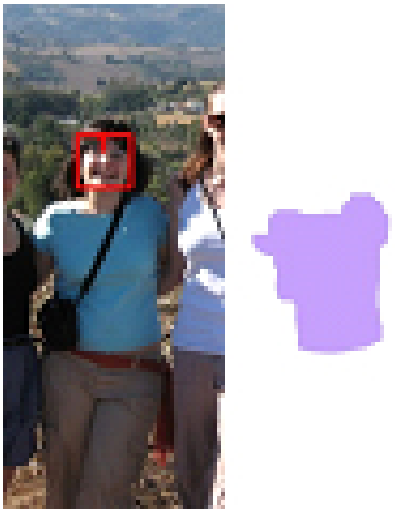
An approach has been presented for automatic semantic clothing segmentation of multiple persons. It does so by estimating whether the clothing is chromatic or achromatic and then applying a histogram based approach on the hues or intensities. In order to initialise points on the cloth, we have proposed a method consisting of skin colour estimation and colour similarity to locate the bottom of the garment's neck. A clothing attribute stage then predicts the clothing colour, brand, neckline, and sleeve length. Finally, we demonstrate the approach with an augmented reality mirror app for mobile tablet devices that can segment a user's clothing in real-time and enable them to realistically see themselves in the virtual mirror wearing variations of the clothing with different colours (or graphics rendered as per chapter 5). We have shown that the proposed framework is able to segment clothing more efficiently than existing state of the art methods, whilst achieving good accuracy and robustness on a difficult



(A)

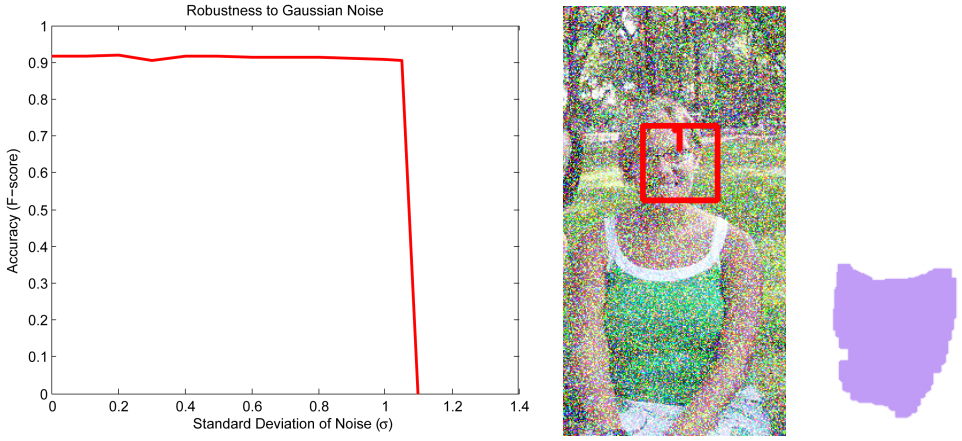


(B)



(C)

FIGURE 4.7: Further segmentation results. Each pair shows the numerically labelled person(s) on the left with their corresponding colour labelled clothing on the right.



(A) (B) (C)

FIGURE 4.8: Robustness to noise: (a) graph of accuracy versus Gaussian noise σ, (b) input with considerable noise ($\sigma = 1.0$), and (c) corresponding clothing segmentation.

dataset. Although our approach is limited to predominantly uniformly coloured clothing (which may contain textured regions), it can be of particular benefit to emerging real-time augmented reality applications such as augmented try on of clothing, sports broadcasting, and computer gaming.

The papers [21] and [22] resulted from the approach presented in this chapter. Even though research in the field of clothing segmentation and parsing has rapidly accelerated in the last few years, with papers such as [24, 25, 29] citing the above, the approaches in this chapter remain unique as they are specifically targeted towards practical real-time applications for mobile devices where very high efficiency is required. Furthermore, the approaches do not have constraints of related work such as requiring the full body of the subject to be visible in the image, manual interaction, or depth images.

Chapter 5

Augmented Reality Re-Texturing

In this chapter, we extend the efficient clothing parsing and augmented reality framework in the previous chapter with a dynamic multi-resolution approach for 3D shape reconstruction of highly deformable surfaces (such as cloth). This enables for clothing on a person to be re-textured with different graphic art work, logos, colours, or 3D special effects. This can be seen in Figure 5.1.

FIGURE 5.1: Demonstration of our augmented reality mirror: (a) input: a monocular video frame capture; (b) output: the clothing is automatically segmented, recoloured and retextured according to a re-projection of a texture-mapped three-dimensional reconstructed clothing surface.

Other applications of this work lie in TV broadcasting and advertising. Advertising space on a sports player's shirt could be dynamically changed and realistically re-textured during broadcast. If the broadcast was streamed online, highly targeted adverts could appear based on the consumer's habits and geographical location. Otherwise, if broadcast live on TV, the shirt could be retextured for different television markets. Existing work may potentially be able to partially accomplish this but is *not* very well suited for retexturing outside of the original textured region, or for high definition (HD) broadcasting.

We argue that a photorealistic and practical reconstruction of cloth geometry with retexturing is important for augmented reality applications and requires a method that:

- is robust to a variety of T-shirt colors and textures under non-uniform illumination against a variety of backgrounds.
- employs monocular vision. Consumer applications usually require a single camera as current smart phones, tablets and laptops commonly have one integrated webcam, but not yet stereo cameras or depth sensors. Additionally, considering the TV broadcasting application, few sports broardcasts on TV currently feature multiple-views of a subject which are in sync and fully calibrated together.
- reconstructs full 3D cloth geometry. The general increase in accuracy over 2D recovery can improve realism, which may become particularly noticeable at high definition (HD) resolutions. Secondly, this allows for the recovered surface to be viewed in 3D, which may have potential applications in gaming and entertainment. Furthermore, 3D recovery is required for the next point.
- exhibits the dynamic behavior of cloth.
- is very fast and suitable for real-time applications.

The tradeoffs and design decisions which we make are founded on satisfying *all* of these requirements. In addition to the aforementioned requirements, monocular vision is chosen to recover the 3D layout since this problem is severely under-constrained, being a very challenging open problem.

Also, by focusing on the challenging monocular case, our approach is generally significantly more applicable than approaches designed for good, less challenging, conditions such as muliple views.

The novel contribution presented in this chapter is a hierarchical multi-resolution method employing thin-plate splines, cloth modeling, and patch tessellation for detailed and dynamic 3D geometric reconstruction of highly deformable surfaces (such as cloth) from a set of sparse 3D points in real-time. A local thin-plate spline model recovers a continuous rectangular surface mesh from a limited set of tracked features on a partially textured cloth. A global cloth model then attempts to (a) increase the accuracy of this reconstruction at regions with less texture and fewer features such as the untextured cloth around arbitrarily shaped textures, and (b) restore cloth dynamics. Patch tessellation is employed to subdivide the mesh, increasing surface smoothness and improving realism at regions of high curvature.

5.1 Recovery of Sparse 3D Points

This section focuses on reconstructing sparse 3D points from highly non-rigid 3D surfaces, such as cloth, captured in monocular video (or alternatively, a single image and a texture template). This work is in preparation for the next section which describes our contribution for a hierarchical reconstruction of a continuous surface from these sparsely recovered 3D surface points.

Our reconstruction method requires geometric calibration of the camera. We use the popular chessboard technique implemented in the OpenCV library in order to calibrate the intrinsic and extrinsic camera parameters. Figure 5.2 shows the extrinsic calibration used to define world space.

Previous work has already been discussed in subsection 2.1.5. Tight clothing, such as *fitted T-shirts*, is characterized by locally near-rigid deformations which can be approximated by surface skinning techniques. We focus on the much more complex case of non-rigidly deforming cloth such as that represented by *standard T-shirts*. Cloth, such as 100% cotton T-shirts,

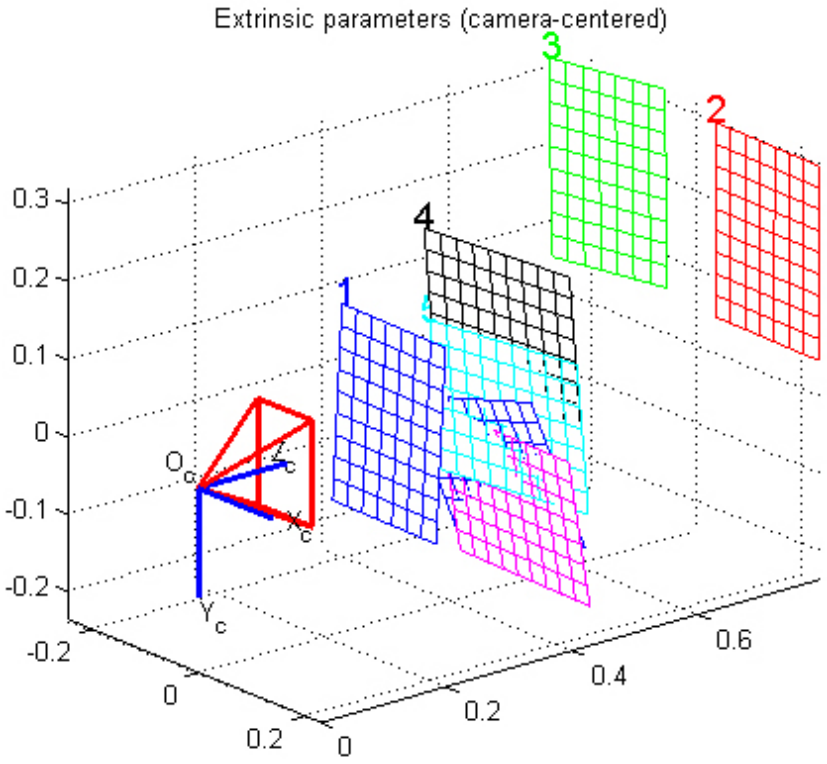


FIGURE 5.2: Webcam Calibration: the red pyramid represents the camera and its field of view whereas the colored grids represent the chessboard positions.

typically expresses some small elasticity. Reconstructing a texture in 3D from monocular vision which can be both loosely hanging and stretched is very challenging (and even more so in real-time) - this is a state-of-the-art topic with little research. We assume that the T-shirt elasticity is negligible and that the texture on the T-shirt is large and screen-printed which gives locally near-inelastic properties, so that we can constrain the reconstruction algorithm to inelastic materials.

For very fast recovery of sparse 3D points from tracked 2D point correspondences we use the approach presented by Perriollat et al. [90]. The planar texture template is deformed to the unknown 3D surface by an unknown isometric transformation. Due to the inextensibility constraint, the *geodesic curve*[1] between two points on the surface has the same length as the geodesic distance in the template, however, the surface deformation causes the Euclidean distance between the 3D points to be less. To find

[1]the shortest path along the surface between two points in a curved space

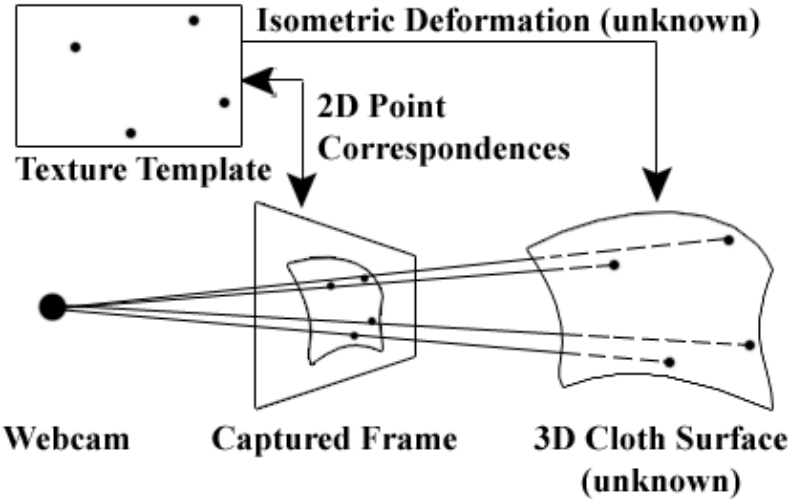


FIGURE 5.3: Cloth layout recovery.

depths of the 2D points, their method computes bounds in a pairwise fashion where two points along with the inextensibility constraint bound the position of these points along their sightlines which intersect at the center of the projective camera. The overall set of upper bounds are iteratively refined to find an optimal solution.

Figure 5.3 shows an overview of the approach. First we describe computation of 2D point correspondences and then the approach based on Perriollat et al. for initialization and refinement of the depth bounds is briefly described.

5.1.1 2D Point Correspondences

The first step involves establishing 2D point correspondences between the deforming surface in the captured image frame and the template. SIFT is often used for this purpose and is employed in [90], however it is not suitable for real-time. We employ simple real-time registration. Since we derive the template from the image sequence itself, there is a known mapping between these images in world space. Robust feature points are established at regions of high curvature on the texture. These points are then tracked over the video sequence using sparse optical flow [159]. However, this optical flow algorithm traditionally finds pixel displacements in

the sub-pixel range, and we require large displacements since we are capturing human movement from a webcam. Therefore, a multi-resolution scheme on a Gaussian image pyramid is integrated to handle larger pixel displacements. Also, to handle occasional loss of features, we use temporal coherence to attempt to re-establish them. This simple method has limitations in that it does not regularize the optical flow field and does not support self-occlusion handling, so it is best suited to short videos. We find this to be an acceptable assumption as a user is unlikely to want to spend a long time trying on one item of clothing in our augmented reality application.

5.1.2 Initializing the Bounds

To find depths of the 2D points, distance bounds are computed in a pairwise fashion where two points along with the inextensibility constraint bound the position of these points along their sightlines which intersect at the center of the projective camera. For n correspondences, there are $n-1$ bounds for each one, but only the most restrictive bound is kept.

The depth μ_i of a point can be converted to a 3D point Q_i on a sightline S_i from the camera center C with the following equation:

$$Q_i(\mu_i) = \mu_i \cdot \mathbf{r}_i + C \tag{5.1}$$

where the camera center $C = -R^{-1}\mathbf{t}$. R is the (3×3) rotation matrix, and $\mathbf{t}$ is the (3×1) translation vector. Both R and $\mathbf{t}$ are derived from camera calibration (refer to § 5.1). Note that the camera matrix P is given by concatenating the rotation matrix with the translation vector, i.e. $P = [R|\mathbf{t}]$. Finally, the direction $\mathbf{r}_i$ of the sightline S_i passing through the frame's image point I_i is given by:

$$\mathbf{r}_i = \frac{R^{-1} \cdot I_i}{\parallel R^{-1} \cdot I_i \parallel} \tag{5.2}$$

The coordinate frame system is chosen for the pair of points under evaluation as:

$$Q_i = \begin{pmatrix} \mu_i \\ 0 \end{pmatrix} \qquad Q_j = \begin{pmatrix} \mu_j \cos(\alpha_{ij}) \\ \mu_j \sin(\alpha_{ij}) \end{pmatrix}$$

where α_{ij} is the angle between the two sightlines S_i and S_j.

Given the depth of the point indexed at i as μ_i, and the distance between points Q_i and Q_j as $d_{ij} = \|Q_i - Q_j\|$, we can find the depth of the point indexed at j:

$$\mu_j(\mu_i) = \mu_i \cos(\alpha_{ij}) \pm \sqrt{d_{ij}^2 - \mu_i^2 \sin^2(\alpha_{ij})} \tag{5.3}$$

so long as:

$$\mu_i \leq \sqrt{\frac{d_{ij}^2}{\sin^2(\alpha_{ij})}}$$

The upper depth bound $\breve{\mu}_i$ is computed from the entire set of n tracked 2D point correspondences (assuming a common camera lens which gives the property $\alpha_{ij} \leq \pi/2$):

$$\breve{\mu}_i = \breve{\mu}_{ii^\star} = \min_{\substack{j=1..n \\ j \neq i}} \left(\frac{d_{ij}}{\sin(\alpha_{ij})} \right) \tag{5.4}$$

The point which achieves the minimum upper bound has index $i^\star$. The notation $i \rightarrow i^\star$ is used to state the property that point $i^\star$ constraints the upper bound of point i. Figure 5.4 depicts initialization of the bounds.

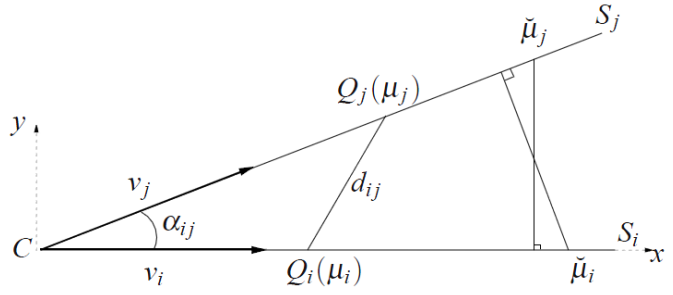


FIGURE 5.4: The upper bounds on the two sightlines S_i and S_j.

5.1.3 Refining the Bounds

The method described in the previous section is suboptimal. This is partly because the property $i \rightarrow i^\star$ is unsymmetric: the reverse is not true. Therefore, an iterative refinement technique for the upper bounds is described.

We recall Equation 5.3 and focus on the upper bound on point j induced by point i:

$$\mu_j(\mu_i) = \mu_i \cos(\alpha_{ij}) + \sqrt{d_{ij}^2 - \mu_i^2 \sin^2(\alpha_{ij})}, \tag{5.5}$$

which is given by the global maximum of the function μ_j at:

$$\mu_i^{\max} = \frac{d_{ij}}{\tan(\alpha_{ij})} \qquad \mu_j(\mu_i^{\max}) = \frac{d_{ij}}{\sin(\alpha_{ij})}. \tag{5.6}$$

Figure 5.5 shows a representation of the function.

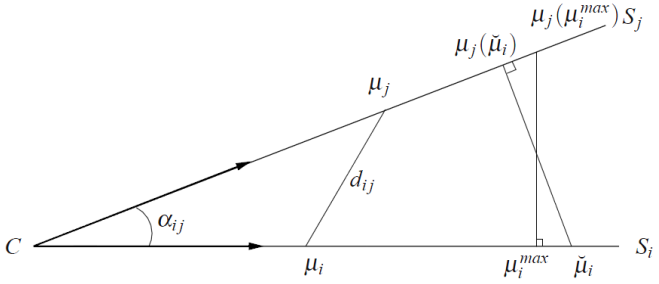


FIGURE 5.5: Parameterization of the function giving the depth of point j against the depth of point i.

Therefore, the upper bound for point j with respect to point i is given by:

$$\breve{\mu}_{ji} = \begin{cases} \mu_i \cos(\alpha_{ij}) + \sqrt{d_{ij}^2 - \mu_i^2 \sin^2(\alpha_{ij})} & \text{if } \breve{\mu}_i \leq \frac{d_{ij}}{\tan(\alpha_{ij})} \\ \frac{d_{ij}}{\sin(\alpha_{ij})} & \text{otherwise} \end{cases} \tag{5.7}$$

The upper bound for point j is then updated by a minimization:

$$\breve{\mu}_j = \min(\breve{\mu}_{jj^\star}, \breve{\mu}_{ji}) \tag{5.8}$$

In the next chapter, we use the upper depth bounds $\breve{\mu}_i$ as the reconstructed depths of the points on the surface $\tilde{\mu}_i$.

$$\tilde{\mu}_i = \breve{\mu}_i \tag{5.9}$$

5.2 Recovery of 3D Cloth Surface

In this chapter, we propose a method to reconstruct a continuous 3D cloth surface from a limited set of recovered sparse 3D points. Our approach

has the following main benefits:

- **a hierarchical scheme for recovery of a rectangular surface from an arbitrarily shaped texture** where a local thin-plate spline model recovers a continuous rectangular surface mesh from a limited set of tracked features on a partially textured cloth. A global cloth model then attempts to increase the accuracy of this reconstruction at regions with less texture and fewer features such as the untextured cloth around arbitrarily shaped textures.
- **a multiresolution scheme** where patch tesselation is employed to subdivide the recovered surface mesh, increasing surface smoothness and improving realism at regions of high curvature. These effects are particularly noticeable at high resolutions and thus our reconstruction is made more suitable for high definition (HD) applications.
- **a dynamic reconstruction** which attempts to retain the dynamic behavior of the worn garment by means of the cloth model in the hierarchical scheme. This could have potential uses for novel future applications which will be briefly discussed.

5.2.1 Local Model

The set of maximal depth bounds $\{\breve{\mu}_i\}_{i=1\ldots n}$ computed in the previous section was shown by Perriollat et al. [90] to exhibit a small error when used as the reconstructed depths $\tilde{\mu}_i = \breve{\mu}_i$. We utilize an optimization similar to theirs which attempts to solve this minor problem by satisfying the inextensibility constraint (mentioned earlier), i.e. the Euclidean distance between recovered 3D points $\parallel Q_i - Q_{i^\star} \parallel$ equals their distance in the template $d_{ii^\star}$. A temporal smoother is simultaneously introduced to help stabilize recovered points by penalizing larger movements between frames. The optimization equation is defined as:

$$\tilde{\mu} = \arg\min_{\mu} \sum_{i=1}^{n} (\breve{\mu}_i - \mu_i)^2 + \gamma \left(\mu_i(t) - \mu_i(t-1)\right)^2 \quad \text{subject to } \parallel Q_i - Q_{i^\star} \parallel = d_{ii^\star} \tag{5.10}$$

where $\tilde{\mu}$ is the recovered depth, μ_i the depth of point i, $\breve{\mu}_i$ the upper bound, n is the point correspondence count, t is the current frame, and operator selected weight γ. Equation 5.10 is a linear least squares problem under non-linear constaints which can be solved with the Levenberg-Marquardt algorithm [160]. Unlike the previous work which is implemented in Matlab, the GaussFit[2] implementation of this algorithm in the C language is employed as a fast solver.

A regular local mesh $\tilde{\xi}_l$ is established on the template image with vertices $\{v_i\}_{i=1\ldots s}$. Although the operator can choose the number of vertices, we empirically define an $s = 20 \times 30$ vertex mesh as this approximates the ratio of many T-shirt logos in our sample whilst providing a compromise between local surface resolution and computation time.

A mapping function can be fitted between the feature points in the 2D template image $\mathsf{F} = \{f_i\}_{i=1\ldots n}$ and the recovered sparse 3D points $\mathsf{Q} = \{q_i\}_{i=1\ldots n}$. Thin-plate splines (TPS) are employed for this purpose because they are popular for representing deformable objects and do so in an efficient manor [161]. The TPS is an $\mathbb{R}^2 \rightarrow \mathbb{R}$ function which is controlled by assigning target values to 2D source points whilst enforcing several conditions [161, 162]. The TPS can be considered to be the Radial Basis Function (RBF) that minimizes the integral bending energy. Our $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ TPS warp S is obtained by effectively stacking three TPS functions and sharing their centres. In terms of matrix manipulation, this involves deriving a parameterization matrix *fdp* from the feature points F, computing its transform, and multiplying by the recovered 3D points Q:

$$\mathsf{S} = \mathsf{Q} \cdot \mathrm{fdp}(\mathsf{F})^{\mathrm{T}} \tag{5.11}$$

In order to compute the deformed 3D surface, we must first kernalize the source points F with the initialized local mesh vertices $\tilde{\xi}_l$:

$$\mathsf{K} = \mathrm{kernelize}(\mathsf{F}, \tilde{\xi}_l) \tag{5.12}$$

[2] http://clyde.as.utexas.edu/Software.html

We implement the feature-driven parameterization function *fdp* and the *kernelize* function to kernalize the local mesh in C++ based on the *generalized thin-plate spline* algorithm proposed by [163]. The generalized thin-plate splines are chosen for their feature-driven parameterization which is concise, efficient, and meaningful, having the advantages over the standard TPS that it separates variables and introduces units.

Finally, the positions of vertices in the deformed local mesh are given by:

$$\xi_l = \mathsf{SK} \tag{5.13}$$

where the columns of ξ_l give the x, y, and z coordinates of each vertex in the deformed local mesh.

5.2.2 Global Model

By design, the surface recovered by the local model is most accurate at regions on the texture where many features are present. Geometric surface errors can be caused by interpolation across relatively untextured regions with few features and extrapolation outside of an arbitrary shaped texture. We propose to alleviate these errors by introducing a global model to infer some knowledge of the worn garment. As we focus on recovering a cloth surface and our demonstration is in respect to T-shirts, we choose to employ a cloth model for this purpose. Additionally, a cloth model gives us a *dynamic* surface recovery which can attempt to retain the real cloth behavior. This could have potential uses for novel future applications such as (a) temporal interpolation: cloth animation inbetween slow webcam frame captures (like an analogy to frame interpolation in high-end TVs); (b) video effects: cloth is animated when pausing the video; and (c) augmented reality gaming: the game is played on the T-shirt and the surface can be realistically changed based on game events.

Previous work has been described in Section 2.1.5. We design our global model around the mass-spring approach since it is popular, efficient, and can achieve an accurate simulation of cloth [98]. Our framework is perhaps most similar to the work of Muller et al. [164]. Figure 5.6 depicts an

example 2×2 mass-spring network where masses are arranged in a rectangular grid and connected to their neighbors by springs. The vertical and horizontal springs constrain cloth stretching and compression whereas the diagonal springs constrain bending. We define five core simulation constraints for cloth stretching, bending, self collisions, edges, and deforming key vertices to the local recovery mesh. We present each constraint $E_{...}$ as a function with respect to a subset of vertex positions along with a weight parameter $k_{...}$. Vertex positions are computed at least once every frame capture based on these constraints.

The procedure for initializing the global cloth mesh ξ_g is now described. We start by finding the approximate y coordinates of the neck and bottom of the T-shirt by minimizing and maximising y across a range of x coordinates at the center of the frame's cloth segmentation mask. Ideally, the user holds their arms out sideways for this frame, as this pose has been found to capture most of the front face of the cloth. Then we iterate upwards from the bottom of the shirt, minimizing and maximising x. We stop and define the bottom of each of the sleeve intersections once the change in x on each side of the shirt reaches a predefined threshold and continues to do so for at least 3 more y iterations, thus adding robustness to noise. The mesh edges are first fitted to the rectangle obtained from the x and y optimizations. The inner topology of ξ_g is structured so that it has a vertex aligned with each of the vertices in the initialized ξ_l. The remaining inner vertices are established horizontally and vertically in proportion according to the ξ_l vertices/texture-width and vertices/texture-height ratios, respectively. Vertices on the mesh edges below the sleeve intersections are then refined to the edges of the segmentation mask. The x values for vertices on the left and right sides above the intersections are set to the x coordinate of each intersection respectively. We let $m\colon i_l \to i_g$ be a function which maps vertex indicies $i_l \in \xi_l$ to $i_g \in \xi_g$. We initialize $\{h_i = 0\}_{i \in \xi_g}$ and then iterate over all possible inputs of the function m to effectively store all existing unique reverse mappings in h.

After initialization, the sleeve intersections continue to be computed every frame for defining the edge constraint E_{edge}.

Since we assume cloth elasticity to be negligible, we propose penalising compression and stretching of vertices which are horizontally or vertically

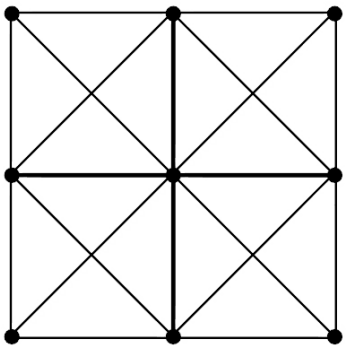


FIGURE 5.6: Our mass-spring cloth model.

adjacent. Therefore, the stretching constraint is evaluated for horizontally or vertically adjacent vertices $\mathbf{v_i}$ and $\mathbf{v_j}$ as:

$$E_{stretch} = \sum_{i,j} \left(\frac{\parallel \mathbf{v_i} - \mathbf{v_j} \parallel - l_{i,j}}{l_{i,j}} \right)^2 \tag{5.14}$$

where $l_{i,j}$ is the rest length of the edge between $\mathbf{v_i}$ and $\mathbf{v_j}$ in the initial state of the mesh. The bending constraint takes the form:

$$E_{bend} = \sum_{a,b} l_{a,b} \cdot (\mathbf{n_a} - \mathbf{n_b})^2 \tag{5.15}$$

where $\mathbf{n_a}$ and $\mathbf{n_b}$ are the normals of adjacent triangles a and b within the triangulated mesh, and $l_{a,b}$ is the length of the common edge between them. The anchor constraint deforms the key anchor points in the global mesh to their corresponding points in the local mesh:

$$E_{anchor} = \sum_{i \in \xi_l} \|\tilde{\mathbf{v}}_\mathbf{i} - \mathbf{v}_{\mathbf{m(i)}}\|^2 \tag{5.16}$$

where $\tilde{\mathbf{v}}_\mathbf{i}$ are the vertices in the local mesh ξ_l and $\mathbf{v}_{\mathbf{m(i)}}$ are the associated vertices in ξ_g. The following constraints are combined with their corresponding weights into one energy function which can be minimized with the Newton-Raphson method:

$$E = k_{stretch} \cdot E_{stretch} + k_{bend} \cdot E_{bend} + k_{anchor} \cdot E_{anchor} \tag{5.17}$$

We choose the Newton-Raphson method because it is popular and its stability does not depend on the size of the time step (which is important when considering a webcam as the frame capture device) but on the order

and shape of the constraint functions.

Handling of cloth self-collisions is carried out separately with spatial hashing [165] according to:

$$E_{selfcol}(\mathbf{v_j}, \mathbf{v_i}, \mathbf{v_{i+1}}, \mathbf{v_{i+2}}) = ((\mathbf{v_j} - \mathbf{v_i}) \cdot \mathbf{n_l}) - h \tag{5.18}$$

ensuring that point v_j remains above the triangle face $(\mathbf{v_i}, \mathbf{v_{i+1}}, \mathbf{v_{i+2}})$ with normal $\mathbf{n_l}$ by the cloth thickness $h = 1$.

There is one external force, gravity $\mathbf{f_g}$, which acts on the vertices. The three parameters for the cloth model comprise of $k_{stretch}$, k_{bend}, and k_{anchor}. The values of $k_{stretch}$, k_{bend} are constant and are discussed later on. The parameter k_{anchor} is effectively a constant and is defined dynamically after feature extraction according to:

$$k_{anchor} = \begin{cases} \rho_{h_i} & \text{if } h_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.19}$$

Each weight ρ_i is characterized by the proximity of point correspondences in the local neighborhood $\aleph_i$ of the corresponding vertex $\tilde{v}_i$ for the initial state of mesh ξ_l in the template where $\rho_i \propto \aleph$.

5.2.3 Surface Smoothing

We discovered experimentally that the global surface ξ_g can appear to have coarse jagged edges at regions of high curvature. A simple way to help alleviate this problem is to increase the resolution of the local and global recovery meshes. However, we introduce a multi-resolution scheme based on GPU patch tessellation. This improves the detail and smoothness of the rendered cloth at GPU level from the coarser mesh at CPU level. Thus the speed of our reconstructions remains fast.

Detail is added using a tessellation patch which has the same shape as a single triangle of ξ_g but it is subdivided into a specified triangle count to determine the increased resolution. We empirically choose a patch tesselation of 5 subdivisions as a compromise between coarse jagged edges and over-smoothing the surface.

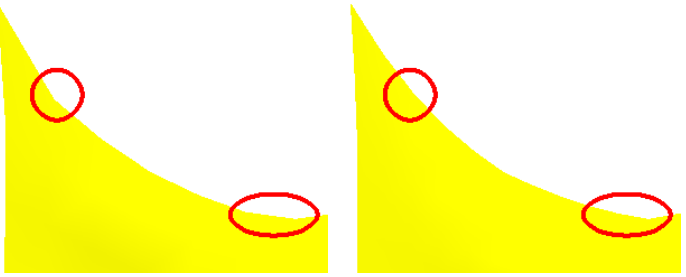


FIGURE 5.7: Surface smoothing by patch tessellation: (a) before, and (b) after.

The region of the global surface ξ_g which corresponds to the local surface ξ_l is used for our reconstructions.

5.3 Experimental Results

Convincing results for deformable surface reconstruction from monocular vision and re-texturing/re-coloring are presented in Figure 5.8. The columns show an arbitrary section of a captured video sequence which exibits cloth deformation. The rows show tracking of 2D point correspondences between a template derived from the start of the video sequence (initialization step) and the respective frames; sparse 3D point recovery (depth view); a mesh and retextured view of our continuous 3D surface recovery; and rendering the augmented reality scene (which is discussed in the previous chapter). Note that due to the way the camera has been calibrated, the more negative the points are along the *z-axis*, the closer they are to the camera.

Figure 5.7 shows the results of surface smoothing by patch tessellation. We can clearly see that the patch tessellated surface is of a higher resolution to the surface constructed from triangle patches. The increase in detail has successfully alleviated the coarse jagged edges which were noticeable at regions of high curvature.

Table 5.1 lists the core model parameters which we use for 3D surface recovery. For the purpose of our demonstration, we empirically define the cloth parameters k_{bend} and $k_{stretch}$ to attempt to mimic behavior of 100%

Parameter	Value	Parameter	Value
$k_{stretch}$	1.0	γ	10^6
k_{bend}	0.7		

TABLE 5.1: Model Parameters

cotton T-shirt cloth. We are not aware of a real-time technique for cloth parameter optimization, so our approach here avoids offline setup, keeping our focus on a fully real-time method with minimal user interaction. Although mass spring models are highly popular for simulating deformable objects such as cloth due to their real-time appeal, conceptual simplicity, and reasonable results, there is no ideal method and only methods which have been met with limited success [79] for obtaining optimal model parameters. This is because there is no obvious relationship between elastic material constitutive laws and model parameters.

We empirically set the weight γ to a very small value as a compromise between regularisation and vertex movement over time. We wish to regularise the model with respect to the temporal domain, as this can help remove anomalies. However, we consider large timesteps from webcam or mobile tablet frame capture and potentially large movement from a person within the scene, so we wish to encourage moderate movement of vertices in ξ_l over time.

We now consider the speed of our framework. Frame capture rate is dependent on factors such as CPU load and lighting conditions. Under our re-texturing testing environment with a Microsoft LifeCam Cinema webcam at a resolution of 480 by 640 there is an average frame capture rate of 10 fps. Our method achieves an average frame rate of 12 fps (including rendering) on an Intel Core i7 CPU (4 cores at 2.93GHz). Thus if we only process new frames, the output fps matches the input fps on average. Our C++ implementation is only partially optimized with a parallel architecture, so there is room for further performance enhancement. Also, in the previous chapter, we successfully demonstrated that the re-colouring part of the method performs in real-time on a recent tablet device.

Applications Our method for cloth reconstruction and augmentation can be used for emerging applications such as:

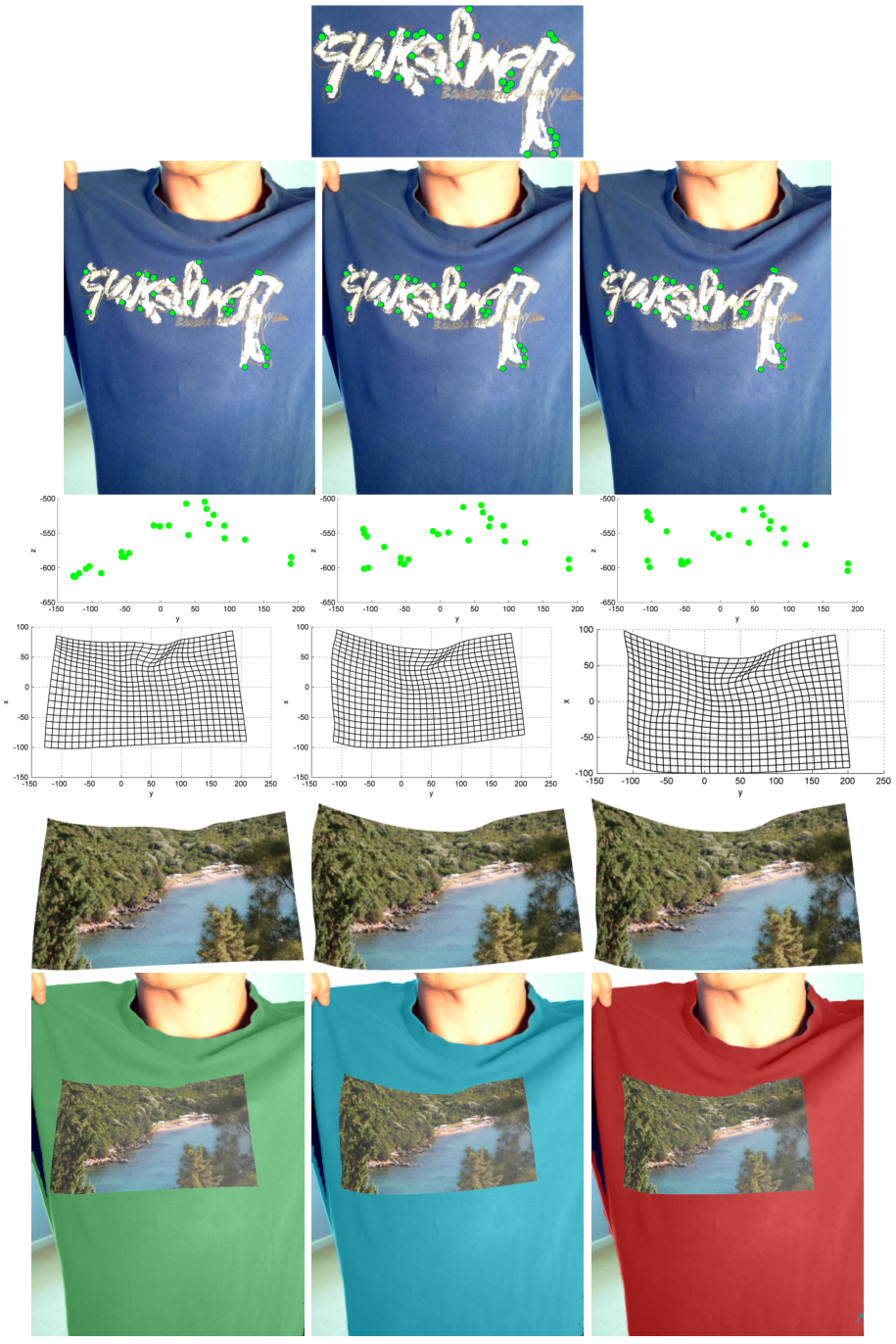


FIGURE 5.8: Results for 3D Cloth Recovery and Retexturing.

- Augmented try on: consumers could try on different clothing items either remotely via an app or using the technology in-store to limit or avoid the time required to find clothing of interest on the shelves and try-on clothing in the fitting room. This is the application demonstrated in the dissertation.
- TV broadcasting and advertising: advertising space on a sports player's shirt could be dynamically changed and realistically retextured during broadcast. Our detailed multi-resolution method is particularly well suited for high definition (HD) broadcasting. If the broadcast was streamed online, highly targeted adverts could appear based on the consumer's habits and geographical location. Otherwise, if broadcast live on TV, the shirt could be retextured for different television markets.
- Medicine: augmentations could be rendered on to skin to aid surgery and for teaching purposes. Although, a simple texture would be required to be drawn on to the skin as a prerequisite.

5.4 Summary

We have presented a hierarchical approach for 3D geometric reconstruction of highly deformable surfaces, such as cloth, which is robust to partially untextured regions given consecutive monocular video frames or a single image and a texture template. A retexturing and recolouring framework has been demonstrated for combining these two methods for the purpose of mobile augmented reality clothing try on. Robustness has been shown to relatively large timesteps (i.e. using a webcam or mobile device) under uncontrolled domestic lighting with variation in T-shirt shape and color, print shape and color, subject, and background. Our results show convincing 3D shape reconstruction and photorealistic retexturing whilst employing a setup which is practical for a consumer. The paper [21] resulted from this research.

The main limitation of our work perhaps lies in our surface reconstruction, in particularly the global model. The model assumes the resting state of

the front face of the upper body clothing to be planar (i.e. a perfectly flat chest). This creates an opportunity for extending the model to attempt to increase accuracy, such as by modelling the underlying body. Also, quantitative reconstruction analysis has not been carried out at this stage. Therefore, the main avenue for future work is to setup a depth sensor such as Microsoft Kinect to capture the ground truth clothing surface for a quantitative evaluation against the reconstructed surface.

Chapter 6

Person Identification

6.1 Introduction

Person re-identification is a critical security task for recognising a person across spatially disjoint sensors. Besides the rapidly growing number of published papers on person re-identification, the importance of this field is recognised in the recent survey by Vezzani et al. [54], published in a book on 'Person Re-Identification' Gong et al. [55]. The identification problem may be classed as either *single shot* or *multi-shot*. Single shot [170–172] approaches are only able to utilize one image of each person whereas multi-shot [173, 174] approaches can exploit multiple images of each person. The related work generally establishes either new feature representations [172, 173] or discriminative matching models [170, 171]. The closest work to ours is perhaps that of [9] and [175]. The approach of [9] is limited to tagging 8 attributes on a simple dataset in a standard lab environment. Whereas Layne et al. [175] discuss the new challenges associated with mobile re-identification. Their work is the state of the art and has only recently been published at the time of writing.

Person identification has been an active area of research for the past decade, but has almost exclusively focused on popular 2D RGB image data captured from fixed positions. This is a logical place to start since many venues already have a large network of traditional surveillance cameras which produce RGB data. Recently, more advanced sensor types such as

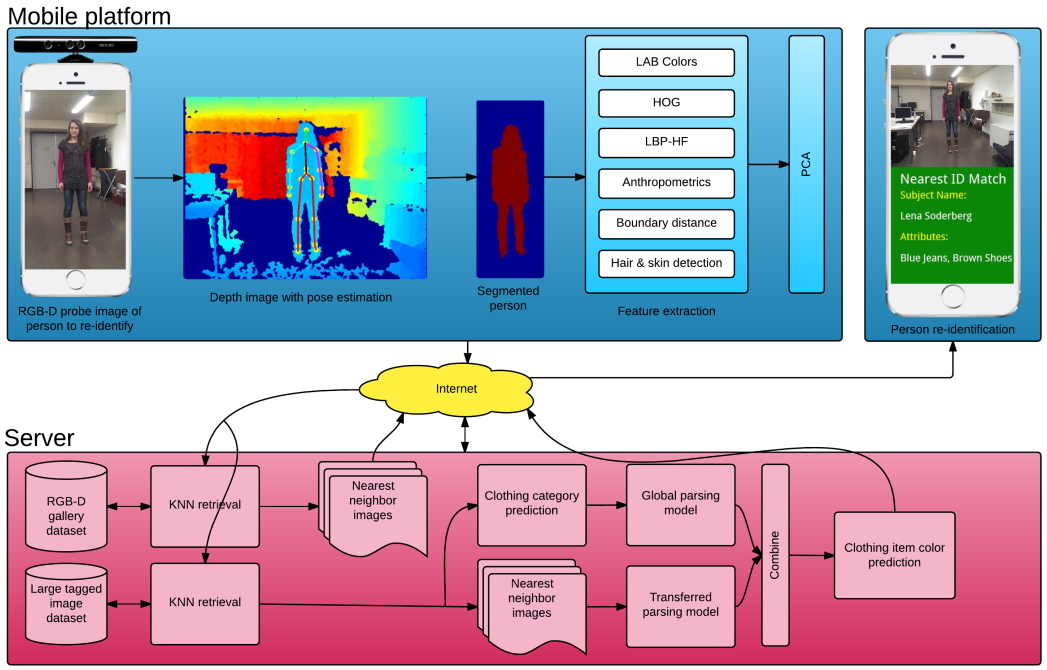


FIGURE 6.1: Overview of our mobile re-identification pipeline.

the Kinect have become very popular and available at a low cost. Smart mobile devices have also become very popular and have been considered in related work for clothing retrieval [4]. Gartner is predicting 1.37 billion smart phone sales and 320 million tablet sales for 2015. In terms of mobile OS, Android is expected to lead with 53% of the market [20]. The diversity of mobile platforms with integrated sensors is growing with new technologies such as wearable devices (Google Glass), intelligent robots, and remotely operated vehicles. Given these observations, we believe it is time to extend the literature to new up-and-coming scenarios. This is the focus of this work, where we present a novel semantic approach that integrates RGB and depth data to extract clothing and skeletal soft biometrics in a re-identification system for mobile devices. An overview of the system is depicted in Figure 6.1. Since mobile devices have very limited computing resources available, much attention is given to efficiency unlike most related work which is computationally intensive and runs on powerful workstations. This mobile approach may be particularly useful for the identification of persons in areas ill-served by fixed sensors or for tasks where the sensor position and direction need to dynamically adapt to a target. Furthermore, we contribute semantic ground truth clothing labels for the *BIWI* dataset to enable evaluation of predicted clothing items.

6.2 Datasets

Most RGB-D datasets of people are targeted towards activity and gesture recognition. Very few RGB-D datasets exist for the purpose of person re-identification. In [65], the first dataset explicitly for the purpose of RGB-D re-identification is created but there are few frames available per subject and the faces are blurred for privacy. We consider the state of the art *BIWI* dataset [176] which overcomes these limitations.

The BIWI dataset is targeted to long-term people re-identification from RGB-D cameras. It contains 50 training and 56 testing sequences of 50 different people captured with a Microsoft Kinect for Windows at approximately 10fps. 28 of the people present in the training set have also been recorded in two testing videos each: *still* and *walking*. These were collected on a different day and in a different location with respect to the training dataset, so most people are dressed differently. Thus, this dataset can provide a challenge for re-identification.

For training the semantic clothing prediction, the annotated subset of the *Fashionista* dataset [5] and the *Paper Doll* dataset [48] are utilized since they contain a significantly more diverse range of clothing than that present in the BIWI training dataset so can be applicable to many scenarios. The Fashionista dataset consists of 685 real-world photos from a popular online fashion social network, chictopia.com. For each of these photos, there are ground truth annotations consisting of 53 different clothing labels, plus hair, skin, and null labels. Whereas the Paper Doll dataset is a large collection of over $300,000$ fashion photos that are weakly annotated with clothing items, also collected from chictopia.com.

In order to evaluate predicted clothing items worn by a subject in a given test sequence, ground truth clothing labels are required. As these are not available in the BIWI dataset, we contribute semantic ground truth clothing labels which will be published online (see URL in section 6.7). First, five crowd-sourcing users are chosen to manually identify the clothing present. They are shown each image sequence in the BIWI dataset and given a choice of clothing attributes from the Fashionista dataset to tag. Three or more votes are required for a tag to become associated with the

sequence. To ensure high quality annotations, the received annotations are verified.

6.3 Mobile Re-Identification

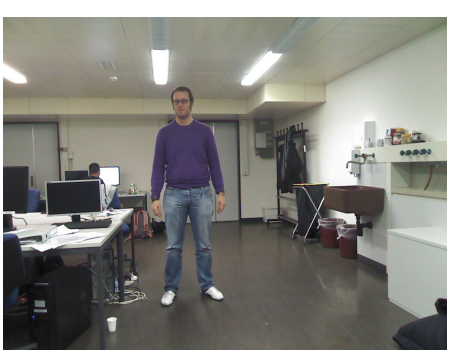
Our client-server framework for mobile devices identifies a subject facing a depth camera given a single frame as input. In order to achieve this goal, we consider two different approaches. First, a clothing descriptor is computed and secondly, a skeletal descriptor is computed from the pose estimation provided by the Microsoft Kinect SDK. These two kinds of soft biometrics have been chosen since they are relatively efficient to compute, which is important in a mobile framework, whilst also showing reasonable performance in previous work.

Person re-identification on mobile devices can differ significantly from the traditional re-identification environment of two or more fixed sensors. Consider the case of a security officer with wearable technology (similar to Google Glass). The mobile infrastructure should consistently detect and identify persons in the officer's field of view using only one sensor, but the target may enter and exit the view multiple times depending on the motion of each party. In this case, there is no longer the concept of separate probe and gallery datasets. An alternative mobile re-identification scenario is where a suspicious person was previously identified and recorded in a probe dataset. Mobile re-identification could help a robot or remotely operated vehicle to locate the subject from the observed sensor data. We focus more on the latter case in terms of probe and gallery datasets.

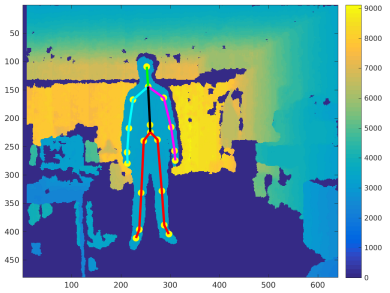
In this section, the pre-processing and features for person re-identification are described. First, a mobile device captures an RGB image and corresponding depth image of the subject using an RGB-D sensor, such as the Microsoft Kinect. A background of the Kinect along with its features and limitations is presented in § 2.2. The Microsoft Kinect SDK is used to provide pose estimation and segmentation of the person from the RGB-D data, since the SDK is available and optimized for this purpose. These steps are shown in Figure 6.2. Microsoft's tracking algorithm can only accurately estimate frontal poses because it is based on a classifier which has only been

trained with frontal poses of people. Hence, we discard frames where any joint is reported by the SDK as untracked or where a face cannot be detected by Viola-Jones.

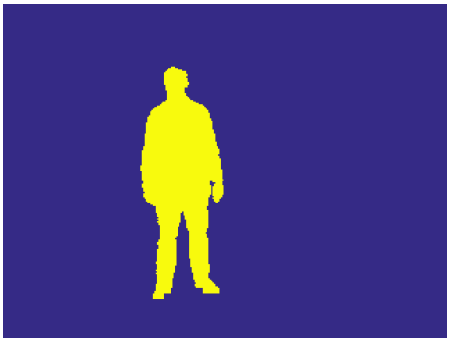
An inherent problem with mobile re-identification is that it is more susceptible to motion blur than traditional re-id with fixed sensor systems. Later in this section, we consider detecting and discarding blurred frames to address this problem.



(A) Input RGB image



(B) Input depth image with predicted pose overlaid



(C) Segmentation

FIGURE 6.2: Pose estimation and person segmentation from RGB-D data.

A local clothing feature vector is calculated for each body part in the pose estimation based on Lab colour, LBP-HF, HOG, skin/hair detection and boundary distance (refer to subsection 6.3.2). These features are normalized and mean-std pooling of the features is calculated on a 4×4 grid. Then pooled feature vectors are concatenated into one representative clothing descriptor and PCA is performed to reduce dimensionality for efficient retrieval. Additionally, the anthropometric descriptor is computed as described below and concatenated for the purpose of re-identification.

6.3.1 Blur Detection

An inherent problem with mobile re-identification is that both the sensor and the subject can be moving, unlike traditional re-identification where only the subject can be moving. Another related issue is that mobile devices and their sensors tend to have significantly less computational processing power available and reduced frame capture speeds when compared to fixed sensor systems. These problems can increase the probability of capturing blurred frames which lack the high frequency detail required to provide an accurate identification of the subject. To address this, we discard frames containing significant motion blur, so that we only process frames where the subject is in focus.

Perhaps the most obvious way to detect the amount of blur in an image would be to compute the Fast Fourier Transform of the image and then analyse the distribution of frequencies. High frequency data represents the details in the image, so the image can be considered blurry if there is a low amount of high frequency data. However, in practice, it is problematic and unintuitive to define thresholds for a low amount of high frequency data and high amount of high frequency data.

Therefore, we examine the state of the art literature. Pertuz et al. [177] present a survey of the field, *Analysis of focus measure operators for shape-from-focus*, where they review 36 different methods to estimate the focus measure of an image.

Ideally, we want to compute a single *blurriness* metric to represent how blurry a given image is. Pertuz et al. review many of these kind of methods in their survey, which range from using greyscale intensity statistics to Local Binary Patterns (LBP).

We utilize the *variation of the Laplacian* approach by Pech-Pacheco et al. [178] since it is very intuitive, relatively efficient, and yields a single blurriness metric. We simply take the greyscale channel of the image and convolve it with the following 3×3 Laplacian kernel:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{6.1}$$

Then we take the variance of the convolution response and if the variance falls below a pre-defined threshold β, the image frame is considered blurry and we discard it:

$$\texttt{Blurred} = \begin{cases} 1 & \text{if } \mathrm{Var}[\texttt{response}] < \beta, \\ 0 & \text{if } \mathrm{Var}[\texttt{response}] \geq \beta. \end{cases} \tag{6.2}$$

The Laplacian operator measures the second derivative of the image such that it highlights regions with rapid greyscale intensity changes. When there is a low variance, there is a small spread of responses, indicating that there are relatively few edges in the image, so we can assume that the image is blurred.

Figure 6.3 depicts the results of our blur detection. The value of the β threshold constant is empirically defined as 130 based on our dataset.

6.3.2 Features

In this section, a discussion of the features employed in this approach is presented.

The texture and shape are described by rotation-invariant local binary pattern histogram Fourier (LBP-HF) features since they have been shown to achieve very good general performance in an efficient manor[179]. The HOG descriptor provides further information about the shape[133]. The boundary distance is given by the negative log distance from the image boundary. The pose distance is given by the negative log distance from the pose estimation joints. A Lab colour descriptor is employed rather than one based on a different colour space such as RGB since it models the human vision system and is more perceptually uniform. Perceptually

(A) Not blurred (B) Blurred

FIGURE 6.3: Detecting and discarding blurred frames.

uniform means that a change in colour value should produce a change of about the same magnitude of visual importance.

Skin/hair detection gives the likelihood of skin or hair at the pixel. Generalized logistic regression is used to compute the likelihood based on Lab, LBP-HF, HOG, boundary distance, and pose distance for input. It is learned from the Fashionista dataset using a one-vs-all strategy.

Skeleton based descriptors yield a signature vector for a given subject based on their anthropometrics and body pose. Generally, skeletal trackers with the highest performance are those that work on 3D data, as opposed to 2D. However, feature-wise, 2D descriptors can perform better than their 3D counterparts [180]. Therefore, we let the Kinect tracker locate skeleton joints in the 3D domain and then re-project them onto the 2D image plane

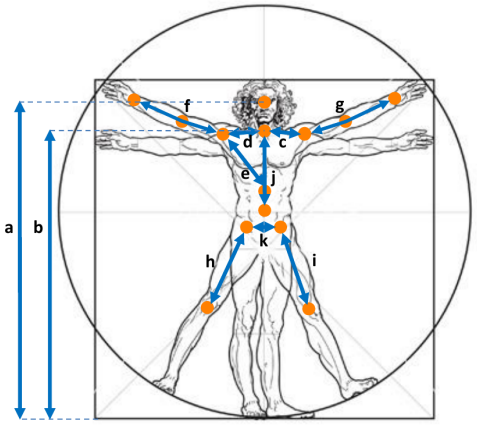


FIGURE 6.4: Distances and ratios utilized for the soft biometric anthropometric descriptor. Image courtesy of [176].

by taking into account the sensor calibration. Once the joints are available in the 2D image domain, the skeleton features can be calculated.

We extract the following 13 skeleton features based on the work of [176]: a) head height, b) neck height, c) neck to left shoulder distance, d) neck to right shoulder distance, e) torso to right shoulder distance, f) right arm length, g) left arm length, h) right upper leg length, i) left upper leg length, j) torso length, k) right hip to left hip distance, l) ratio between torso length and right upper leg length (j/h), m) ratio between torso length and left upper leg length (j/i). These labels (a)-(m) correspond to those depicted in Figure 6.4. The 13 features are then normalized and concatenated to form the skeleton descriptor.

It is assumed that the re-identification system is applied in an indoor scenario or outdoors during summertime when people often wear just one layer of clothing and not large heavy clothes that may occlude key feature points and distort the estimated anthropometrics.

6.3.3 Retrieval

In our approach, there are two retrieval algorithms. The former is used for retrieving identification results and the latter is used for predicting semantic attributes.

In a mobile infrastructure, there is a need for efficient identification and retrieval of subjects based on matching the feature vectors between the probe and each subject that is enrolled in the gallery database. For this purpose, the L_2 distance is minimized over the descriptors to obtain the k nearest neighbours (kNN) in the BIWI dataset. A kd-tree is constructed to efficiently index the samples. A background on kNN and kd-trees is presented in § 2.5.

The second retrieval algorithm is similar but operates on the Paper Doll dataset to retrieve similar clothing attributes to those present in the query image. It only considers the clothing descriptor as input and not the combined clothing and anthropometric descriptor like that used in the former retrieval algorithm.

Note that some retrieval precision could be sacrificed for increased speed by using a more approximate algorithm. This trade-off is explored in [181]. It is shown that with approximations, a speed increase of up to 3 orders of magnitude can be achieved over linear search (kd-trees) if we are willing to accept a lower precision and hence less neighbors returned are exact nearest neighbors. A significant decrease in precision would be unacceptable for re-identification purposes.

6.4 Clothing Parsing

In this section, an approach is described based on the work of Yamaguchi et al. [48] for detecting clothing attributes and localizing clothing items on the query image to enable semantic colour prediction of each clothing item.

Let y_i be the semantic label of the clothing item at pixel i in the image. After clothing attributes have been predicted by the retrieval stage, the algorithm begins to parse the clothing in the query by computing the clothing likelihood S of assigning clothing label l to y_i at pixel level by combining global S_{global} and transfer $S_{transfer}$ models. This likelihood function S is

modeled as:

$$\begin{aligned} S(y_i \mid \mathbf{x}_i, D) \equiv & S_{global}\left(y_i \mid \mathbf{x}_i, D\right)^{\lambda_1} \cdot \\ & S_{transfer}\left(y_i \mid \mathbf{x}_i, D\right)^{\lambda_2} \end{aligned} \tag{6.3}$$

where $\mathbf{x}_i$ denotes the features at pixel i, $\Lambda \equiv [\lambda_1, \lambda_2]$ are weighting parameters. Since the gallery (retrieval) dataset that we utilize has a limited range of clothing items, we introduce a large dataset of tagged fashion images, Paper Doll, for predicting attributes present in the query image. Therefore, we let D be the set of nearest-neighbours retrieved from the Paper Doll dataset.

6.4.1 Global Parsing

Global clothing likelihood is the first term in the clothing parsing model. It is modeled as a logistic regression that computes a likelihood of a label assignment to each pixel for a given set of possible clothing items:

$$S_{global}\left(y_i \mid \mathbf{x}_i, D\right) \equiv P\left(y_i = l \mid \mathbf{x}_i, \theta_l^g\right) \cdot \mathbf{1}[l \in \tau(D)] \tag{6.4}$$

where P is a logistic regression based on the feature vector $\mathbf{x}_i$ and model parameter θ_l^g. Let $\tau(D)$ be a set of predicted clothing attributes given by the Paper Doll nearest-neighbour retrieval stage. Finally, let $\mathbf{1}[\cdots]$ be an indicator function defined as:

$$\mathbf{1}[l \in \tau(D)] = \mathbf{1}_{\tau(D)}(l) = \begin{cases} 1 & \text{if } l \in \tau(D), \\ 0 & \text{if } l \notin \tau(D). \end{cases} \tag{6.5}$$

The following features are calculated for $\mathbf{x}_i$ in the logistic regression: Lab colour, pose distances, LBP-HF, and HOG. Note that unpredicted items are set a probability of 0. The model parameter θ_l^g is trained on all the clothing items in the annotated training subset of the Fashionista dataset.

6.4.2 Transferred Parsing

The transferred parse is the second stage of the parsing model. The mask likelihoods that were estimated by the global parse S_{global} are transferred from the retrieved Paper Doll images to the query image.

First we compute an over-segmentation [182] of both the query and retrieved images. For each super-pixel in the query image, we find the nearest super-pixels in each retrieved image using the aforementioned L2 pose distance and compute a concatenation of bag of words (BoW) from Lab, LBP-HF, and gradient features. The closest super-pixel from each retrieved image is chosen by minimizing the L2 distance on the BoW feature.

Let the transfer model be defined as:

$$S_{transfer}\left(y_i \mid \mathbf{x}_i, D\right) \equiv \frac{1}{Z} \sum_{r \in D} \frac{M(y_i, s_{i,r})}{1 + \|h(s_i) - h(s_{i,r})\|} \tag{6.6}$$

where s_i is the super-pixel of pixel i, $s_{i,r}$ is the corresponding super-pixel from image r, $h(s)$ is the BoW features of super-pixel s, and Z is a normalization constant. Additionally, let us denote M as the mean of the global parse over super-pixel $s_{i,r}$:

$$\begin{aligned} &M(y_i, s_i, r) \equiv \\ &\quad \frac{1}{|s_{i,r}|} \sum_{j \in s_{i,r}} P\left(y_i = l \mid \mathbf{x}_i, \theta_l^g\right) \cdot \mathbf{1}[l \in \tau(r)] \end{aligned} \tag{6.7}$$

where $\tau(r)$is the set of clothing attributes for image r.

6.4.3 Overall Likelihood

Once the two likelihood terms have been computed, the final pixel likelihood S is given by the previously defined equation 6.3.

However, there is the problem of choosing the weighting parameters Λ. The weights are chosen such that an optimal foreground accuracy is achieved from the MAP assignment of pixel labels in the Fashionista training subset:

$$\max_{\Lambda} \sum_{i \in F} \mathbf{1} \left[\tilde{y}_i = \arg\max_{y_i} S_{\Lambda}(y_i \mid \mathbf{x}_i) \right] \tag{6.8}$$

where F are the pixels in the foreground and $\tilde{y}_i$ is the ground-truth annotation of pixel i. The optimization is implemented with a simplex search algorithm.

6.4.4 Semantic Clothing Color

A final stage is presented to assign soft biometric clothing color attributes. These are important to enable natural language searching.

The English language contains eleven main color terms: 'black', 'white', 'red', 'green', 'yellow', 'blue', 'brown', 'orange', 'pink', 'purple', and 'gray'. The estimated clothing color attribute will be chosen from this set.

A color histogram is computed on each localized clothing item detected in the previous stage. In order to convert this numeric color space representation of clothing color to one that is more meaningful to humans, the X11 color names [183] are considered. X11 is part of the X Window System which is commonly found on UNIX like systems such as the popular Ubuntu operating system. X11 colour names are represented in a file which maps certain color strings to color values. The mappings for the eleven main colors mentioned above are extracted from this file, allowing the dominant clothing color value to be matched to the closest value in the list, giving a semantic clothing color attribute for each item of clothing detected in the image.

6.5 Experimental Results

Qualitative results for clothing parsing are shown in Figure 6.5 and Figure 6.6. To aid understanding of the full pipeline, Figure 6.5 uses the same query image as shown earlier in the example of segmentation and pose estimation (Figure 6.2) and also appears as the query in the second row of

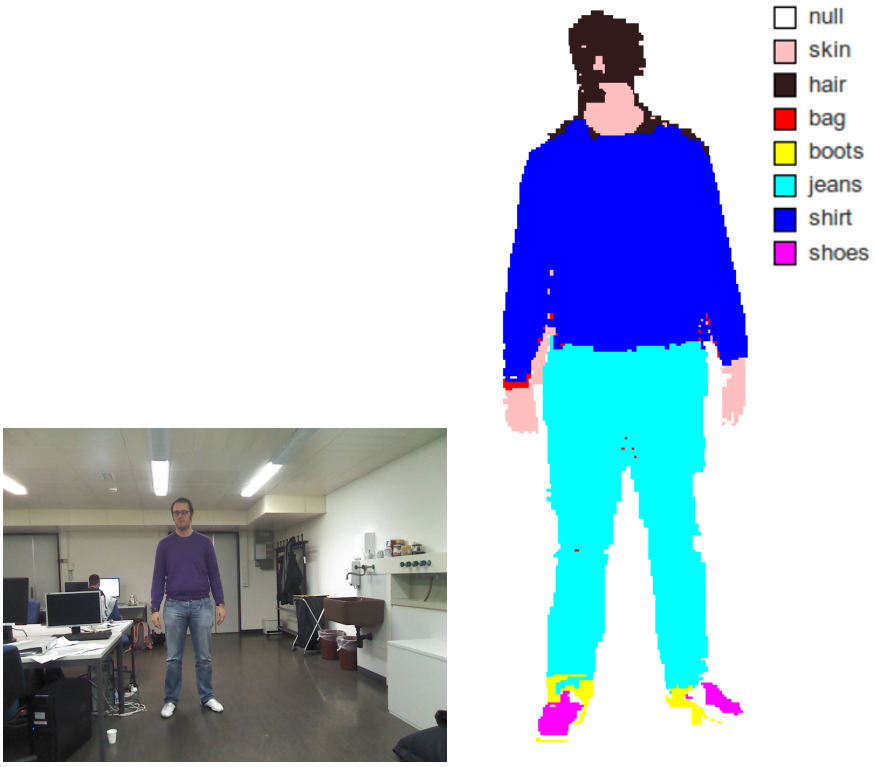
the re-identification retrieval results (Figure 6.7). Figure 6.5 and Figure 6.6 show a reasonable localization of predicted clothing attributes, even in the very challenging case of similarly coloured top and trousers. This is visually comparable to the state of the art [48] and in many cases improved due in part to the feature processing and also the depth-based pose estimation which enables an accurate segmentation and localization of clothing labels.

However, there are some minor parsing inaccuracies such as with hair over-segmentation and dual labelling of shoes and boots. It appears that the hair segmentation is sensitive to shading on the face and the reduced detail in the image caused by motion blur during image capture. This may be improved by performing more advanced correction of non-uniform illumination or capturing a dataset in an environment with more uniform lighting. The dual shoe and boot labelling may be improved by constraining each image to only one type of footwear based on a metric such as pixel voting.

In the first identification experiment, the aim is to qualitatively test short-term person re-identification between the BIWI *still* and *walking* datasets. Although the results are preliminary, our system is capable of operating near real-time and we can see in figure 6.7 that the results are encouraging as the persons are generally correctly re-identified between the *still* and *walking* datasets, implying a favorable rank-1. In some cases, clothing labels which are not observed in the probe image can rank highly - this may in part be due to the similarity of anthropometric features that are used in the retrieval alongside visual cues, and additional optimisation may be required.

Considering related work, the approach in [9] is faster than ours, but is less practical as it is limited to tagging 8 attributes on a simple dataset in a lab environment without consideration of mobile scenarios or RGB-D.

Due to the large-scale nature of the dataset and limited computing resources available, (re-)training, model tuning and comprehensive evaluation are very time consuming. A Hadoop based framework is proposed in the next section to address this problem.

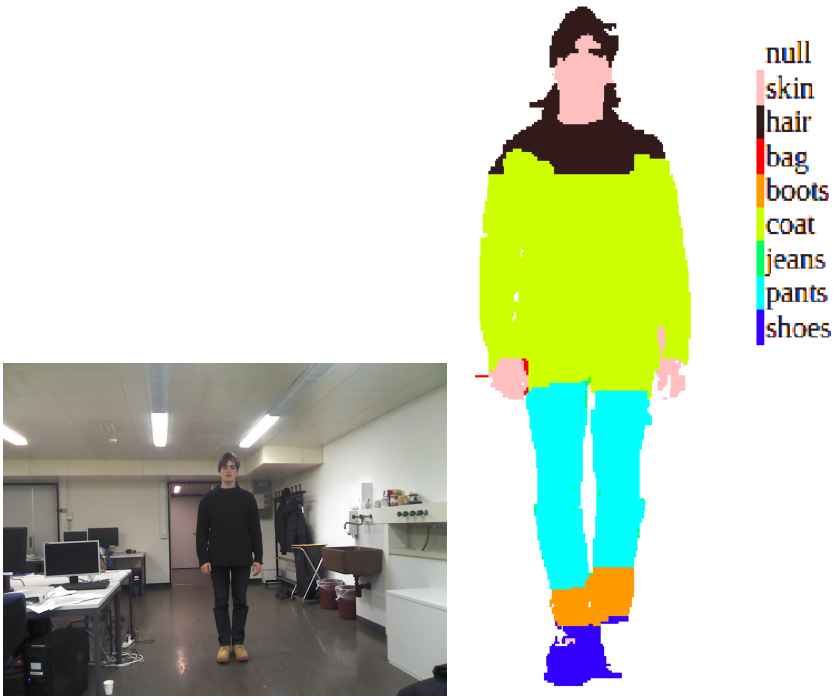


(A) Query image (B) Predicted attributes

FIGURE 6.5: Clothing parsing results. Given the parsing, predicted colour attributes are obtained as *purple shirt*, *blue jeans*, *white shoes/boots*.

The framework consists of the mobile client, which may be a wearable device, smart phone, or robot/drone, and a server. The devices are wirelessly connected over a network such as the internet. In this work, we implement on a smart phone since these are currently the most popular kind of mobile device. Mobile consumer devices with depth sensors are yet to be made available, so we do not capture any input data on the mobile device for testing or demonstration and instead use the pre-recorded RGB-D data from the BIWI dataset as input. The client side is implemented with the Android SDK and NDK to achieve native processing speed and demonstrated on the Samsung Note 4 (1.3GHz Exynos 5433). The server side is implemented in Matlab and C++, running on a quad-core 2.7GHz CPU.

Additionally, we implement on the Google Cardboard platform to demonstrate our new concept of wearable immersive augmented reality re-identification. Google Cardboard is a virtual reality platform developed by Google for use with a smart phone in a head mount made from folded



(A) Query image (B) Predicted attributes

FIGURE 6.6: A challenging image to parse. Although the top is mistaken as a coat occluded in parts by long hair, the black top and bottoms are correctly segmented.

cardboard, as seen in Figure 2.4. Refer to § 2.2 for further reading about the platform.

Figure 6.8 shows the results obtained using a smart phone, where Figure 6.8a is the live real-time image from the real world and Figure 6.8b depicts the predicted attributes which can be augmented into the real scene along with the predicted identity of the person that the Google Cardboard user is looking at. Note that the phone should be placed inside a Google Cardboard head mount for correctly viewing the immersive augmented reality, as the binocular display is split up in to two halves, one for each eye. Currently, there is not a more practical wearable augmented reality system available to demonstrate with. However, the future looks very promising, with a large number of companies known to be developing products, including Google who are working on improving their technology that was originally developed for the now discontinued Google Glass. For implementation, we use the Unity3D engine combined with the Cardboard SDK. These tools can be utilized to build apps for Android and Apple

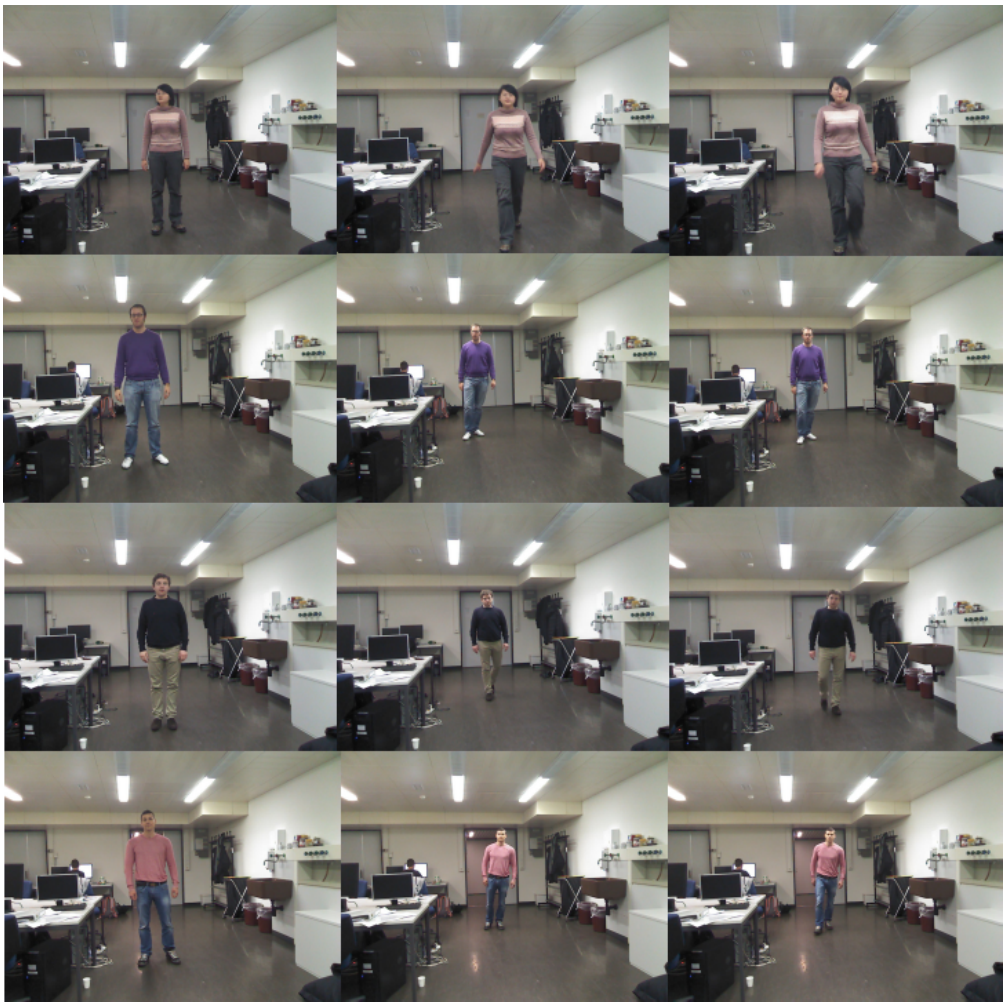


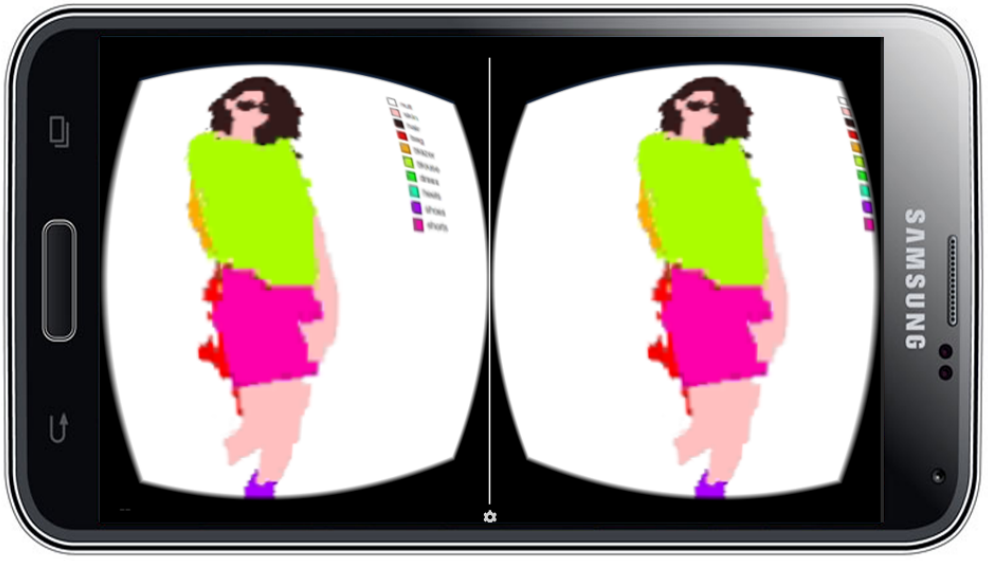
FIGURE 6.7: Preliminary results on the BIWI dataset. First column shows RGB part of the RGB-D image for the subject (probe). Next two columns show retrieval results for subject re-identification.

iOS smart phones that display 3D augmentations with binocular rendering, track and react to head movements, and interact with apps through trigger input.

6.6 Spark and Big Data

Despite the theoretical scalability of the person re-identification algorithm towards handling a large number of images, it was observed that using Matlab to train over 300K images and tune a classifier on the single quad-core 2.7GHz CPU available to us is impractical. Such image collections can be considered as medium to large scale and cannot be stored and

(A)



(B)

FIGURE 6.8: Our prototype Google Cardboard based immersive augmented reality re-identification. The phone is placed inside the Cardboard head mount for correct viewing.

processed efficiently on a single computer. Image processing and machine learning on these medium to large image scale image collections ideally require distributed computing. However, a large compute cluster with the specific proprietry Matlab dependencies was unavailable and the Matlab infrastructure for distributed computing is very costly, proprietary, and limited.

To address these issues, we consider Apache Spark, the industry standard for large scale data processing, and begin porting our implementation to the powerful open-source Python language.

Spark is an open-source framework based on the *MapReduce* principle that allows large scale datasets to be stored and processed in a distributed environment across clusters of computers. Crucially, it can scale up from single servers to thousands of machines based on demand and is tolerant of hardware faults. Two functions are written, called Map and Reduce. The system then manages the parallel execution and coordination of tasks that execute Map or Reduce. Figure 6.9 depicts an overview for execution of a MapReduce program.

Large scale Spark MapReduce tasks are typically run on a computing cluster. Computing nodes with processor chip, memory, and storage are mounted in racks and connected together by high speed Ethernet to form a cluster architecture. Microsoft, Amazon, and Google currently offer Spark in the cloud.

The revised framework therefore consists of Spark, Python, and OpenCV deployed in the Amazon Web Services cloud. A potential avenue for future work is to finish porting the code to the new framework and perform more comprehensive experiments in order to gain further results and insights.

6.7 Summary

In this chapter a person re-identification framework for mobile devices (such as wearable augmented reality glasses, smart phones, and robots) is presented and the BIWI dataset is extended with ground truth clothing labels. For the case of smart phones and tablets, we anticipate that a

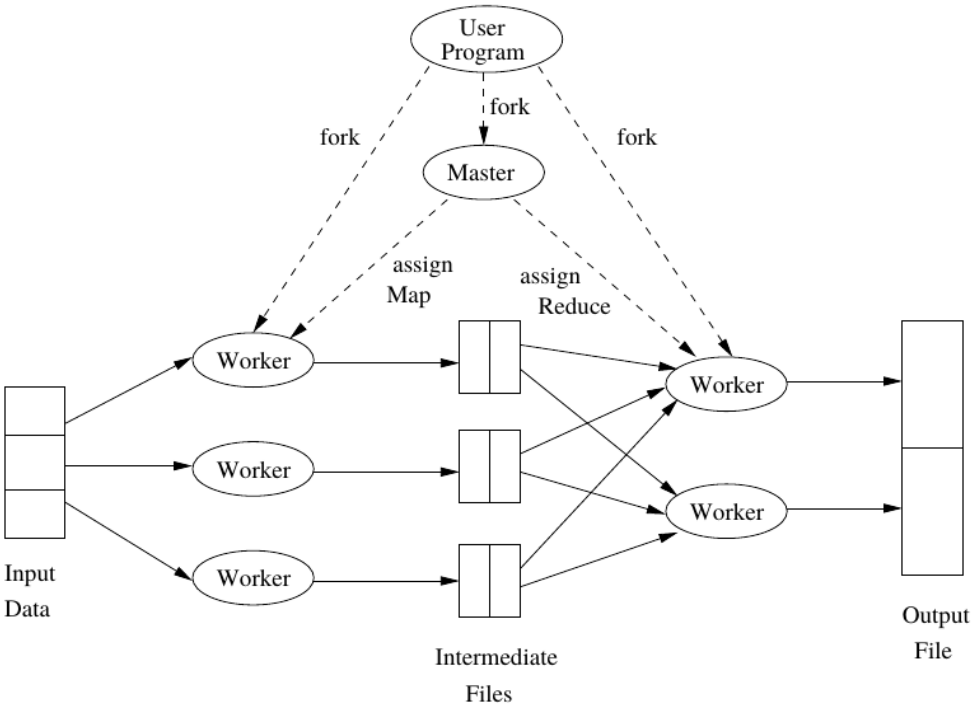


FIGURE 6.9: Execution of a MapReduce program [184].

depth sensor will be integrated into consumer mobile devices in the near future and we currently demonstrate with a pre-recorded input. The mobile device extracts clothing and skeletal features, reducing dimensions by PCA. The features are transmitted over the internet to a server which computes K-nearest neighbours to retrieve the closest matches from persons enrolled in the database and predicts semantic clothing attributes. Predicted clothing labels provide a meaningful soft biometric and can be useful to enable natural language based person searching or to yield a meaningful semantic description even when the subject has not been previously enrolled in the database.

The results presented are preliminary but encouraging. Due to the nature of processing large datasets for machine learning with proprietary Matlab software on a standard PC, major issues were encountered with resource and time constaints. To address this, a new infrastructure consisting of Spark, Python, and the Amazon Web Services cloud is proposed as an avenue for potential future work.

Chapter 7

Conclusions

The dissertation investigated the relatively new and exciting fields of semantic parsing (segmentation/classification) and retrieval of clothing given colour and/or depth images of individuals. Specifically, the problems of *visual clothing search*, *augmented reality try-on of clothing*, and *person re-identification using wearable augmented reality devices* are considered.

The massive continued growth and popularity of social networks and their associated image datasets, augmented reality, and smart phone/tablet/wearable devices (with electronics consumers shifting towards buying and using mobile devices and away from PCs) has created a huge demand for *fast*, *efficient*, and *scalable* image analysis solutions. To this end, these three characteristics underpin the proposed solutions.

In chapter 3, a complete novel mobile client-server system is presented for automatic visual clothes searching of challenging real-world images. A smart phone user can capture a photo (or select a social networking photo) of somebody wearing clothing they like and retrieve similar clothing products that are available at nearby retailers. Our system first identifies the clothing region in the image and segments it from the background using a fast DenseCut implementation. HoG, LBP and novel Lab features are extracted to describe the clothing shape, texture and colour. To enable large scale retrieval on mobile devices, PCA reduces the feature dimensions, Fisher Vectors encode the features, and the encodings are compressed by product quantization. The compressed features are sent to the server alongside the phone's GPS coordinates. Similar products are retrieved

from the database, re-ranked by retailer location and the resulting top product images (including product URLs) are downloaded to the smart phone. The user can then view a map detailing where they can locally purchase the products or click their associated URLs for purchasing online.

In chapter 4, computer vision techniques were investigated for automatic semantic segmentation/parsing of dressed clothing in real-world photos. It is shown that the proposed framework is able to segment clothing and classify clothing attributes more efficiently than existing state of the art methods, whilst achieving good accuracy and robustness on a difficult dataset. We demonstrate the approach with an augmented reality mirror app for mobile tablet devices that can segment a user's clothing in real-time and enable them to realistically see themselves in the virtual mirror wearing variations of the clothing with different colours (or graphics rendered as per chapter 5). Although the approach is limited to predominantly uniformly coloured clothing (which may contain textured regions), it may also be of particular benefit to emerging real-time augmented reality applications such as in sports broadcasting and computer gaming.

In chapter 5, the aforementioned augmented reality mirror is extended with a hierarchical approach for 3D geometric reconstruction of highly deformable surfaces, such as cloth, which is robust to partially untextured regions given consecutive monocular video frames or a single image and a texture template. A real-time retexturing and recoloring framework has been demonstrated for combining this method with clothing parsing for the purpose of augmented reality upper body clothing try on. Robustness has been shown to relatively large timesteps (i.e. using a webcam or mobile device) under uncontrolled domestic lighting with variation in clothing shape and color, texture shape and color, subject, and background. Empirical results show convincing 3D shape reconstruction and photorealistic retexturing of graphics on clothing whilst employing a setup which is practical for a consumer and could be run on smart phones/tablets.

In 6, a person re-identification framework for mobile devices (such as wearable augmented reality glasses, smart phones, and robots) is presented and the BIWI dataset is extended with ground truth clothing labels. For the case of smart phones and tablets, we anticipate that a depth sensor will be integrated into consumer mobile devices in the near future and

we currently demonstrate with a pre-recorded input. The mobile device extracts clothing and skeletal features, reducing dimensions by PCA. The features are transmitted over the internet to a server which computes K-nearest neighbours to retrieve the closest matches from persons enrolled in the database and predicts semantic clothing attributes. Predicted clothing labels provide a meaningful soft biometric and can be useful to enable natural language based person searching or to yield a meaningful semantic description even when the subject has not been previously enrolled in the database. The results presented are preliminary but encouraging.

This dissertation offers insights into the fields of clothing retrieval, clothing parsing for augmented reality, and soft biometrics which can be applied to many important and practical applications including crime suspect identification, visual product search for clothing ecommerce, product retrieval for fashion advertising in social networks, personal fashion analysis, in-store customer profiling, and augmented reality clothes try on.

The number of photos uploaded and shared on online social networks continues to have strong growth and a significant proportion of these images appear to contain people wearing clothing. Clothing can be considered to be one of the core cues of human appearance and can also be used to infer information about the wearer. However, the segmentation and parsing of clothing worn on a subject is challenging due to the wide diversity of clothing designs, the complexity of scene lighting, dynamic backgrounds, and self/third-party occlusions. Therefore, it is clear that the challenges in the fields of clothing parsing and soft biometrics will remain a key goal of computer vision researchers for years to come.

Various avenues for future work have been discussed at the end of the previous chapters. Despite the theoretical scalability of the person re-identification algorithm towards handling a large number of images, the Matlab infrastructure coupled with the lack of access to a large compute cluster with the specific proprietry dependencies imposed tough challenges on carrying out a comprehensive large scale analysis. Hence, the main avenue for potential future work is to complete the transition of porting the re-identification implementation away from Matlab to an open-source Python and Spark infrastructure. Spark is an open-source, distributed processing system commonly used for big data workloads and

without the limitations of Matlab. A Spark based application can be deployed for a low cost in the cloud, such as with Amazon Web Services. This would enable further results and insights to be gathered.

References

[1] Emarketer. Retail Sales Worldwide Will Top $22 Trillion This Year - eMarketer. http://www.emarketer.com/Article/Retail-Sales-Worldwide-Will-Top-22-Trillion-This-Year/1011765, 2014.

[2] Statista. Global apparel market size projections 2012-2025. http://www.statista.com/statistics/279757/apparel-market-size-projections-by-region/, 2015.

[3] "Kleiner Perkins Caufield Byers". 2015 Internet Trends. http://www.kpcb.com/internet-trends.

[4] G. A. Cushen and M. S. Nixon. Mobile Visual Clothing Search. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*. IEEE, 2013.

[5] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*. IEEE, 2012.

[6] X. Wang and T. Zhang. Clothes search in consumer photos via color matching and attribute learning. In *MM*, pages 1353–1356. ACM, 2011. doi: 10.1145/2072298.2072013.

[7] Y. Song and T. Leung. Context-aided human recognition–clustering. In *ECCV*, pages 382–395. Springer, 2006.

[8] A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *CVPR 2008*, pages 1–8. IEEE, 2008.

[9] M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In *IEEE ICIP*, pages 2937–2940, Brussels, Belgium, 2011.

[10] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H. P. Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum*, volume 28, pages 337–346. Blackwell Publishing, 2009.

[11] X. Chao, M. J. Huiskes, T. Gritti, and C. Ciuhu. A framework for robust feature selection for real-time fashion style recommendation. In *Workshop on IMCE*. ACM, 2009.

[12] H. Chen, A. Gallagher, and B. Girod. Describing Clothing by Semantic Attributes. In *ECCV*. Springer, 2012.

[13] CCS Insight. Augmented and Virtual Reality Device Forecast, 2015-2019. http://www.ccsinsight.com/press/company-news/2251-augmented-and-virtual-reality-devices-to-become-a-4-billion-plus-business-in-three-years, 2015.

[14] AppLift. The Global Mobile Games Market (Infographic). http://www.applift.com/blog/mobile-games-market.html, October 2013.

[15] KZero. Consumer Virtual Reality market worth $5.2bn by 2018. http://www.kzero.co.uk/blog/consumer-virtual-reality-market-worth-13bn-2018/, 2015.

[16] Companies and Markets. Biometrics: Technologies and Global Markets. Technical report, Companies and Markets, January 2014.

[17] TechNavio. Biometrics Market in Europe 2014-2018. Technical report, TechNavio, July 2014.

[18] House of Commons Science and Technology Committee. Current and future uses of biometric data and technologies. Technical Report Sixth Report of Session 2014–15, House of Commons Science and Technology Committee, March 2015.

[19] Northrop Grumman. Written evidence submitted by Northrop Grumman (BIO0030). Technical report, Northrop Grumman, September 2014.

[20] Gartner Inc. Worldwide device shipments by segment. http://www.gartner.com/technology/home.jsp, 2015.

[21] G. A. Cushen and M. S. Nixon. Markerless Real-Time Garment Retexturing From Monocular 3D Reconstruction. In *IEEE ICSIPA*, pages 88–93, Malaysia, November 2011.

[22] G. A. Cushen and M. S. Nixon. Real-Time Semantic Clothing Segmentation. In *ISVC*, pages 272–281. Springer, 2012.

[23] G. A. Cushen. A Person Re-Identification System For Mobile Devices. In *Signal Image Technology & Internet Systems (SITIS)*. IEEE, 2015.

[24] Y. Kalantidis, L. Kennedy, and L. J. Li. Getting the Look: Clothing Recognition and Segmentation for Automatic Product Suggestions in Everyday Photos. In *International Conference on Multimedia Retrieval (ICMR)*, Dallas, TX, April 2013. ACM.

[25] Nataraj Jammalamadaka, Ayush Minocha, Digvijay Singh, and C. V. Jawahar. Parsing clothes in unrestricted images. In *Proc. of British Machine Vision Conference*. The British Machine Vision Association (BMVA), 2013.

[26] Jia-Lin Chen, Wan-Yu Chen, I-Kuei Chen, Chung-Yu Chi, and Liang-Gee Chen. Interactive clothing retrieval system. *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, pages 347–348, 10-13 Jan. 2014. ISSN 2158-3994. doi: 10.1109/ICCE.2014.6776035.

[27] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. Retrieving Similar Styles to Parse Clothing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(5):1028–1040, 2015.

[28] Kota Yamaguchi, Takayuki Okatani, Kyoko Sudo, Kazuhiko Murasaki, and Yukinobu Taniguchi. Mix and Match: Joint Model for Clothing and Attribute Recognition. In *British Machine Vision Conference*. BMVA, 2015.

[29] Andres Traumann, Gholamreza Anbarjafari, and Sergio Escalera. A New Retexturing Method for Virtual Fitting Room Using Kinect 2 Camera. In *Computer Vision and Pattern Recognition (CVPR)*, pages 75–79, Boston, USA, 2015. IEEE.

[30] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval

via parts alignment and auxiliary set. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3330–3337, 16-21 June 2012. ISSN 1063-6919. doi: 10.1109/CVPR.2012.6248071.

[31] J. Sivic, C. L. Zitnick, and R. Szeliski. Finding people in repeated shots of the same scene. In *BMVC*, volume 3, pages 909–918, 2006.

[32] M. W. Lee and I. Cohen. A model-based approach for estimating human 3D poses in static images. *IEEE TPAMI*, pages 905–916, 2006.

[33] Si Liu, Jiashi Feng, Zheng Song, Tianzhu Zhang, Hanqing Lu, Changsheng Xu, and Shuicheng Yan. Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM international conference on Multimedia*, pages 619–628, Nara, Japan, 2012. ACM. ISBN 978-1-4503-1089-5.

[34] Hui Zhang, Jason E. Fritts, and Sally A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *computer vision and image understanding*, 110(2):260–280, 2008.

[35] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Computer Vision–ECCV 2006*, pages 1–15. Springer, 2006. ISBN 3-540-33832-2.

[36] Stephen Gould, Tianshi Gao, and Daphne Koller. Region-based segmentation and object detection. In *Advances in neural information processing systems*, pages 655–663, 2009.

[37] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1972–1979. IEEE, 2009. ISBN 1-4244-3992-2.

[38] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010. ISBN 3-642-15554-5.

[39] David Eigen and Rob Fergus. Nonparametric image parsing using adaptive neighbor sets. In *Computer vision and pattern recognition*

(CVPR), 2012 IEEE Conference on, pages 2799–2806. IEEE, 2012. ISBN 1-4673-1226-6.

[40] Gautam Singh and Jana Košecká. Semantic Context for Nonparametric Scene Parsing and Scene Classification. In *Scene Understanding Workshop, CVPR*. Citeseer, 2013.

[41] Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3001–3008. IEEE, 2013. ISBN 1063-6919.

[42] Y. Schnitman, Y. Caspi, D. Cohen-Or, and D. Lischinski. Inducing semantic segmentation from an example. In *ACCV 2006*, pages 373–384. Springer, 2006.

[43] Z. Hu, H. Yan, and X. Lin. Clothing segmentation using foreground and background estimation based on the constrained Delaunay triangulation. *Pattern Recognition*, 41(5):1581–1592, 2008.

[44] B. Hasan and D. Hogg. Segmentation using Deformable Spatial Priors with Application to Clothing. In *BMVC*, pages 1–11, 2010.

[45] N. Wang and H. Ai. Who Blocks Who: Simultaneous Clothing Segmentation for Grouping Images. In *ICCV*, November 2011.

[46] S. Vittayakorn, K. Yamaguchi, A.C. Berg, and T.L. Berg. Runway to Realway: Visual Analysis of Fashion. *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 951–958, 5-9 Jan. 2015. doi: 10.1109/WACV.2015.131.

[47] Kota Yamaguchi, Tamara L. Berg, and Luis E. Ortiz. Chic or Social: Visual Popularity Analysis in Online Fashion Networks. In *Proceedings of the ACM International Conference on Multimedia*, pages 773–776. ACM, 2014. ISBN 1-4503-3063-0.

[48] K. Yamaguchi, M.H. Kiapour, and T.L. Berg. Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items. *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3519–3526, 1-8 Dec. 2013. ISSN 1550-5499. doi: 10.1109/ICCV.2013.437.

[49] Marco Manfredi, Costantino Grana, Simone Calderara, and Rita Cucchiara. A complete system for garment segmentation and color classification. *Machine Vision and Applications*, 25(4):955–969, 2014.

[50] Li Liu, Ruomei Wang, Fan Zhou, Zhuo Su, and Xiaodong Fu. Semantic Segmentation and Labeling of 3D garments. In *Digital Home (ICDH), 2014 5th International Conference on*, pages 299–304. IEEE, 2014. ISBN 1-4799-4285-5.

[51] A. Hilsmann and P. Eisert. Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications. In *MIRAGE 2009*, page 94, 2009.

[52] Wei Di, C. Wah, A Bhardwaj, R. Piramuthu, and N. Sundaresan. Style Finder: Fine-Grained Clothing Style Detection and Retrieval. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 8–13, 23-28 June 2013. doi: 10.1109/CVPRW.2013.6.

[53] Qiang Chen, Junshi Huang, Rogerio Feris, Lisa M. Brown, Jian Dong, and Shuicheng Yan. Deep Domain Adaptation for Describing People Based on Fine-Grained Clothing Attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324, 2015.

[54] Roberto Vezzani, Davide Baltieri, and Rita Cucchiara. People reidentification in surveillance and forensics: A survey. *ACM Computing Surveys (CSUR)*, 46(2):29, 2014.

[55] Shaogang Gong, Marco Cristani, and Shuicheng Yan. *Person Re-Identification (Advances in Computer Vision and Pattern Recognition)*. Springer, January 2014. ISBN 1-4471-6295-1.

[56] Daniel Reid. *Human identification using soft biometrics*. PhD thesis, University of Southampton, 2013.

[57] D. Reid and M. Nixon. Using comparative human descriptions for soft biometrics. In *IJCB*. IEEE, 2011.

[58] A. Jain, S. Dass, and K. Nandakumar. Soft biometric traits for personal recognition systems. In *Biometric Authentication*. Springer, 2004.

[59] A. Dantcheva, C. Velardo, A. D'angelo, and J. Dugelay. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, 51(2), 2011.

[60] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*. IEEE, 2010.

[61] S. Bak, E. Corvee, F. Brémond, and M. Thonnat. Person re-identification using haar-based and dcd-based signature. In *AVSS*. IEEE, 2010.

[62] W. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*. IEEE, 2011.

[63] Emad Sami Jaha and Mark S. Nixon. Soft Biometrics for Subject Identification using Clothing Attributes. In *International Joint Conference on Biometrics (IJCB 2014)*, 2014.

[64] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. Sarc3d: a new 3d body model for people tracking and re-identification. In *Image Analysis and Processing–ICIAP 2011*, pages 197–206. Springer, 2011. ISBN 3-642-24084-4.

[65] I. Barbosa, M. Cristani, A. Del Bue, L. Bazzani, and V. Murino. Re-identification with RGB-D Sensors. In *ECCV Workshops*. Springer, 2012.

[66] Emad Sami Jaha and Mark S Nixon. Analysing Soft Clothing Biometrics for Retrieval. In *International Workshop on Biometrics (BIOMET 2014)*, pages 234–245. Springer International Publishing, 2014. ISBN 3-319-13385-3.

[67] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. Mapping Appearance Descriptors on 3D Body Models for People Re-identification. *International Journal of Computer Vision*, 111(3):345–364, 2015.

[68] R. Azuma. A survey of augmented reality. *Presence-Teleoperators and Virtual Environments*, 6(4):355–385, 1997.

[69] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.

[70] D. Bradley, G. Roth, and P. Bose. Augmented reality on cloth with realistic illumination. *Machine Vision and Applications*, 20(2):85–92, 2009.

[71] V. Gay-Bellile, A. Bartoli, and P. Sayd. Deformable Surface Augmentation in Spite of Self-Occlusions. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4. IEEE Computer Society, 2007.

[72] J. Pilet, V. Lepetit, and P. Fua. Real-time nonrigid surface detection. In *Computer Vision and Pattern Recognition, 2005*, volume 1, pages 822–828. IEEE, 2005.

[73] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

[74] V. Scholz and M. Magnor. Texture replacement of garments in monocular video sequences. *Rendering techniques*, 2006:305–312, 2006.

[75] R. White and D. Forsyth. Retexturing single views using texture and shading. *Computer Vision–ECCV 2006*, pages 70–81, 2006.

[76] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *CVPR '06*, pages 519–528, 2006.

[77] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. In *ACM SIGGRAPH 2007*, pages 34–es. ACM, 2007.

[78] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. In *ACM SIGGRAPH 2008 papers*, pages 1–9. ACM, 2008.

[79] K. S. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popović, and S. M. Seitz. Estimating cloth simulation parameters from video. In *ACM SIGGRAPH SCA '03*, pages 37–51, 2003.

[80] L. D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1131–1147, 1993.

[81] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 518–523. IEEE, 1993.

[82] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 337–343. IEEE, 1993.

[83] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 715–729, 1991.

[84] L. V. Tsap, D. B. Goldof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(5):526–543, 2000.

[85] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[86] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Computer Vision—ECCV'98*, page 484, 1998.

[87] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

[88] L. Torresani, A. Hertzmann, and C. Bregler. Learning non-rigid 3d shape from 2d motion. *Advances in Neural Information Processing Systems*, 16, 2004.

[89] A. Ecker, A. Jepson, and K. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. *Computer Vision–ECCV 2008*, pages 127–140, 2008.

[90] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International Journal of Computer Vision*, pages 1–14, 2010.

[91] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. *Computer Vision–ACCV 2010*, pages 52–66, 2011.

[92] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3d surface registration. In *ECCV*, pages 581–594, 2008.

[93] B. K. P. Horn and M. J. Brooks, editors. *Shape from shading*. MIT Press, Cambridge, MA, USA, 1989.

[94] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.

[95] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, 1999.

[96] A. H. Ahmed and A. A. Farag. A new formulation for shape from shading for non-lambertian surfaces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1817–1824. IEEE, 2006.

[97] Hing N. Ng and Richard L. Grimsdale. Computer Graphics Techniques for Modeling Cloth. *IEEE Comput. Graph. Appl.*, 16(5):28–41, 1996.

[98] D. H. House and D. E. Breen. *Cloth modeling and animation*. AK Peters, Ltd. Natick, MA, USA, 2000.

[99] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 205–214. ACM New York, NY, USA, 1987.

[100] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, page 278. ACM, 1988.

[101] J. Weil. The synthesis of cloth objects. *ACM Siggraph Computer Graphics*, 20(4):49–54, 1986.

[102] J. A. Thingvold and E. Cohen. Physical modeling with B-spline surfaces for interactive design and animation. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 129–137, Snowbird, Utah, United States, 1990. ACM.

[103] M. Carignan, Y. Yang, M. Thalmann, and D. Thalmann. Dressing animated synthetic actors with complex deformable clothes. *SIGGRAPH Comput. Graph.*, 26(2):99–104, 1992.

[104] H. Okabe, H. Imaoka, T. Tomiha, and H. Niwaya. Three dimensional apparel CAD system. *ACM SIGGRAPH Computer Graphics*, 26(2):110, 1992.

[105] David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 365–372, New York, NY, USA, 1994. ACM.

[106] P. Volino, N. M. Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph.*, 28(4):1–16, 2009.

[107] D. Baraff, A. Witkin, and M. Kass. Untangling cloth. *ACM Trans. Graph.*, 22(3):862–870, July 2003. doi: 10.1145/882262.882357.

[108] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 28–36, San Diego, California, 2003. Eurographics Association. ISBN 1-58113-659-5.

[109] L. D. Cutler, R. Gershbein, X. C. Wang, C. Curtis, E. Maigret, L. Prasso, and P. Farson. An art-directed wrinkle system for CG character clothing. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 125. ACM, 2005.

[110] X. Provot. Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior. In Wayne A. Davis and Przemyslaw Prusinkiewicz, editors, *Graphics Interface '95*, pages 147–154. Canadian Human-Computer Communications Society, 1995.

[111] D. Baraff and A. Witkin. Large Steps in Cloth Simulation. *Computer Graphics*, 32(Annual Conference Series):43–54, 1998.

[112] O. Etzmuss, M. Keckeisen, and W. Strasser. A fast finite element solution for cloth modelling. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*, pages 244–251, 2003.

[113] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 140. Eurographics Association, 2004.

[114] A. Selle, J. Su, G. Irving, and R. Fedkiw. Robust High-Resolution Cloth Using Parallelism, History-Based Collisions, and Accurate Friction. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2): 339–350, March 2009. doi: 10.1109/TVCG.2008.79.

[115] Matt Weinberger. Why Xbox Kinect didn't take off - Business Insider. http://uk.businessinsider.com/why-microsoft-xbox-kinect-didnt-take-off-2015-9, September 2015.

[116] Kourosh Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS workshop laser scanning*, volume 38, page W12, 2011.

[117] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[118] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and*

Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007. ISBN 1-4244-1179-3.

[119] Ken Chatfield, Victor S. Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, volume 2, page 8, 2011.

[120] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[121] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010. ISBN 3-642-15560-X.

[122] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006. ISBN 0-7695-2597-0.

[123] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012. ISBN 1-4673-1226-6.

[124] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.

[125] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1169–1176. IEEE, 2009. ISBN 1-4244-3992-2.

[126] Hervé Jégou, Florent Perronnin, Matthijs Douze, Javier Sanchez, Pablo Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, 2012.

[127] You Jia, Jingdong Wang, Gang Zeng, Hongbin Zha, and Xian-Sheng Hua. Optimizing kd-trees for scalable visual descriptor indexing. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3392–3399. IEEE, 2010. ISBN 1-4244-6984-8.

[128] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2130–2137. IEEE, 2009. ISBN 1-4244-4420-9.

[129] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[130] Marcin Eichner, Vittorio Ferrari, and S. Zurich. Better Appearance Models for Pictorial Structures. In *BMVC*, volume 2, page 5, 2009.

[131] Ming-Ming Cheng, Victor Adrian Prisacariu, Shuai Zheng, Philip HS Torr, and Carsten Rother. DenseCut: Densely Connected CRFs for Realtime GrabCut. In *Computer Graphics Forum*, volume 34, pages 193–201. Wiley Online Library, 2015. ISBN 1467-8659.

[132] D. Chai and K. N. Ngan. Face segmentation using skin-color map in videophone applications. *CSVT, IEEE Trans on*, 9(4):551–564, 1999.

[133] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. ISBN 0-7695-2372-2.

[134] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[135] T. Sikora. The MPEG-7 visual standard for content description-an overview. *CSVT, IEEE Trans on*, 11(6):696–702, 2001.

[136] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(2), 1998.

[137] Brent Berlin and Paul Kay. *Basic color terms: Their universality and evolution*. Univ of California Press, 1991. ISBN 0-520-07635-4.

[138] Paul Kay and Luisa Maffi. Color appearance and the emergence and evolution of basic color lexicons. *American anthropologist*, 101(4): 743–760, 1999.

[139] "World Wide Web Consortium". CSS Color Module Level 3. https://www.w3.org/TR/css3-color/, 2011.

[140] Andrew D. Bagdanov, Lamberto Ballan, Marco Bertini, and Alberto Del Bimbo. Trademark matching and retrieval in sports video databases. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 79–86. ACM, 2007. ISBN 1-59593-778-1.

[141] Wei-Ta Chu and Tsung-Che Lin. Logo recognition and localization in real-world images by using visual patterns. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 973–976. IEEE, 2012. ISBN 1-4673-0045-4.

[142] Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 581–584. ACM, 2009. ISBN 1-60558-608-0.

[143] Jim Kleban, Xing Xie, and Wei-Ying Ma. Spatial pyramid mining for logo detection in natural scenes. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1077–1080. IEEE, 2008. ISBN 1-4244-2570-0.

[144] Christoph H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 987–994. IEEE, 2009. ISBN 1-4244-4420-9.

[145] Jingjing Meng, Junsong Yuan, Yuning Jiang, Nitya Narasimhan, Venu Vasudevan, and Ying Wu. Interactive visual object search through mutual information maximization. In *Proceedings of the international conference on Multimedia*, pages 1147–1150. ACM, 2010. ISBN 1-60558-933-0.

[146] Raymond Phan and Dimitrios Androutsos. Content-based retrieval of logo and trademarks in unconstrained color image databases using color edge gradient co-occurrence histograms. *Computer Vision and Image Understanding*, 114(1):66–84, 2010.

[147] Apostolos P. Psyllos, Christos-Nikolaos E. Anagnostopoulos, and Eleftherios Kayafas. Vehicle logo recognition using a sift-based enhanced matching scheme. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):322–328, 2010.

[148] Jerome Revaud, Matthijs Douze, and Cordelia Schmid. Correlation-based burstiness for logo retrieval. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 965–968. ACM, 2012. ISBN 1-4503-1089-3.

[149] Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, Trento, Italy, 2011. ACM. ISBN 978-1-4503-0336-1.

[150] Chia-Hung Wei, Yue Li, Wing-Yin Chau, and Chang-Tsun Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386–394, 2009.

[151] Pengfei Xu, Hongxun Yao, and Rongrong Ji. SIGMA: Spatial integrated matching association algorithm for logo detection. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1086–1089. IEEE, 2010. ISBN 1-4244-4295-8.

[152] Wei Xu and Yang Yi. A robust replay detection algorithm for soccer video. *Signal Processing Letters, IEEE*, 18(9):509–512, 2011.

[153] Jiebo Luo and David Crandall. Color object detection using spatial-color joint probability functions. *Image Processing, IEEE Transactions on*, 15(6):1443–1453, 2006.

[154] The Anh Pham, Mathieu Delalandre, and Sabine Barrat. A contour-based method for logo detection. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 718–722. IEEE, 2011. ISBN 1-4577-1350-0.

[155] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[156] Guoshen Yu and Jean-Michel Morel. Asift: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 2011, 2011.

[157] A. Gallagher and T. Chen. Understanding Images of Groups Of People. In *CVPR*, 2009.

[158] N. M. Thalmann, E. Lyard, M. Kasap, and P. Volino. Adaptive Body, Motion and Cloth. In *Motion in Games*, page 71. Springer, November 2008.

[159] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679, 1981.

[160] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000.

[161] F. L. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, June 1989. ISSN 0162-8828. doi: 10.1109/34.24792.

[162] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Analyse Numerique*, 10:5–12, 1976.

[163] A. Bartoli, M. Perriollat, and S. Chambon. Generalized thin-plate spline warps. *International journal of computer vision*, 88(1):85–110, 2010.

[164] M. Muller, B. Heidelberger, M. Hennix, and J. Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[165] M. Teschner, B. Heidelberger, M. M"uller, D. Pomeranets, and M. Gross. Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of Vision, Modeling, Visualization VMV'03*, pages 47–54, 2003.

[166] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[167] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *Image Processing, IEEE Transactions on*, 10(8):1200–1211, 2001.

[168] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.

[169] A. Hilsmann and P. Eisert. Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[170] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2288–2295. IEEE, 2012. ISBN 1-4673-1226-6.

[171] Tamar Avraham, Ilya Gurvich, Michael Lindenbaum, and Shaul Markovitch. Learning implicit transfer for person re-identification. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 381–390. Springer, 2012. ISBN 3-642-33862-3.

[172] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3586–3593. IEEE, 2013. ISBN 1063-6919.

[173] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2360–2367. IEEE, 2010. ISBN 1-4244-6984-8.

[174] Pietro Salvagnini, Loris Bazzani, Matteo Cristani, and Vittorio Murino. Person re-identification with a ptz camera: an introductory study. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3552–3556. IEEE, 2013.

[175] Ryan Layne, Timothy M. Hospedales, and Shaogang Gong. Investigating Open-World Person Re-identification Using a Drone. In *Computer Vision-ECCV 2014 Workshops*, pages 225–240. Springer, 2014. ISBN 3-319-16198-9.

[176] Matteo Munaro, Andrea Fossati, Alberto Basso, Emanuele Menegatti, and Luc Van Gool. One-shot person re-identification with a consumer depth camera. In *Person Re-Identification*, pages 161–181. Springer, 2014. ISBN 1-4471-6295-1.

[177] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.

[178] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martinez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 3, pages 314–317. IEEE, 2000. ISBN 0-7695-0750-6.

[179] Guoying Zhao, Timo Ahonen, Jiří Matas, and Matti Pietikäinen. Rotation-invariant image and video description with local binary pattern features. *Image Processing, IEEE Transactions on*, 21(4): 1465–1477, 2012.

[180] Loris Nanni, Matteo Munaro, Stefano Ghidoni, Emanuele Menegatti, and Sheryl Brahnam. Ensemble of different approaches for a reliable person re-identification system. *Applied Computing and Informatics*, 2015.

[181] Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP (1)*, 2, 2009.

[182] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[183] Wikipedia. X11 color names - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/X11_color_names#Color_names_identical_between_X11_and_HTML.2FCSS, 2015.

[184] Jure Leskovec, Anand Rajaraman, and Jeff Ullman. Mining of Massive Datasets. http://www.mmds.org/, 2015.