FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Martin Côté

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Video Segmentation for Markerless Motion Capture in Unconstrained Environments

TITRE DE LA THÈSE / TITLE OF THESIS

Prof. P. Payeur

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Prof. W. Lee

Prof. A. Whitehead

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Video Segmentation for Markerless Motion Capture in Unconstrained Environments

By:

Martin Côté

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree of

Master in Applied Science

Ottawa-Carleton Institute of Electrical and Computer Engineering

School of Information Technology and Engineering

Faculty of Engineering

University of Ottawa

Your file Votre référence
ISBN: 978-0-494-49181-2
Our file Notre référence
ISBN: 978-0-494-49181-2

Abstract

Segmentation is an important first step in many computer vision applications. The identification of key regions within an image or video allows for a higher level analysis of the media content. This thesis explores the application of this low level process to the monitoring of human performance. In such a context, a proposed segmentation algorithm would be required to impose a minimum of constraints in order to assure the integrity of the performance and the proper transfer of key data to higher level analysis components.

Classical approaches to the segmentation problem either make assumptions on the content of the media or impose unreasonable constraints on their targets and environments. In doing so, the integrity of performance measurements cannot be assured and semantic interpretation therefore becomes skewed. The method presented within this thesis allows for unconstrained environments by using a spatiotemporal colour-texture segmentation routine that represents the media content as a set of homogenous texture regions. The routine is assisted by a non-parametric clustering algorithm in order to produce an initial colour-texture representation. The regions obtained from this algorithm undergo a merging and tracking process in order to produce a final segmented representation of a target. Experimental results reveal that the system is robust for complex environments and provides several advantages over current segmentation processes.

Acknowledgements

I would like to thank my thesis supervisor Dr. Pierre Payeur for his direction, guidance and assistance throughout my studies. I would also like to acknowledge the contributions from the University of Ottawa, the Natural Sciences and Engineering Research Council of Canada and the Piano Pedagogy Research Laboratory for their financial and resource contributions to this project. Thanks also go to the musicians, for their time and effort that have made this work possible. Finally, I would also like to give thanks to my family and friends for their support, patience and encouragement.

Table of Contents

ABSTRACT ..... **II**

ACKNOWLEDGEMENTS ..... **III**

TABLE OF CONTENTS ..... **IV**

TABLE OF FIGURES ..... **VII**

TABLE OF TABLES ..... **IX**

CHAPTER 1 INTRODUCTION **1**

1.1 CONTEXT 1

1.2 MOTIVATIONS 3

1.3 CHALLENGES 4

1.4 OBJECTIVES 5

1.5 PROPOSED FRAMEWORK 5

1.6 ORGANIZATION 7

CHAPTER 2 LITERATURE REVIEW **9**

2.1 TRADITIONAL MOTION CAPTURE TECHNIQUES 9

2.2 MOTION CAPTURE SEGMENTATION TECHNIQUES 11

2.2.1 Sequential Image Techniques *12*

2.2.1.1 Background Modelling Methods 12

2.2.1.2 Contour-Based Methods 15

2.2.1.3 Region-Based Methods 16

2.2.1.4 Statistical Methods 20

2.2.1.5 Other Segmentation Methods 24

2.2.2 Video Block Techniques *26*

2.3 CHAPTER SUMMARY 27

CHAPTER 3 METHOD COMPARISONS 29

3.1 MIXTURE OF GAUSSIANS 29

3.2 PROBABILISTIC SKIN FILTERS 32

3.3 CONTINUOUSLY ADAPTIVE MEAN-SHIFT 33

3.4 NEURAL NETWORK – MULTI-LAYERED PERCEPTRON 40

3.5 CHAPTER SUMMARY 44

CHAPTER 4 MOTION CAPTURE IN UNCONSTRAINED ENVIRONMENTS USING *J*-VALUE SEGMENTATION 45

4.1 NON-PARAMETRIC CLUSTERING OF IMAGES 46

4.2 *J*-VALUE SEGMENTATION 53

4.3 SOFT-CLASSIFICATION MAPS 57

4.4 JOINT-CRITERIA REGION MERGING 59

4.5 REGION TRACKING 62

4.5.1 Intra-Video Stack Tracking 63

4.5.2 Inter-Video Stack Tracking 64

4.6 CHAPTER SUMMARY 66

CHAPTER 5 EXPERIMENTATION 68

5.1 EXPERIMENTAL SETUP 68

5.1.1 Software Design 70

5.1.2 Test Environments 73

5.2 ALGORITHM ANALYSIS 77

5.2.1 Clustering Results 77

5.2.1.1 Impact of Locality-Sensitive Hashing Parameters 77

5.2.1.2 Impact of Adaptive Bandwidths 82

5.2.1.3 Comparing FAMS with K-Means Clustering 84

5.2.2 Soft-Classification Results 88

5.2.3 J-Value Segmentation Results 92

5.2.4 Merging Results *97*

5.2.5 Tracking Results *101*

5.3 EXPERIMENTAL RESULTS 105

5.3.1 Laboratory Environment Results *106*

5.3.2 Home Environment Results *109*

5.3.3 Studio Environment Results *113*

5.4 EXPERIMENTAL RESULTS IN OTHER CONTEXTS 118

5.5 CHAPTER SUMMARY 120

CHAPTER 6 CONCLUSION 121

6.1 SUMMARY 121

6.2 CONTRIBUTIONS 123

6.3 FUTURE WORK 124

REFERENCES 126

APPENDIX A K-MEANS CLUSTERING 135

APPENDIX B KERNEL DENSITY ESTIMATION 137

APPENDIX C EXAMPLES OF HOMOGENOUS COLOUR-TEXTURE MAPS 140

Table of Figures

FIGURE 2.1 - PER PIXEL REPRESENTATION OF WEIGHTED GAUSSIAN DISTRIBUTIONS 14

FIGURE 2.2 - GAUSSIAN REPRESENTATION OF SKIN COLOURED PIXELS 21

FIGURE 3.1 - MIXTURE OF GAUSSIANS PROGRESSION (FOUR FRAMES OVER A PERIOD OF 19 SECONDS) 31

FIGURE 3.2 - GAUSSIAN SKIN COLOUR REPRESENTATION 33

FIGURE 3.3 - SEGMENTATION USING CAMSHIFT APPROACH 36

FIGURE 3.4 - USE OF MULTIDIMENSIONAL RECEPTIVE FIELD HISTOGRAM FOR SEGMENTATION 37

FIGURE 3.5 - SEGMENTATION USING MODIFIED CAMSHIFT WITH RECEPTIVE FIELD HISTOGRAMS 39

FIGURE 3.6 - NEURAL NETWORK TRAINING DATA EXAMPLES 41

FIGURE 3.7 - NEURAL NETWORK BASED SEGMENTATION RESULTS BASED ON SEVERAL DIFFERENT INPUT VECTORS 43

FIGURE 4.1 - LOCALITY-SENSITIVE HASHING FOR TWO DIMENSIONAL DATA 52

FIGURE 4.2 - SAMPLE REGION MERGING PROCESS 62

FIGURE 4.3 - VIDEO STACK TRACKING 64

FIGURE 5.1 - INFRASTRUCTURE DESIGN 69

FIGURE 5.2 - INFRASTRUCTURE CAMERA SETUP 69

FIGURE 5.3 - MOTION CAPTURE INTERFACE 72

FIGURE 5.4 - MOTION CAPTURE CONFIGURATION INTERFACE 73

FIGURE 5.5 - LABORATORY ENVIRONMENT EXAMPLES 74

FIGURE 5.6 - HOME ENVIRONMENT EXAMPLES 75

FIGURE 5.7 - STUDIO ENVIRONMENT EXAMPLES 76

FIGURE 5.8 - IMPACT OF K AND L PARAMETERS IN LOCALITY-SENSITIVE HASHING 79

FIGURE 5.9 - FAMS AND K-MEANS CLUSTERING COMPARISON IN LABORATORY ENVIRONMENT 85

FIGURE 5.10 - FAMS AND K-MEANS CLUSTERING COMPARISON IN HOME ENVIRONMENT 87

FIGURE 5.11 - FAMS AND K-MEANS CLUSTERING IN STUDIO ENVIRONMENT 88

FIGURE 5.12 - EFFECT OF HISTOGRAM SUB-SAMPLING ON THE PROBABILISTIC REPRESENTATION OF CLUSTERS 89

FIGURE 5.13 - IMPACT OF SOFT-CLASSIFICATION MAPS ON *J*-VALUE COMPUTATIONS 91
FIGURE 5.14 - SEGMENTATION RESULTS IN LABORATORY ENVIRONMENT 94
FIGURE 5.15 - SEGMENTATION RESULTS IN HOME ENVIRONMENT 95
FIGURE 5.16 - SEGMENTATION RESULTS IN STUDIO ENVIRONMENT 96
FIGURE 5.17 - IMPACT OF WEIGHT SELECTION ON MERGING PROCESS 98
FIGURE 5.18 - MERGING RESULT COMPARISON 100
FIGURE 5.19 - INTRA-BLOCK TRACKING RESULTS 102
FIGURE 5.20 - INTRA-BLOCK TRACKING RESULT COMPARISON 102
FIGURE 5.21 - INTER-VIDEO STACK TRACKING IN LOW MOTION SCENE 104
FIGURE 5.22 - INTER-VIDEO STACK TRACKING IN MEDIUM MOTION SCENE 105
FIGURE 5.23 - FINAL LABORATORY SEGMENTATION RESULTS 107
FIGURE 5.24 - FINAL LABORATORY MOTION CAPTURE RESULTS 108
FIGURE 5.25 - FINAL HOME SEGMENTATION RESULTS 111
FIGURE 5.26 - FINAL HOME MOTION CAPTURE RESULTS 112
FIGURE 5.27 - FINAL STUDIO SEGMENTATION RESULTS 115
FIGURE 5.28 - FINAL STUDIO MOTION CAPTURE RESULTS 117
FIGURE 5.29 - RESULTS FROM OUT OF CONTEXT VIDEO 119
FIGURE B.1 - HISTOGRAM REPRESENTATION OF DATA DISTRIBUTIONS 138
FIGURE B.2 - DATA REPRESENTATION USING A DISCONTINUOUS KERNEL ESTIMATOR 139
FIGURE C.1 - EXAMPLE OF *J*-VALUE COMPUTATIONS FOR VARIOUS CLASS MAPS 141
FIGURE C.2 - *J*-VALUE KERNEL MASKS 142
FIGURE C.3 - *J*-VALUE REPRESENTATION OF A SIMPLIFIED IMAGE 142

Table of Tables

TABLE 5.1 - CLUSTERING RESULTS USING OPTIMAL (K, L) PAIR 80

TABLE 5.2 - CLUSTERING RESULTS USING SUBOPTIMAL (K, L) PAIR 81

TABLE 5.3 - ADAPTIVE BANDWIDTH COMPUTATION TIMES 83

TABLE 5.4 - CLUSTERING RESULTS USING FIXED BANDWIDTHS 84

Chapter 1 Introduction

The following chapter introduces the work and research done within this thesis on the motion capture of human targets within unconstrained environments. The context in which this research has been undertaken gives perspective on the importance this work has on the pedagogical and the computer vision communities. The motivations and challenges associated to this work are also described and pave the way to the introduction of a newly developed framework which attempts to address the issue at hand.

1.1 Context

Despite recent advancements in information technology, very few techniques have been proposed that would allow for the robust capture of motion involved in human performance without imposing constraints on the environments in which they are executed. While several commercial and academic research tools have dealt with the problem of motion capture, often called MoCap, few have been able to provide the necessary information for a complete evaluation and comparison of gestures involved within a performance. When taken in the context of athletics, ergonomics or musical performance, a tool such as this could potentially provide the means with which chronic stress injuries could be observed, analysed and corrected before they became a serious problem. The application of this technique can be extended into the field of pedagogy to provide quantitative measurements on performance and allow observation of the evolution of habits and practices involved within the learning process.

Current techniques involving motion capture rely heavily on active sensor technologies in order to record gestures executed by performers. These sensors are often encumbering devices that inhibit the natural movement of performers and thus compromise the integrity and accuracy of the very motions they are acquiring. The environments in which these sensors can operate must also be controlled or may introduce error in the acquisition process. These technologies are typically costly and tedious to setup, making their use limited at best. This research uses the term markerless [1] in order to indicate a technique that does not require any form of physical apparatus or limiting visual cue to perform motion capture.

The research introduced by this thesis stems from a multi-disciplinary effort that aims to bring together professionals from the fields of information technology and piano pedagogy in order to advance current motion capture technology by means of passive techniques. Ultimately the goal of this research is to provide the means with which a musician's movements can be captured without any direct interaction with the individual or his environment. This would allow pianists to perform in day-to-day environments, without the need to wear sensors or markers, thus allowing their performance to be uncompromised. These day-to-day environments, referred to as unconstrained environments within this work, are free of any type of manipulations that may have been done in order to simplify the motion capture process.

Within the field of professional pianists, the impact of injuries related to posturing and motion is quite severe. The current injury rates for professional adult pianists vary from 39% to 47%; for students the rate is lower, but still significant at 17% [2]. These injuries often result in professionals having to seek medical treatment and incur a lost of

income, often compensated by a shift in career focus. Medical costs associated to the type of musculoskeletal disorders described here can climb up to several thousand dollars per year for an individual. With an estimated 100 000 students graduating from musical schools within the United States and Canada every year [2], the need to provide music professionals with better injury prevention instruments is obvious.

1.2 Motivations

Recently there has been significant advancement in the field of computer vision techniques. However, none have yet addressed the complex problem faced here without having to impose unreasonable constraints upon musicians or athletes and their environments. Many techniques in the field of motion capture using passive sensors still rely on contrasting backgrounds or even assumptions on the motion and complexity of the scene. These impositions yield an environment that is foreign to a performer and can potentially compromise the integrity of his actions, leading the performer to behave differently than he would in a more comfortable environment. The limitations of such techniques may also obfuscate key performance markers through the application of arbitrary data representations or manipulations.

Deng *et al.* [3] have proposed a segmentation technique which not only relies on colour information but also on texture data in order to cut a scene into semantic regions. While the technique was proven efficient for the segmentation of real world scenes, it suffers from several shortcomings and often imposes assumptions on data prior to manipulations. Their technique is also known to over-segment and, while the authors have proposed a colour histogram based merging process, it fails to take into account

potential and yet important edges between regions. This thesis uses this technique as a starting point, improves upon the shortcomings of the original ideas and applies the resulting scheme to the context of human segmentation in unconstrained environments.

1.3 Challenges

The goal of this framework is to allow performers to conduct their activity in an environment that is familiar to them. In the context of piano playing the described environments could be classrooms, concert halls or even home studios. Identification or tracking markers also need to be avoided since they may inhibit natural movement; these include sensor devices or constraints on attire. The chosen process should even go as far as allowing single target segmentation when the performance is being assisted by another person. This is particularly relevant in pedagogical contexts where a student may be performing alongside an instructor.

Severe algorithmic challenges are also present and need to be addressed. Since the motion capture is strictly based on visual representation, the chosen algorithm must allow for a certain level of robustness whenever scene changes occur. These changes may be subtle lighting changes, shadowing effects, sudden movements or even lack of movements. The list of visual challenges goes on and while the work done here is by no means a panacea, it should attempt at handling some of the major difficulties often encountered for this context.

1.4 Objectives

This work attempts to fulfill the following set of objectives for the segmentation of targets within unconstrained environments using passive computer vision and pattern recognition techniques:

1) Develop a process that does not, by any means, interfere with the performance or natural behaviour of the targets. The process should allow humans to perform in day-to-day environments uninhibited by outside influences.

2) Use techniques that manipulate visual data with a minimal set of assumptions with regards to their representation or distribution. Parametric models or specific data representations may lead to incorrect performance evaluations brought on by faulty assumptions or misrepresentations.

3) Identify and track human body parts and key motions exhibited throughout a performance using visual data only.

1.5 Proposed Framework

The goal of this research is to provide the foundation framework to a more complete motion capture system that does not interfere with the movement of a target. A purely passive system should rely strictly on visual data in order to capture motion. A multi-camera setup with sophisticated calibration algorithms provide the means with

which three-dimensional data can be computed. However, the first step in motion capture for the performance evaluation of a musician is the identification of the target from within the scene. This type of visual identification is commonly known as segmentation and is used here as the basis for the proposed framework. A segmentation algorithm is used in conjunction with several other techniques to allow not only the identification of the target but its tracking throughout a performance.

The following thesis presents a framework capable of region-based segmentation of targets within complex environments while imposing a minimum of constraints. The framework can be described in terms of five key modules that allow the creation of semantically significant regions that, when amalgamated in a meaningful way, allow for the proper segmentation of targets within a video or image content. The first module uses a non-parametric clustering algorithm in order to perform an initial analysis of colours. The module yields a classification for each pixel colour present within the media content. This classification for every pixel can also be interpreted as a colour-texture representation of the media. The second module computes the set of membership values each pixel has with the various classes. By taking into consideration the fact that a pixel colour cannot always be strictly classified within a specific colour group, the framework is able to adapt its colour-texture representation to take into account gradient colours. The third module performs an analysis on the final colour-texture representation in order to extract homogenous regions. These regions are identified based on an iterative and hierarchical methodology and are said to be homogeneous in the sense that they are local regions having a consistent colour-texture pattern. The fourth module attempts to merge similar adjacent regions based on colour and edge criteria. The merging process is used

to avoid potential over-segmentation problems. The fifth and final module tracks the regions produced between media sections. The tracking process takes advantage of the high frame rate of video capture devices in order to identify correspondences based on overlap. The segmentation process is performed on an overall image or frame and produces regions for the entire media content. The onus of initially identifying which regions are parts of a surveyed target is put on a human operator. The identification requires intimate knowledge of key performance markers for any given activity. Once the identification is made, it can be maintained throughout a sequence by means of the presented framework.

The framework presented here is applied to the context of human target segmentation in real world scenes; however it can be generalized for a single image or even multi-dimensional data sets. The process is intended for unconstrained environments and uses a minimal set of assumptions. Sensor quality and scene representation may affect the outcome of the segmentation in a number of different ways.

1.6 Organization

This thesis is organized into six chapters. Chapter One introduces the work and research done for this project. Chapter Two reviews some of the important and most recent techniques relevant to the field of motion capture and segmentation. Chapter Three proposes an experimental comparison of some of the techniques discussed in the literature and puts a focus on their shortcomings within unconstrained environments and for the context presented here. Chapter Four presents an in-depth description of the proposed segmentation framework. It highlights the major procedures involved in the

segmentation and tracking as well as describes how the technique has been modified and enhanced from the original algorithm proposed by Deng *et al.* [3]. Chapter Five discusses the experimental setup of the framework and its performance. Chapter Six provides conclusions and a look at future work.

Chapter 2 Literature Review

Motion capture is a complex task often achieved through the use of sophisticated tracking sensors and environmental setups. The first portion of this chapter will give a short review on some of the traditional motion capture approaches. This review will cover active sensing techniques and highlight the many shortcomings and constraints these techniques must impose in order to succeed. The second portion of this chapter will strictly review passive techniques and in particular focus on segmentation algorithms that can be used to achieve the motion capture.

2.1 Traditional Motion Capture Techniques

As previously mentioned, many of the current motion capture techniques used today by professionals require the use of encumbering sensor devices or impractical environmental setups. The two most popular methods of active sensing include the use of magnetic and optical markers. Each of these markers has its own advantages and disadvantages, their common property being that they must be secured onto the moving target. These approaches use an active sensing methodology in order to extract three dimensional positional data regarding the human target.

The magnetic trackers can be used to capture motion by projecting magnetic fields. These fields are measured by a stationary sensor able to interpret the signals into positional information. The size of the actual markers makes it difficult to acquire a large number of data points and incur a lower resolution motion capture. These sensors must

also rely on cabling in order to transmit information and further inhibit natural motion. Typical magnetic systems operate at a frequency of approximately 100 Hz and are mostly appropriate for larger movements [4]. Several issues with regards to measurement errors are present with this type of capture [5]. Cross-talk between sensors, influence from external magnetic fields and marker design [6] can all contribute to a loss in precision. This type of motion capture is best suited for large general movements and recovery of three-dimensional data.

In the case of optical trackers, motion capture is performed through the segmentation of key colour or optical markers placed upon a target. Marker size can be quite small and allow a very fine resolution of motion capture. Typical systems operate at a frequency of approximately 240 Hz thus allowing very fine movements to be observed [4]. Due to the large number of sensors that have to be positioned in order to acquire small movements, this approach usually interferes with the target's performance. The VICON [7] system, popular in medical applications and gait analysis is a perfect example of an optical based motion capture system. This system uses specialized reflective orbs that are placed on the target; cameras sensitive to the reflectance emitted by these orbs are used in order to acquire the data. With the help of a multi-camera setup and a calibration procedure, three dimensional positional data can be reconstructed. Other systems such as the one proposed by Drouin *et al.* [8] do not require specialized materials in order to acquire data. In their technique, brightly coloured balls are secured onto the target and traditional segmentation algorithms are used. This however requires that the targets wear very dark clothing in order to create a contrast with these markers.

The motion capture techniques reviewed here provide full three dimensional positional data at the cost of comfort and simplicity. While their ability to correctly acquire movement data is not an issue, their capacity for non-interfering with a performance is questionable. The use of either magnetic or optical markers severely restricts the natural movement of a performer and cannot be used for the context considered here.

2.2 Motion Capture Segmentation Techniques

Segmentation is the foundation to many computer vision applications and it has been explored for many years and in multiple contexts. As such, there are numerous methodologies available and this section will give an overview of those most pertinent to the field of motion capture. The techniques presented here only acquire motion data in two dimensions. A multi-camera setup with sophisticated calibration and reconstruction algorithms can be used in order to obtain the full three dimensional information. The proposed techniques are broken down into the following categories: background modelling, contour-based, region-based, statistical methods, and others. The classification of segmentation methodologies is difficult due to the fact that many of them have been devised for very specific applications and do not necessarily correspond to one specific category. The review also makes the distinction between sequential image and video block segmentation techniques.

2.2.1 Sequential Image Techniques

The field of sequential image techniques has been explored thoroughly for a variety of applications. These techniques achieve segmentation on a single image or frame at a given time. In the case of a video, the frames would be segmented individually and in sequence. It should be noted that despite the method's sequential nature, *a priori* information can be gathered in order to improve the results as the sequence progresses. Computational simplicity and minimal memory requirements are key advantages to these techniques.

2.2.1.1 Background Modelling Methods

The first set of methods reviewed stems from the classical background differentiation and threshold approaches to segmentation. Background differentiation implies that the objects that compose a scene can be separated into two categories: objects of interest or those that belong to the foreground and trivial non-important objects belonging to the background. This separation is achieved by modelling the background component. The modelling process will typically define the background either by using a parametric representation or an adaptive image created using some initial assumptions. The creation of a parametric criterion, or threshold, can be done using a trial-and-error methodology. An alternate method of creating this threshold would be through the acquisition and analysis of current scene data.

One of the more established threshold approaches was proposed by Otsu [9]. The technique attempts to identify the sections of a histogram that belong to the foreground and background. These histograms are constructed using grey-level images under the

assumption that these images have a distinct bimodal distribution. This technique could equally be categorized as a statistical method since it performs an analysis on the image histograms in order to determine how best to segment colour. Due to its strict foreground and background classification however, it is more appropriately described in this section. Unfortunately this approach is dated and has several key limitations. Not only does the use of grey-level images severely impede the distinctiveness of objects contained within some image sections, but the assumption that an image histogram would follow a bimodal distribution is inappropriate for complex scenes where a vast number of colours and textures are present.

In the case of background modeling techniques one of the most successful algorithms was introduced by Stauffer and Grimson [10]. In their technique the authors fully acknowledge the reality that foreground representation cannot entirely be satisfied using a parametric methodology and that the chosen algorithm must also account for the dynamic nature of colours within video sequences. Instead of opting for the global model of a background, the authors chose to model the colour behaviour for each individual pixel. The model is created using Gaussian statistics in which the mean and variance of a distribution is determined by a pixel's mean colour and variation through time. Each pixel can be modelled using several Gaussian distributions depicting its various colour behaviours as seen in Figure 2.1. As new images are interpreted, the pixel values are analyzed against current Gaussian distributions to identify which best describes the current behaviour. Once identification is made, the distributions in question are updated using an Expectation-Maximization (EM) algorithm. If a set of image pixels have re-occurring colour patterns, the Gaussian distributions for these patterns will gain

importance in the form of a weight indicator. Distributions having sufficient weight are considered redundant colours for a given pixel and are assumed to indicate a background colour.

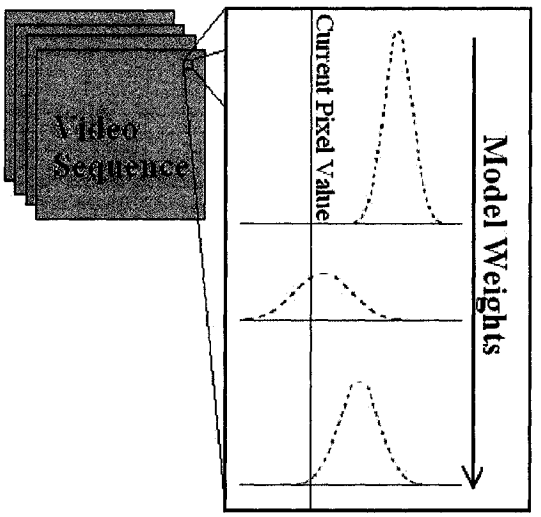


Figure 2.1 - Per Pixel Representation of Weighted Gaussian Distributions

Wren *et al.* [11] have shown that this kind of background modelling is feasible for the tracking of human bodies in scenes having good to ideal lighting conditions, low temporal noise and well defined motion patterns. Several authors have proposed variations and improvements to this type of background modelling. Horprasert *et al.* [12] introduced a variant to this technique where shadows are identified in order to reduce the number of false-positive segmented pixels. The detection takes into account both the chromatic and brightness aspects of a pixel's colour; if the brightness is reduced but the chromaticity remains relatively similar, the pixel is considered shadow and is not falsely segmented. In the case of Atev *et al.* [13] the technique is modified to take into account

sudden brightness and contrast changes. These modifications, while novel in concept, are mostly suited to applications relating to traffic monitoring.

The use of Mixture of Gaussians for background modelling has several shortcomings for the type of application considered in this research work. Its capacity to learn background and foreground models relies heavily on the motion of foreground objects. When tracking musicians, movements performed by the subject can be quite subtle causing the colours that make up each individual pixel to change rather infrequently and would ultimately be classified as a background component. This problem represents a serious challenge in the application of the technique. The next difficulty in applying this algorithm stems from its initialization procedure. While it is not necessary to initialize the system without any foreground objects, not doing so would mean that the modelling time of the background would be increased significantly. Newly revealed background sections would register as a foreground object since the system would not have spent the required amount of time learning the particular distributions of that section. More on the shortcomings of this type of technique will be discussed in Chapter 3.

2.2.1.2 Contour-Based Methods

An important class of segmentation techniques includes those that rely on image edge information in order to delineate objects. One of the most popular edge-based techniques are Snakes (or active contours), introduced by Kass *et al.* [14]. The goal of this technique is to deform a contour so that it matches the boundary of a given object. The deformation of the contour is driven by an iterative energy minimization procedure

that allows contour curves to converge on a target's edges. The energy function involved is designed in such a way that its local minima are achieved when the contour corresponds to the bounding edges of the object being examined.

The technique has also been extended for the segmentation of video objects in [15]. In this case, the contours are projected onto subsequent frames using a rigid body motion estimation process after which they are re-adapted to the edge information of the object. The computational complexity is quite high and as is indicated by the authors, the technique is not well suited for large non-rigid movements.

Active contours are typically not appropriate for situations where objects may be partially occluded or where reliable edge-information is difficult to obtain. The former problem was resolved by Peterfreund [16] with the introduction of Kalman Snakes. Using a combination of optical flow measurements along with a Kalman filter, the contours were made resilient to partial occlusion effects. In the case of unreliable edge-information, Sun *et al.* [17] proposed the use of a Veterbi search algorithm in order to find the best possible positioning of key points along the boundary curve. For the complex scenes used in this research as well as the potential for a large dynamic range of motions, contour-based methods are not ideal.

2.2.1.3 Region-Based Methods

Another popular approach to image segmentation consists in dividing an image into coherent regions that could be used to represent a given object. These techniques perform an analysis of the data space in order to produce a simplified grouped representation of the data. The union of these regions makes the process of segmentation

and tracking much simpler. The means by which an algorithm decides to group data varies from technique to technique.

Vincent and Soille [18] introduced a technique based on an immersion process analogy that computes watersheds in greyscale images. The authors simulated the flooding of water within an image in order to produce so-called water basins corresponding to local image minima. Watersheds are defined as the boundaries between these minima. The method is efficient enough that it has become the foundation to multiple popular segmentation techniques [19]-[23]. Wang [19] uses the watershed technique for a video segmentation algorithm where an initial region partition of the video frames is obtained based on a multi-scale gradient image. In other words, the watershed algorithm is driven by image edges. To produce a more coherent partitioning throughout the video, Wang uses a motion based merging process to bring together similar regions. Motion estimation is also used to project these regions into subsequent frames for the purpose of finding correspondences between regions. In the case of Shien *et al.* [20] the region tracking process is driven by modifying current regions thereby speeding up the process since it no longer necessitates the computation of the watershed algorithm in later video frames. Tsai *et al.* [21] introduce the concept of 3D watershed volumes. These volumes are generated by amalgamating several regions together which are found to correspond between image frames. The spatiotemporal relationship between these volumes is computed in order to merge together similar volumes and extract pertinent video objects.

An important issue occurring with the use of the watershed technique is that it is prone to severe over-segmentation. The use of gradient data in order to produce regions

in complex images yields a plethora of regions that would otherwise be merged together. In Haris *et al.* [22], this issue is addressed with the introduction of a Region Adjacency Graph (RAG). RAG is the means by which the region merging process can be represented. A graph is constructed where its nodes are represented by the image regions and its vertices are linking together adjacent regions. The traversal cost of a vertex is given by a merging metric; typically a difference in region features such as colour. The minimum cost edge within the graph is found and used to merge together similar regions. Once the merge is completed, the graph is updated to reflect the new segmentation configuration. The process results in an iterative means in which similar regions can be combined in order to obtain a final segmentation. Hernandez *et al.* [23] take this method one step further by introducing a joint region merging criterion. The criterion combines both colour and edge information to produce more relevant graph vertices, thereby combining similarly coloured areas while maintaining edge integrity throughout the process.

Other clustering-based methods such as that of Chen *et al.* [24] opt to produce regions based on the similarity of pixel properties. In their technique the authors group together pixels based on an algorithm derived from a modified *k*-means (see Appendix A). The original modification was proposed by Pappas [25]. The authors acknowledge the lack of spatial constraint in the application of the *k*-means and address this issue through the use of a Gibbs random field model. This Adaptive Clustering Algorithm (ACA) allows for the grouping of pixels based not only on their colour similarity but also on their connectivity. Following a texture analysis based on predefined filters a final segmentation is derived. The advantage of this technique is the fact that the clustering

algorithm allows for gradient colours. The disadvantage however is that if texture regions cannot be well represented or interpreted through the selected filters, the method may fail in properly producing accurate colour-texture regions.

Another technique used in the segmentation of colour-texture regions is proposed by Deng *et al.* [3]. This technique, named JSEG by its authors, constitutes the basis for the framework proposed within this thesis. In their work, image data undergoes an initial clustering based on their Peer Group Filtering method [26]. This clustering methodology is very similar to the *k*-means but is preceded by a filtering process that blurs pixel colours based on Gaussian statistics while trying to conserve edge integrity and remove impulsive noise. No attempt is made to spatially constrain the clustering process. Instead the authors take advantage of the spread nature of the clusters and choose to interpret this as its own colour-texture representation of the image. The colour-texture data is interpreted in order to measure points of local homogeneity. This homogeneity is represented by what the authors call the *J*-value, a measure of spatial cluster distribution. The set of *J*-values for an image in turn becomes a gradient representation of not only edge information, but also of colour-texture boundaries. Using a novel seed growing algorithm, regions can be produced based on this gradient representation.

Despite the fact that JSEG was shown to be useful in fields of segmentation other than real life scenes, such as satellite imagery [27], several points of improvements were brought up by Wang *et al.* [28]. Their first observation was the fact that the clustering method used in the technique was still strongly based on the *k*-means algorithm which heavily relies on a parametric description of the data. In the case of real world scenes, colours are not always known to follow specific statistical distributions. For this reason

Wang *et al.* propose the use of a non-parametric clustering technique called the Fast Adaptive Mean-Shift initially introduced by Georgescu *et al.* [29]. Another suggested improvement was the modified representation of *J*-values that the authors called *Soft J*-Values. Clustered pixels may have properties akin to more than just a single cluster group; in fact, a membership value of a given pixel can be computed for each of the cluster groups. Wang *et al.* produce a set of *Soft J*-Values that are a weighted sum of these membership computations. This new weighted clustering allows for *J*-values to have smoother transitions between regions connected with gradient colours. This resolves many of JSEG's most important shortcomings.

2.2.1.4 Statistical Methods

Many techniques rely on the inherent statistical properties of the information found within an image in order to accurately segment objects of interest. Similar to the background modelling techniques introduced earlier, the exception with these techniques is that they attempt to model actual objects rather than background-foreground relationships.

Colour is an important and powerful cue in identifying objects, the advantage of using colour characteristics stems from the fact that they are highly invariant to most transformations [30]. That is, whether an object rotates, translates or deforms in some way, the colours that comprise it remain approximately the same. Many techniques have taken advantage of this fact and have used colour in the segmentation of human skin in order to segment faces and hands [30]-[34]. While invariant to most movements, colours are highly susceptible to lighting conditions; shadows, sensor errors, light colour

temperature and directionality of the light source, all contribute in the way colours are represented [32]. Yang *et al.* [33] have observed that despite environmental conditions, skin colours have a tendency to cluster within the RGB colour space. Specifically the clustering effect can be modeled using a Gaussian distribution (see Figure 2.2).

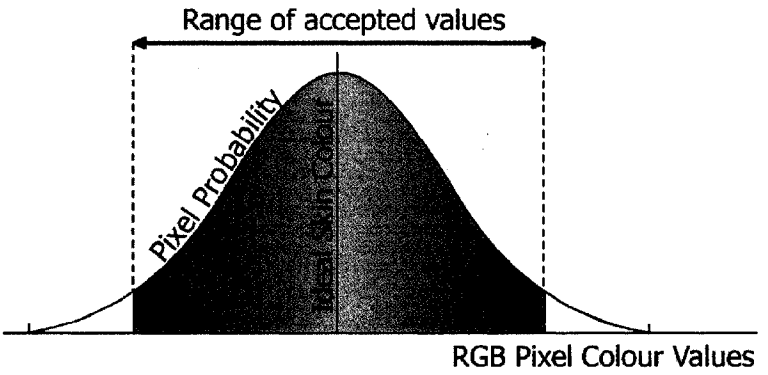


Figure 2.2 - Gaussian Representation of Skin Coloured Pixels

To mitigate the effects of illumination on the colour distributions, a normalization of the RGB tri-stimulus values can be done. Du *et al.* [34] have defined a 2D Gaussian probability function representing the likelihood of a pixel belonging to skin. A threshold is applied to the likelihood value to obtain an appropriate representation of skin patches within an image. This technique is called a Probabilistic Skin Filter. The challenge with this technique is in the initialization of the Gaussian distribution model. While the model can be created using *a priori* knowledge of skin colour, it is still highly dependent on the actual skin representation within a sequence. The extension of this type of segmentation to other colours comprising a target is questionable. While many have assumed that

colours will exhibit a Gaussian-like distribution within an image, this criterion is not sufficiently discriminative for proper segmentation.

Comaniciu and Meer [35] introduced a new method of analysing colour distributions based on a non-parametric representation. Their so-called Mean-Shift analysis iteratively converges a local data window onto the nearest suitable distribution mode. The convergence is achieved through the estimation of the density gradient which, in turn, provides a vector pointing toward the direction of the highest distribution concentration. A non-parametric interpretation such as this allows colours to have any type of distribution and does not impose any constraint to their behaviour. The authors have also shown how this type of analysis can lead to a very good segmentation of images where colour regions have sufficiently contrasting edges. Several improvements have been introduced to the analysis process to allow for variable bandwidth windows during the density gradient estimation process as well as optimizing the overall performance for high dimensional data [29]. In Bradski [36], Mean-Shift was used in the context of a perceptual user interface. Human face segmentation, position and orientation were provided using a Mean-Shift convergence where distribution modes were computed based on image moments within local windows. Since this modification allowed the Mean-Shift procedure to re-adapt its density gradient estimation for every new frame, the technique was called the Continuously Adaptive Mean-Shift (CAMSHIFT). The CAMSHIFT technique was later extended by Allen *et al.* [37] to allow the segmentation to work for an arbitrary number and types of feature spaces. In their improvement, the authors used a histogram back-projection technique [38] to estimate a feature space's

density gradient. Despite its efficient design, the Mean-Shift and CAMSHIFT techniques are susceptible to the discriminative capabilities of colour analyses or lack thereof.

Finally, the use of histograms is also an important and classical statistical approach to segmentation. They offer an invariant, non-parametric representation of the colours of an image region. Swain and Ballard [38] have shown how histograms can be used to differentiate objects among different viewpoints. They also introduce the concept of histogram back-projection that allows for a probabilistic representation of pixel based on histogram data. More recently histograms have grown to encompass other types of data than just colour. Schiele and Crowley [39] propose the use of multidimensional receptive field histograms. These data representations are based on the local response of an image to various operators called receptive fields. An example of such a neighbourhood operator includes the use of a Laplacian or gradient operator, the response to a Gabor filter and the use of Gaussian derivatives. By amalgamating these responses into a multidimensional data representation the authors claim to have a highly discriminatory description of an image or object. A discussion is provided within [39] on the different comparison methods available to test these discriminatory descriptions. For real scene images however, where prior data is unavailable, the selection of receptive fields is made ambiguous. Improper selection can yield poor segmentation results or divergence whenever image characteristics evolve. Despite this, Pelisson *et al.* [40] show the potential of the overall process by identifying advertisements in sport sequences.

2.2.1.5 Other Segmentation Methods

As mentioned before the categorization of segmentation techniques is made difficult due to the number of novel and specialized methods. This section tries to give notice to some of the more influential approaches that cannot clearly be associated with any of the previous categories. In particular, it looks at methods based on neural networks and those relying on feature identification and meshing.

Over the years, several techniques [41]-[45] have proposed the use of neural networks for assisting in the segmentation process. Neural networks employ a large number of interconnected processing nodes that perform simple computations. Neural networks aim to imitate the biological reasoning capabilities of human beings. Their ability to learn and generalize patterns makes them powerful classifiers. Several authors [41]-[43] have proposed the use of Multi-Layered Perceptrons (MLPs) in order to classify image pixels into appropriate segmentation classes (typically foreground and background). In each case the networks are trained using a set of pre-segmented images that may or may not contain the object of interest. The major variations between techniques involve the choice of image features fed to the input layer of the network. McCrae *et al.* [41] use a simple three-dimensional RGB vector for classification, while [42] and [43] suggest the use of more comprehensive input vectors of 9 and 31 dimensions respectively. The network's ability to segment is highly dependent on the available training data and the possible transformations an image may undergo. The previously mentioned neural network classifiers are not resilient to any kind of environmental changes. The network configurations remain static and do not adapt to changes in data representation. Only recently have a few papers proposed adding an

adaptability mechanism to neural networks [44][45]. These techniques rely on complex retraining algorithms in order to modify the network in response to a decrease in performance. Along with the retraining of the network comes the difficulty in obtaining new training data and evaluating current segmentation performance.

In feature-based segmentation, discriminative key points of an image are identified and tracked through each video frame. The Scale-Invariant Feature Tracking (SIFT) technique introduced by Lowe [46] can achieve this goal quite successfully. There is also a set of techniques such as [47] which map a mesh to feature points in order to perform tracking based on feature inter-relationships. In the case of human performance evaluation, the set of features which are key indicators of an individual's performance may not be known or may be difficult to segment or track. This drawback makes feature-based approaches unsuitable for human performance evaluation without some kind of known relationship between performance indicative features and segmented features. In the case of Shi and Malik [48], the segmentation is interpreted as a graph partitioning problem. They introduce the normalized cut criterion which measures both the similarity and dissimilarity between groups within the graph. The optimal partitioning of this graph is shown to correspond to the solving of a generalized Eigen-value system. Graph construction however, relies on the creation of a set of weights between nodes. The authors suggest the use of a probabilistic interpretation of the differences in brightness or colour between two nodes. In complex real scene images however, colours may exhibit particular textures that should also be taken into consideration.

2.2.2 Video Block Techniques

The concept of processing a set of frames in blocks rather than consecutively has received an increasing amount of attention. The lack of popularity was, in part, due to the high computational cost of processing multiple frames at a time. However, with advances in the technology used to perform these operations, video block processing has quickly expanded into a viable option for future techniques. The obvious advantage of such techniques is the fact that they allow image data to be manipulated across multiple frames at once; temporal properties and behaviours can then be observed. Segmentation performed on multiple frames at once makes tracking obsolete since the identification is being performed dependently of nearby frames.

Allmen *et al.* [49] were among the first to introduce this type of multi-frame segmentation. Their algorithm relies on what they call dynamic perceptual organization groups. They compute and organize together sets of motion through a sequence of images in order to provide a higher level representation of objects in a scene. More recently Shi and Malik [50] have also introduced a method which achieves segmentation by grouping regions of similar motion. This grouping is obtained by extending their previous normalized cuts algorithm using graph weights obtained through a probabilistic representation of motion. In DeMenthon [51] however, motion is not the main grouping criterion. Video block segmentation is done using a modified Mean-Shift approach. The algorithm is applied to a 3D volume representation of a video sequence in order to group together 7-tuple feature vectors. These vectors include colour motion angle components. While proven to be quite effective, the technique is highly computationally expensive requiring a hierarchical approach in order to render it feasible.

Another interesting entry into these types of techniques consists of the modifications that Deng *et al.* [3] added to their JSEG algorithm in order to allow it to segment multiple frames in a batch process. While initially the technique does not lend itself very well to video block segmentation, its authors have introduced a supplementary temporal segmentation criterion named the J_t-value. This value represents a measure of the temporal disparity among pixel classifications between two subsequent frames stacked on top of each other. When seed determination is undergone, intersection with seeds in subsequent frames is taken into account to assure a temporal homogeneity. This determination takes into consideration the J_t-value in order to attenuate the influence of sections having high temporal disparity. This in fact allows the algorithm to produce a segmented video without the need for costly motion analysis.

2.3 Chapter Summary

In this chapter an overview of the contributions of several segmentation algorithms was presented. Particular distinctions were made among the algorithms that are performed on a sequential basis and those that are performed on a video by grouping several frames into blocks. The set of single image techniques were roughly categorized into background modelling, contour-based, region-based, statistical and other groupings. A majority of the algorithms, regardless of their categorization were found to make simplistic assumptions relating to the colour composition of the images. In the case of background modelling, techniques often relied on the necessity of motion while, in contour-based methods, the prominence of edges was the driving factor. Statistical methods relied on feature distinctiveness and region techniques seemed to have issues

with over-segmentation. Multi-image segmentation techniques gave insight on how the process could be achieved by looking at image properties through time. Many of the methods still relied on the pervasiveness of motion or made assumptions that limited their application for different contexts.

Chapter 3 Method Comparisons

Many of the techniques discussed in the literature review suffer from multiple key shortcomings in the context of motion capture in unconstrained environments. In order to better illustrate these limitations this chapter illustrates some sample results from trial implementations of these techniques. The algorithms chosen for this comparison are only a subset of the overall review but include the Mixture of Gaussian, Probabilistic Skin Filter, CAMSHIFT and Neural Networks techniques.

3.1 Mixture of Gaussians

The first technique presented in this comparison is the Mixture of Gaussian algorithm, first introduced by Stauffer and Grimson [10]. As explained within the literature review, the technique attempts to model individual pixel behaviours using a Gaussian representation. The pixels are given any number of Gaussian models that may adequately represent their colour patterns throughout a video. These models evolve based on the prevalence of pixel colours. The more frequently a colour appears for a given pixel, the more weight is attributed to the corresponding Gaussian model. This weight will determine if a model is identified as background or foreground.

In the trial implementation no consideration was given to the initial background. For a truly unconstrained segmentation to occur, restrictions such as static backgrounds and assumptions on scene composition cannot be made. The Mixture of Gaussian algorithm must therefore rely on its ability to learn background and foreground colour

behaviours in order to function. One of the immediate pitfalls in this type of application is introduced by the variable duration of a sequence. A performance could last anywhere between a few seconds to more than a couple of hours. Learning parameters for the proper modelling of colours and their behaviours must be carefully selected in order to avoid any foreground-background misrepresentations. A shorter sequence would in fact require a faster learning process that would permit background elements to quickly be identified. For longer sequences more time can be spent learning these background elements provided that the initial correctness of the segmentation is not crucial. Consequently, the faster the learning rate, the faster foreground objects must move in order to avoid having their own colour behaviours being identified as background.

Figure 3.1 clearly demonstrates the issues in background learning exhibited by the Mixture of Gaussian technique. The background elements that remain static are quickly identified through the sequence frames (depicted in black). However, in evaluating performance such as piano playing, some foreground elements also exhibit this same static behaviour, thus the algorithm interprets these elements as background. This phenomenon can be seen along the lower body of the pianist within Figure 3.1. The misrepresentation evolves as the sequence goes on to eventually encompass the majority of the pianist's lower body, torso and upper arm. Quicker moving elements such as the pianist's head and hands remain clearly visible. Initial occlusions in the scene also result in segmentation error. Occluded elements are always identified as foreground since their colour behaviours have yet to be observed. In a scene where prior background information is not available, no guarantee can be made that the occluded elements will have sufficient exposure to result in their integration to the background. This issue is

observed by the visible background wall surrounding the pianist; the wall becomes un-occluded as the pianist moves.

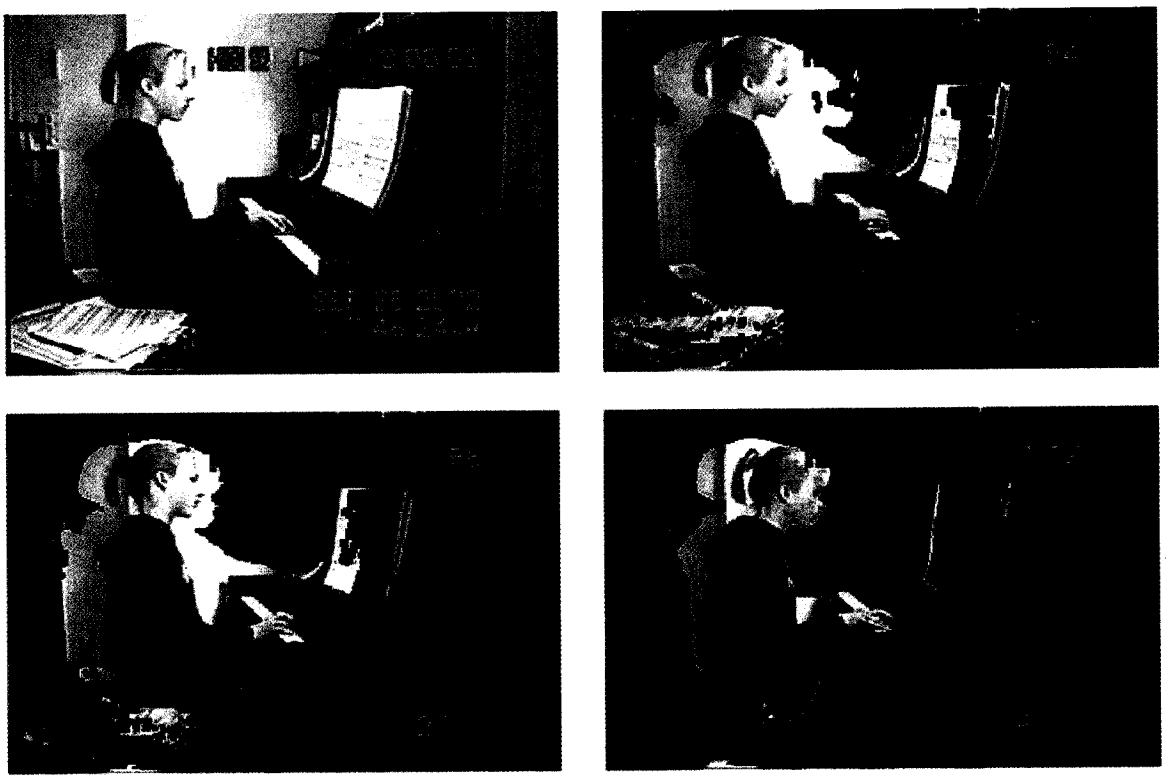


Figure 3.1 - Mixture of Gaussians Progression (four frames over a period of 19 seconds)

Other problems presented by the Mixture of Gaussians stem from its inability to handle subtle lighting changes, the introduction of new objects or the need to distinguish between two closely moving objects. In indoor environments there is a high potential for light changes to occur. If such a change does in fact happen the algorithm must spend time re-learning all the pixel colour behaviours under the new conditions. Some research has recently been introduced to mitigate this issue [11]-[13]. In the case where other superfluous moving objects may be present or introduced into the scene, the original algorithm cannot make a distinction between the key target and these new additions. The

ability to distinguish between moving objects requires an additional layer of complexity to the overall segmentation. In cases of performance monitoring, trainers, instructors or teachers may be present in the scene but are not objects of interest. While a powerful means of segmentation, the Mixture of Gaussian approach finds it uses within a confined set of applications. Its shortcomings make its application difficult at best for unconstrained environments.

3.2 Probabilistic Skin Filters

In recent research [30]-[34], multiple applications have used the fact that skin tones tend to cluster within the RGB colour space in order to segment human features. This colour property is said to follow Gaussian statistics [33]. By constructing an appropriate statistical model and applying a threshold to each pixel's membership to this model, identification of skin patches can be made. The problem with this approach is in the adequate construction of the Gaussian model and the computation of an appropriate threshold.

The trial implementation done within this work used skin colour samples extracted throughout the sequence in order to estimate proper variance and mean of a Gaussian model that would conform to the skin tone of the pianist. A membership threshold was computed through trial and error in order to obtain the best perceivable results throughout the sequence. As can be observed in Figure 3.2 much of the skin can clearly be segmented using this type of statistical analysis. Skin pixels are depicted in white in Figure 3.2. However it does not prevent other colours within the scene to overlap with these statistics, thus creating false identification of skin patches. The identification

is also heavily dependent on the manner in which the skin has been parameterized, in Figure 3.2 both the musician's hand and legs are left unidentified. While steps can be taken to reduce the amount of segmentation noise brought upon the application of the threshold, in cases such as these the complexity of the noise removal supersedes the statistical analysis. Ultimately this type of segmentation may only be appropriate in very precise situations where the object of interest is characterized by skin and found in somewhat less complex backgrounds.

Figure 3.2 - Gaussian Skin Colour Representation

3.3 Continuously Adaptive Mean-Shift

The techniques compared up to this section have relied on parametric description of both the spatial and temporal properties of image colour components. In the application of skin filters, research demonstrated that the use of Gaussian models was appropriate for that type of segmentation. However, the overlapping colour properties of the different elements within a scene make it difficult if not impossible to properly segment a complex target such as a human. Comaniciu has introduced a new method of

interpreting complex data such as image colour spaces using a non-parametric approach [35]. This methodology will be reviewed in-depth within Chapter 4. Building upon this analysis, more complex segmentation and tracking algorithms have been introduced. This section takes a closer look at the Continuously Adaptive Mean-Shift [36] algorithm as well as its application to the segmentation of human targets within unconstrained environments.

The goal of this implementation was to segment multiple user selected areas throughout a sequence. These areas were first defined by a human operator as a set of rectangular windows at the start of the sequence. A colour histogram is constructed for each area and acts as its descriptor for the subsequent frame in the sequence. When an area is to be segmented from a new frame, its histogram descriptor is back-projected [38] onto the image. Histogram back-projection involves replacing pixel colours with their normalized histogram bin value. An image having undergone a histogram back-projection will result in a probabilistic representation of its colours. The area's window is then iteratively converged from its previous position to its new found location. This iterative displacement is computed based on the center-of-gravity found within the window. Window dimensions and orientations are set using various image moment calculations.

In Figure 3.3 the progression of the CAMSHIFT algorithm can be seen. In the initial frame a user has selected to segment the pianist's general torso (depicted by the window superimposed in red). The subsequent frames demonstrate the results of a histogram back-projection and the final converged region of the window. In the case of the histogram back-projection, greyscale values are use to represent the histogram bin

values to which pixels belong. While a colour histogram is clearly insufficient to uniquely identify the region of interest, it does attribute the general area where the region was found with the highest probability. Despite its general correctness, the segmentation of these user specified regions quickly degenerates as colour properties of a region evolve throughout the sequence. As colour changes or as objects having similar colour properties collide, the window convergence process is skewed and may erroneously jump from one image element to another. In the sequence above, the initial area window jumps from the pianist's torso to the upper arm. This jump is caused by the fact that the upper arm adopts colour properties that are similar to the torso while the colour properties of this latter element are modified due to local lighting changes brought upon by motion. While the algorithm has the capacity to adapt to histogram colour changes over time, this adaptability is highly dependent on the slow progress of the change and the histogram's ability to uniquely identify a region of interest. These problems are only aggravated when smaller less distinguishable image elements require segmentation. In these cases, the use of a colour histogram does not provide a sufficiently unique descriptor and prevents any type of adaptation.

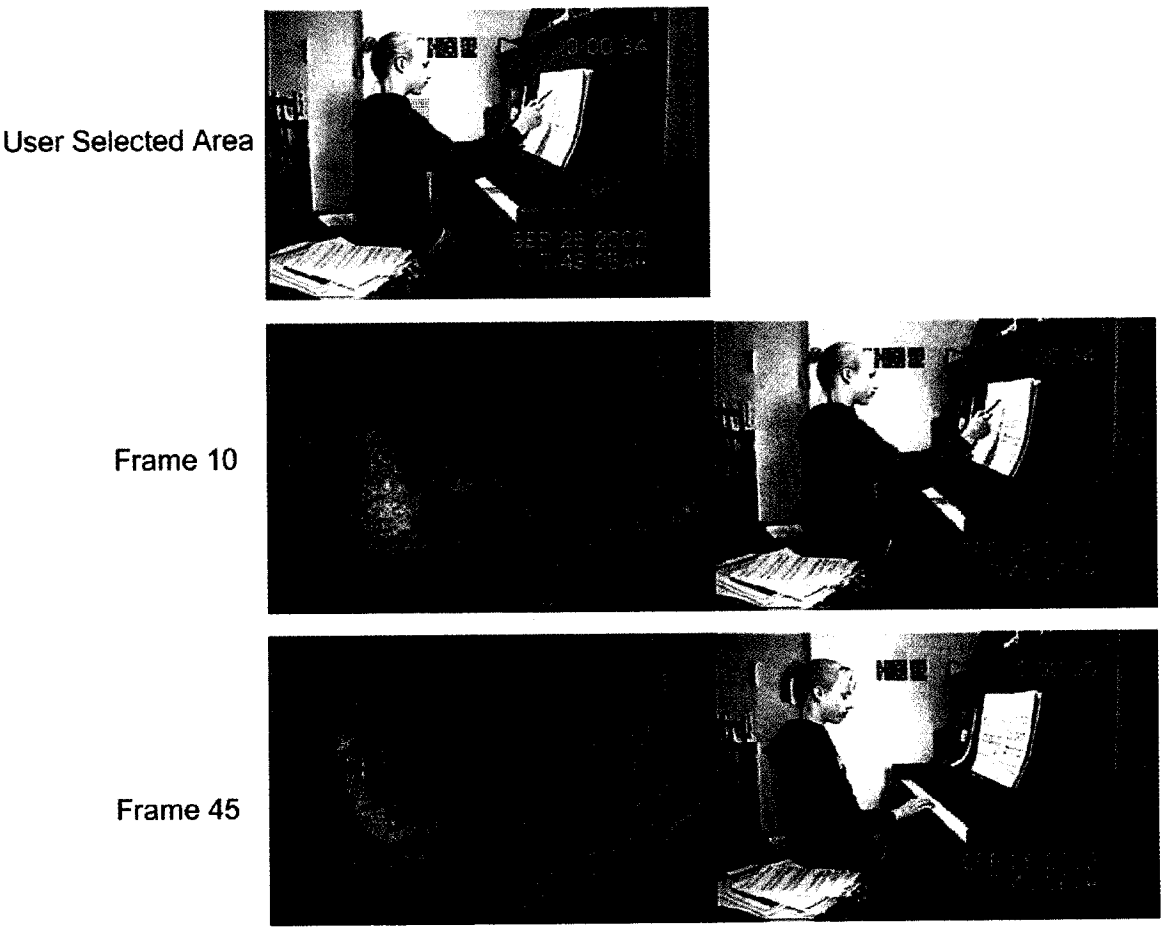


Figure 3.3 - Segmentation Using CAMSHIFT Approach

In an attempt to improve on the performance of the general algorithm, several measures were put into place to allow the construction of more descriptive histograms [52]. The construction of the histograms was extended using the concepts of Multidimensional Receptive Field Histograms (MRFH) [39]. By using the local responses of several filters a more descriptive representation of an area was achieved. In this extension of the original CAMSHIFT implementation, the MRFHs were created using a combination of colour space as well as gradient magnitude and Laplacian data computed at different scales. Figure 3.4 shows a slice of this multi-dimensional histogram and its probabilistic response when a back-projection is performed into the

originating scene. Once again the back-projection will result in a probabilistic representation of the image based on the histogram values. The histogram's ability to produce a discriminating description of an image element is uncanny. To improve upon these results, histogram construction was performed using a weight mask on the user selected regions [53].

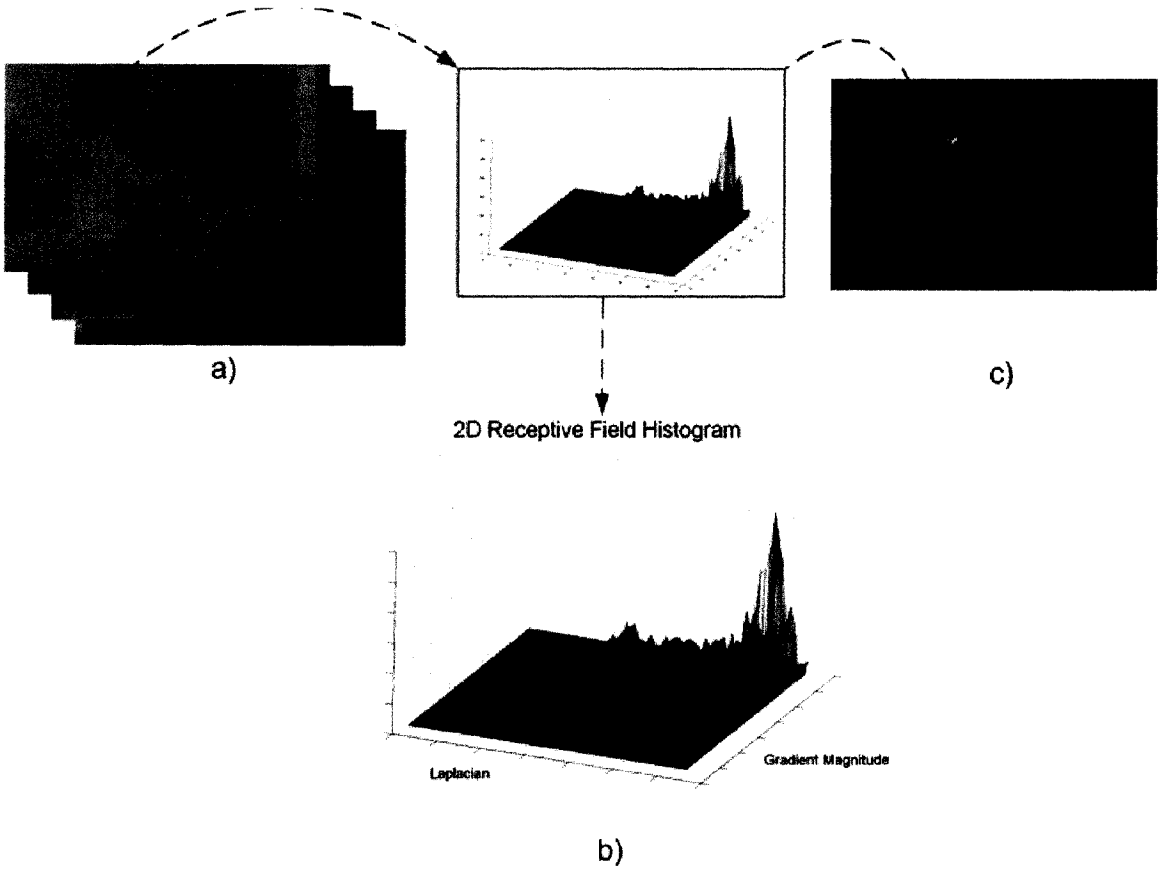


Figure 3.4 - Use of Multidimensional Receptive Field Histogram for Segmentation: a) Results of Several Receptive Fields, b) A 2D Slice of a Multidimensional Receptive Field Histogram, c) Results of Back-Projecting the MRFH

Since the regions being selected will most likely contain non-pertinent information, in particular around the boundaries of the selection, a Gaussian kernel is used to apply weights to the particular values being used in the histogram. The last improvement introduced within the original CAMSHIFT allows for the non-parametric

description of selected image elements. Geometrical bounding boxes can often poorly define an image region and may even introduce too many background elements to the histogram construction process thereby compromising data integrity. To reduce this compromising factor, bounding boxes are only used in the initialization process to define which image elements are of interest.

Once the histograms have been constructed and back-projected within a subsequent frame, the image element of interest is segmented by applying a threshold to the histogram response. In order to constrain the search space in which an image element can be found, the application of the threshold is confined within a boundary that is slightly larger than the segmented result of the previous frame. The results produced after the threshold application are cleaned using standard image noise reduction and hole filling techniques. The segmented area can be used to adapt the MRFH through evolving scene changes and to better describe the region in subsequent frames.

The added discrimination to the histograms and the non-parametric region identification significantly improved results to the overall technique. Figure 3.5 demonstrates how well these added features allowed for a better segmentation. Despite all these improvements the algorithm has a tendency of converging towards the most prominent features of the MRFHs. The boundary pixels of a region tend to have properties that are not predominant within the histograms. When a back-projection is made these boundary pixels have a very low probability which results in their exclusion from the final segmentation. This phenomena progresses throughout a sequence and eventually results in segmented regions that insufficiently cover the regions of interest. The exclusion of near boundary pixels is clearly observed within Figure 3.5. In this

figure the boundary of the pianist's shirt is excluded from the segmentation, this is aggravated by local light changes brought upon by motion. The segmentation of smaller regions is also a major problem. Since these regions do not contain an abundance of information, the Receptive Fields selected to describe them must be acutely tuned in order to capture a discriminative description. This is not trivial and often involves considerable amount of guesswork for arbitrary elements.

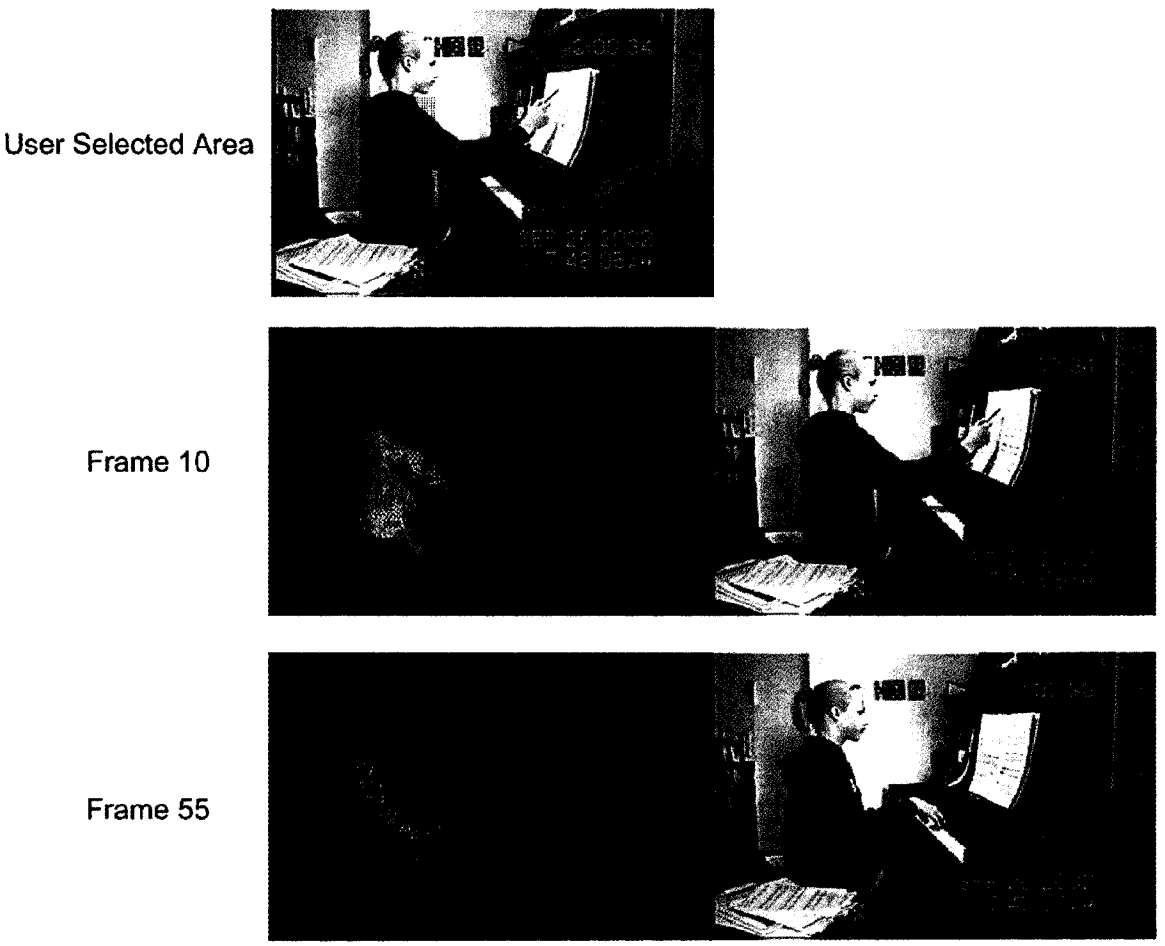


Figure 3.5 - Segmentation Using Modified CAMSHIFT with Receptive Field Histograms

The technique presented within this section shows great potential in segmentation of human targets in unconstrained environments. However, a significant amount of work is required in order to identify which characteristics should be used when segmenting particular image elements. For now the modified CAMSHIFT algorithm is best suited to simple environments with good lighting conditions, small slow motions and good object colour contrasts.

3.4 Neural Network – Multi-Layered Perceptron

The following section looks at some trial implementations of neural network based algorithms for segmenting human targets. The goal of these implementations was to assess the feasibility of using a neural network for segmentation within unconstrained environments. All the implementations shown here are based on a multi-layered perceptron (MLP) neural network; literature covering this type of network is both abundant [41]-[43] and applications similar to the one expressed within this work can be found.

One of the disadvantages of using an MLP network is the required training that enables it to learn how to classify input vectors. Training was performed using a gradient descent approach and used pre-segmented images as input. These pre-segmented images were taken at various points within the sequence and were segmented by hand. The creation of the training set was very tedious and prone to human error. Figure 3.6 shows training set examples with the original frame image on the left and the corresponding pixel classification on the right, where the region of interest is depicted in white. A large amount of frames, approximately 1 frame per second, were used in the training set in

order to optimize the algorithm's results. While this type of initialization was less than ideal, the goal here was only to test the approach's feasibility within the context of this work.

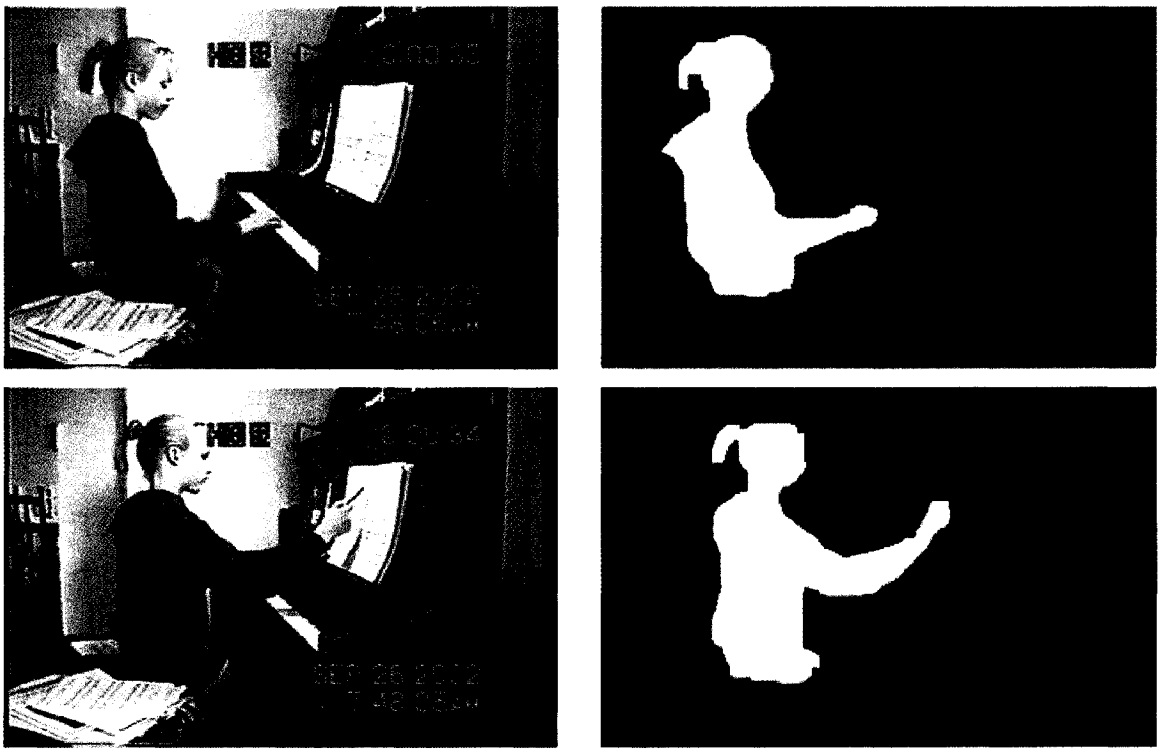


Figure 3.6 - Neural Network Training Data Examples

One of the major problems in using neural networks for segmentation stems from the data's high level of dimensionality. Image frames used here had a width of 320, a height of 240 and used a 24 bit colour representation. Feeding an entire image to an MLP network would have required an unreasonable number of input nodes and consequently computing resources. Literature on the topic [41]-[44] has suggested various data representations that minimize the required number of inputs. This literature was reviewed in the previous chapter and formed the basis of the different input sets tried

within this implementation. Each trial network was fed one pixel data vector at a time in order to perform its classification; this avoids the need for an arbitrarily large input layer to the network. Figure 3.7 shows the resulting segmentation of several different types of input sets. The various input vectors were tested in order to determine which would provide the best results. In Figure 3.7a) the input data consisted solely on colour space information. Several colour spaces were used such as L*u*v*, YCrCb, RGB and HSV in order to identify colour groupings that could be used to classify pixel segmentation. In Figure 3.7b) gradient information was used as well local neighbourhood information. Nearby pixel data such as variance, mean, colour and gradient were also fed into the network in order to distinguish pixels based on their local inter-relationships. Finally in Figure 3.7c) an amalgamation of the best discriminating inputs was chosen. A combination of local data values as well as colour space information was used.

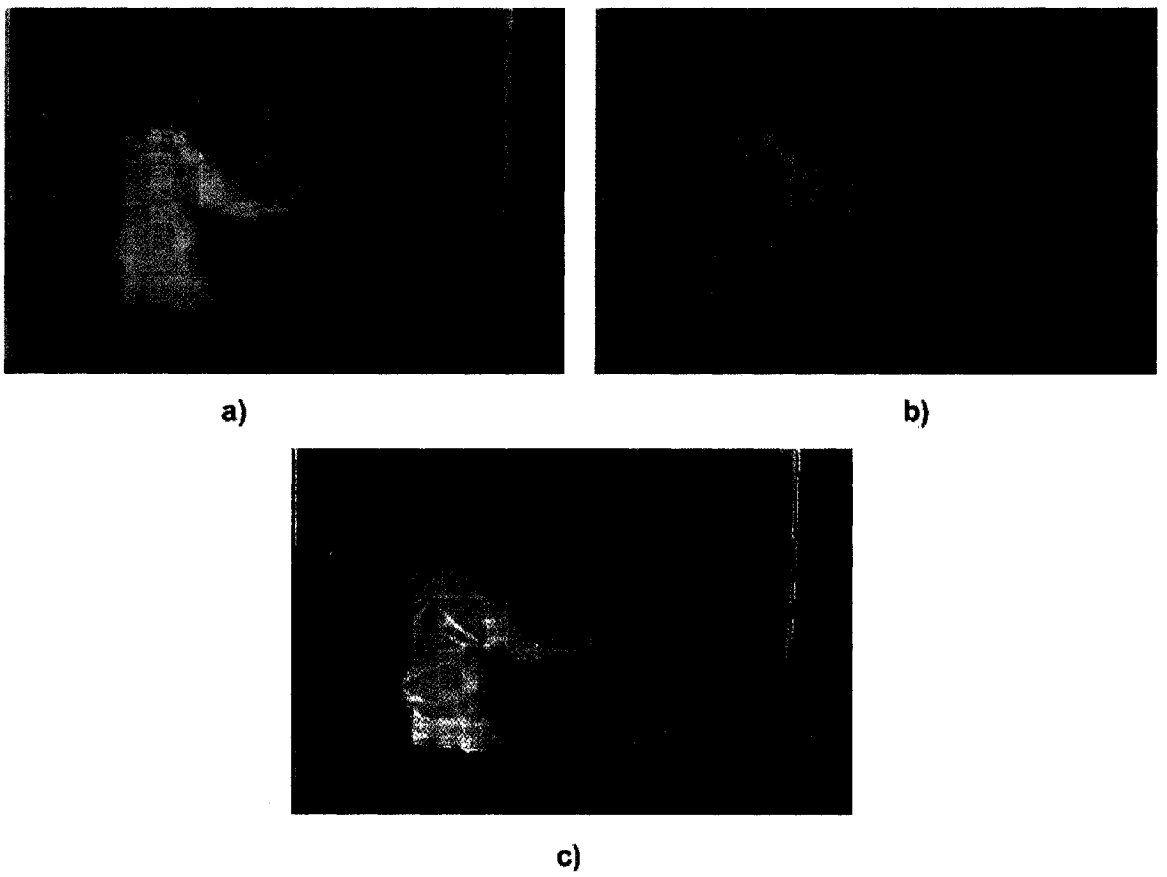


Figure 3.7 - Neural Network Based Segmentation Results Based on Several Different Input Vectors: a) Using Multiple Colour Space Vector, b) Using Gradient Magnitude and Neighbourhood Vector, c) Using Statistical and Colour Vector

Despite the MLP's powerful ability as a classifier, the segmentation of images in unconstrained environments remains mediocre. The largest problem stems from the selection of input data. The data input to the network should be able to properly discriminate the target of interest from the remaining image elements. The selection of discriminating data is not obvious and requires a hefty amount of data pre-processing. There is also no guarantee that a discriminative input set will be appropriate across different sequences. The input sets chosen within these implementations were able to properly distinguish certain features such as the pianist's upper body and arm, but were not as successful in bringing out small image elements such as the pianist's head and

hand. The inputs used here were chosen arbitrarily and thus might not take advantage of the images' most discriminating data. In order to get better results a methodology for identifying which data would be most appropriate needs to be created. Another problem with this kind of MLP use is the overwhelming imbalance in the classification sets. The number of possible background or non-segmented image elements far exceeds the amount of segmented image data. It becomes virtually impossible to train a network to identify all possible instances of data that is not to be segmented. While MLPs have great potential for classification, their use in segmentation is limited to simple environments having a limited set of possible backgrounds.

3.5 Chapter Summary

This chapter examined the various implementations of the techniques introduced within the literature review. The performance of these algorithms was tested in an unconstrained environment to determine their feasibility for the context presented here. While many of the techniques have great potential in segmenting human targets, few seem able to extend their applicability beyond that of their originating papers. Addressing the shortcomings of the above techniques would require a considerable amount of work. The Mixture of Gaussian, probabilistic skin filters, CAMSHIFT and MLP network based algorithms were only able to achieve partial results.

Chapter 4 Motion Capture in Unconstrained Environments Using *J*-Value Segmentation

The approach discussed here aims at providing a robust means of segmenting natural colour images in the hope of identifying and tracking human beings and extracting the necessary information to later perform a performance evaluation on the individual's activity. The techniques compared in the previous chapter only partially succeeded to capture the human target residing in a minimum constraint environment. To produce results that could eventually lead to a performance evaluation, significant if not overly tedious changes need to be made to the compared algorithms. The focus of this research and this chapter is to introduce such a technique which succeeds in the two dimensional motion capture of musicians in an unconstrained environment.

The proposed technique is categorized as a region-based motion capture segmentation algorithm and uses colour-texture information in order to produce homogenous regions within a set of frames that are then tracked throughout the sequence. As mentioned before, the technique is based on Deng and Manjunath's JSEG implementation [3] with key improvements in order to make it more appropriate to the context considered here. Our algorithm is described as a set of five key processes: clustering, soft-classification map creation, *J*-value segmentation, merging and tracking. This chapter will cover in detail the workings of all the key processes and explain how these processes work together to produce a final result.

4.1 Non-Parametric Clustering of Images

The first important step of the technique presented here is an initial clustering of the image colour data. Originally proposed within the JSEG technique was a clustering algorithm based on a k-means method that produced Gaussian parameterized clusters. The technique also made used of a Peer-Group Filtering approach described in [26] to initially filter noise and large colour variations. This JSEG clustering procedure proposed by Deng *et al.* [3] introduced a serious limitation for the application of the algorithm on real life images. This limitation was identified by Wang *et al.* in [28]. The use of a k-means clustering approach assumes that the colour data present within a sequence must follow Gaussian-like statistics. In real scenes and texture-rich images such an assumption cannot always be made. Also, in natural scenes colours often have a tendency to smoothly transition between two regions. These colour gradients are usually the result of local lighting changes and should, in most cases, be considered as a single colour region rather than separate regions. In order to address this limitation Wang *et al.* [28] suggested the use of a non-parametric approach to colour clustering, a subsequent major modification to the original JSEG technique.

The improvement to clustering is also adopted within this work by using the Fast Adaptive Mean-Shift (FAMS) algorithm introduced by Georgescu *et al.* [29]. FAMS is a refined version of the original Mean-Shift algorithm and supports adaptive bandwidth filters based on a pilot density estimate routine of nearby values. The authors of FAMS have also acknowledged the fact that the original Mean-Shift algorithm became prohibitively slow for high dimensional spaces, and have introduced an approximation technique based on locality-sensitive hashing (LSH) in order to optimize the algorithm.

The FAMS algorithm allows for a better initial clustering and can be applied just as easily to single or multiple sequence frames.

In order to properly cover the FAMS clustering algorithm used as part of this technique, the properties of the Mean-Shift algorithm must first be covered. The Mean-Shift algorithm is, in short, a non-parametric, kernel-based procedure to analyze multimodal feature spaces [35]. It is used within this technique to cluster colour distributions within a set of frames without applying constraints with regards to the nature of the distributions. In many of the papers covering the use of a Mean-Shift clustering, the data vectors used are composed of both spatial and colour information. In the case presented here, only colour information is of interest. By clustering video data independent of spatiality a subsequent region creation process can make its own decision on how best to group nearby pixels.

The following explanation of the Mean-Shift and its improved version, FAMS, is given with respects to a d-dimensional data space. However, as mention before, for the purpose of the technique presented within this thesis, only colour data is used, in this case the L*u*v* channels. Some basic concepts with regards to kernel density estimation are given in Appendix B. Suppose that we are given n data points such as $x_i \in R^d$, $i = 1,\ldots,n$ associated with a bandwidth $h_i > 0$. The multivariate kernel density estimator at location x is defined with the following:

$$\hat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \tag{4.1}$$

The density estimator is based on a spherically symmetric kernel K satisfying:

$$K(x) = c_{k,d} k(\|x\|^2) > 0, \|x\| \leq 1 \tag{4.2}$$

where $k(x)$ is a function defining the kernel profile and $c_{k,d}$ is a normalization constant so that $K(x)$ integrates to 1. The density gradient estimator can be obtained from the gradient of the density estimator from equation (4.1) yielding:

$$\hat{\nabla} f_{h,K}(x) \equiv \nabla \hat{f}_{h,K}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k' \left(\left\| \frac{(x - x_i)}{h} \right\|^2 \right) \tag{4.3}$$

If the derivative of the kernel profile $k(x)$ exists such that the function $g(x) = -k'(x)$ is defined, the introduction of $g(x)$ into equation (4.3) will give the following:

$$\begin{aligned} \hat{\nabla} f_{h,K}(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x_i - x) g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} x_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \end{aligned} \tag{4.4}$$

where the last term is called the Mean-Shift:

$$m_{h,G}(x) = \left[\frac{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \tag{4.5}$$

This term, by definition, points in the direction of the maximum increase in the density. In the segmentation algorithm the kernel profile is represented by a simple polynomial equation where the highest order can be modified by the operator.

Since the Mean-Shift term always points to the maximum increase in density, the kernel can be shifted onto this maximum iteratively until a distribution mode is reached. Data points that have been shifted onto during the iterative procedure can be labelled based on the mode to which they have converged. Similarly, if a set of Mean-Shift iterations lands on a point that has already been labelled, then all preceding points can inherit the label inclusively. Thus, data pixels are grouped based on the modes to which they converge. This type of clustering is highly desirable since it does not impose any type of parametric representation on the data clusters and also allows the clustering of points that may have smooth transitions. In terms of colour clustering, Mean-Shift allows gradient colours to be group together despite local, yet smooth, colour changes and allows for complex colour patterns to be clustered together.

The FAMS algorithm improves upon the Mean-Shift in two significant ways; it attempts to optimize the selection of the bandwidth size h_i for a given point x_i and also introduces a way to speed-up the overall clustering process. In [29] the bandwidth

parameter h_i is computed based on a pilot density estimate of the point x_i using its k-nearest neighbours. The bandwidth h_i is given by the following:

$$h_i = \left\| x_i - x_{i,k} \right\|_1 \tag{4.6}$$

Here the term x_i is the data point for which we want to compute the kernel bandwidth h_i using its k-nearest neighbour pilot density estimate $x_{i,k}$. The L_1 norm, the summed magnitude difference, is used for the computation. In short, this adaptive bandwidth allows larger kernel windows to be used in cases of low data variations, and smaller kernel windows in the opposite case. The adaptability of the bandwidth means a more precise and less costly Mean-Shift clustering. Note that if the term h_i is fixed as a static value h, then the adaptability is removed and the algorithm becomes the original Mean-Shift.

The major bottleneck to the Adaptive Mean-Shift algorithm presented up to now is the need to perform neighbourhood queries in order to compute the Mean-Shift term, eq. (4.5). The naive way of determining if point x is covered by the kernel of point x_i is to scan the entire data space and test the hypothesis. In order to speed up the process Georgescu *et al.* [29] use a principle called locality-sensitive hashing. This process tessellates the dataset L times, each partition containing K inequalities. Each inequality is defined as follows:

$$x_{i,d_k} \leq v_k \quad i = 1,\ldots,n \tag{4.7}$$

Here x_{i,d_k} is the value of the point x_i along the d_k dimension which is randomly chosen for the k^{th} inequality. The random value v_k defines the inequality along the dimension, creating the appropriate tessellation. This data grouping is done L times and allows the dataset to be described by a K-dimensional Boolean vector for each value of L. These vectors can easily be stored within a hash table for future reference. When a query is made for the neighbours of a point q, L Boolean vectors are computed using (7) and index the colliding cells throughout the tessellation. The union of these cells provides a subset of points that can be used to compute the region covered by the Mean-Shift kernel, eq. (4.5). Approximation errors can be mitigated by making the union of the tessellation cells bigger. Figure 4.1 depicts a visual representation of the hashing for a sample two dimensional data set. Each plane represents a set of K inequalities, defining a total of L groupings. The union of the cells containing a point q_i define the neighbourhood used in the computation of the kernel for the Mean-Shift algorithm.

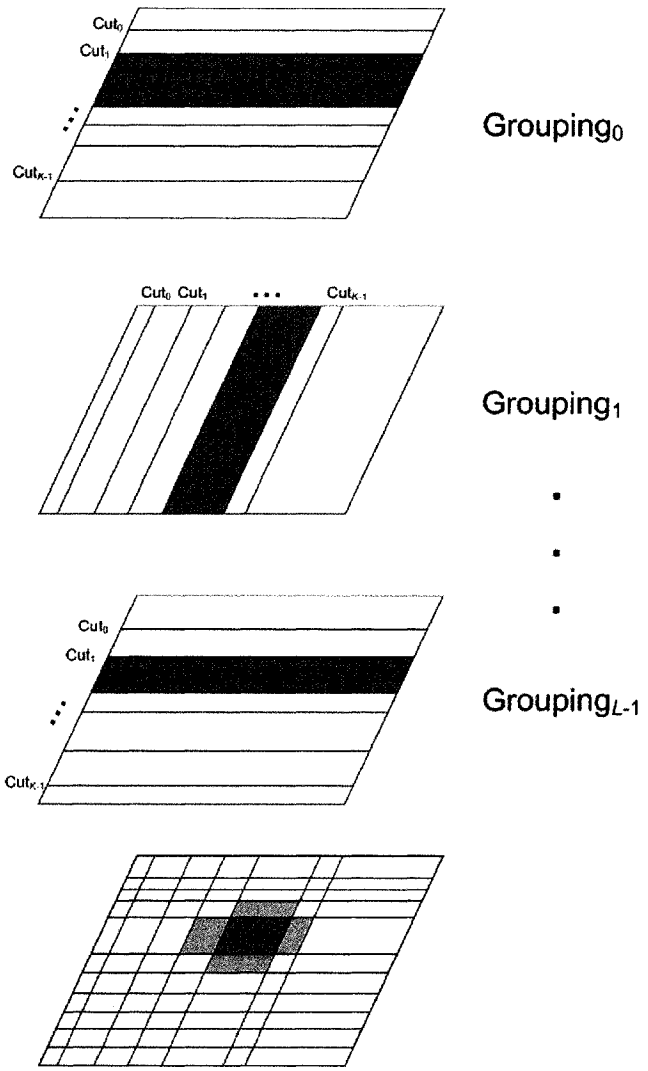


Figure 4.1 - Locality-Sensitive Hashing for Two Dimensional Data

Results and performance issues using this type of clustering are given in the next chapter. A discussion with regards to the implementation as well as the application of the adaptive and locality-sensitive hashing is also provided within the context of the application of this thesis.

4.2 *J*-Value Segmentation

The next process executed within the technique consists of an analysis on the clusters generated using the FAMS algorithm. However, in order to better understand why this analysis is even required the third step is presented first: *J*-value segmentation. This segmentation relies on the basic principles of JSEG [3] and involves the determination of homogenous colour-texture regions.

JSEG is a novel segmentation technique that attempts to produce regions out of pixel labelled images. In this case the labels are generated by the FAMS process described earlier. The labels represent the colour classes to which each given pixel belongs. The first step in the segmentation is to compute a homogeneity measure on the colour-texture property of a pixel based on its neighbours. In other words, this measurement depicts the local variation in colour classification surrounding a pixel. This value, called the *J*-value, is presented here following the same notation adopted by Deng *et al.* [3]. First the mean position of classes is defined as:

$$m = \frac{1}{N}\sum_{z \in Z} z \tag{4.8}$$

where m is the mean, Z the set of all N data points within a local region around a pixel and $z = (x, y), z \in Z$. Assuming that there are a total of C colour classes, the mean position of a particular class i is defined as:

$$m_i = \frac{1}{N_i}\sum_{z \in Z_i} z \tag{4.9}$$

The total spatial variance of classes is given as:

$$S_T = \sum_{z \in Z} \|z - m\|^2 \tag{4.10}$$

And the sum of all class variances as:

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z_i} \|z - m_i\|^2 \tag{4.11}$$

The J-value of the local region is obtained based on these variances:

$$J = (S_T - S_W) / S_W \tag{4.12}$$

The original paper provides examples on how a particular local class distribution would affect the outcome of the J-value. For a local region where classes are distributed approximately uniformly, the J-value will remain relatively small. Inversely, should the local region consist of densely segregated classes; the J-value will increase. The result of an image wide J-value computation is a gradient image corresponding to homogeneous colour-texture edges. Example results are given in Appendix C.

The set of points which define the local region on which the J-value is computed is described by a circularly symmetric kernel mask. This mask is applied to every pixel in an image to produce a J-image. The kernel size varies depending at which scale J-images are to be created. At a larger scale smoother texture edges are detected while at smaller scales hard edges are detected. The scale changes the size of the kernel mask; for example, at a scale of 2, the mask is up-sampled along the X-axis and Y-axis by a factor of 2. As the kernel is up-sampled the new coordinates are not included in the J-value computations, this means that the algorithm does not incur an increased amount of

processing at larger scales. By up-sampling the kernel the local neighbourhood taken into consideration is increased and the resulting J-value will reflect the homogeneity measured of this larger neighbourhood. The segmentation begins with region determination using the largest scale. Once regions have been determined the process is repeated for each region at the next smallest scale. Regions will split if they lack homogeneity at the smaller scale. A seed growing algorithm is used to create regions by amalgamating nearby pixels having a low J-value. In order to reduce the number of regions and to avoid over-segmentation an initial set of seeds are created through the application of a J-value threshold. An iterative process then assigns individual values to their closest appropriate labelled region.

The JSEG algorithm also allows for video segmentation by way of seed tracking. When a set of region seeds are discovered within a current frame, their overlap with previous seeds is computed. If an overlap is found between two seeds in two subsequent frames, the newer seed is assigned the same label as its predecessor. If a seed overlaps with multiple previous seeds, then a new label is generated and assigned to each of the overlapping seeds, thus changing both the current and previous frame labelling. The seed tracking algorithm presented by Deng *et al.* [3] requires that all video frames be segmented at once and depends on small motion between frames. This is not practical for very large or lengthy videos; a solution to this is presented within this chapter under the section 4.4. To further increase robustness with regards to false merging due to motion effects, the term J_t is introduced. This value represents the temporal texture-colour homogeneity and is used to determine whether a pixel should be used in overlap determination. Much like its J-value counterpart, the J_t-value is computed as:

$$m = \frac{1}{N} \sum_{z \in Z} t_z \tag{4.13}$$

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} t_z \tag{4.14}$$

Shown above are modifications to equations (4.8) and (4.9). They have been modified to compute the temporal mean of all classes and of a particular class i respectively. Here t_z represents the relative time index of a point $z = (x, y)$. Since only two consecutive frames are taken into account for this computation, $t_z \in \{0,1\}$. Note that equation (4.13) will always give a value of 0.5 since each frame contains the same number of points within a local defined area. Finally the J_t-value is computed using the following procedure.

$$S_T = \sum_{z \in Z} \| t_z - m \|^2 \tag{4.15}$$

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z_i} \| t_z - m_i \|^2 \tag{4.16}$$

$$J_t = (S_T - S_W) / S_W \tag{4.17}$$

The J_t-value of a pixel is shown to be large whenever its surroundings between frames have changed significantly and vice versa if the cluster grouping between frames remains fairly static. By computing this value for the seed pixels prior to overlap determination, regions that have undergone a lot of motion can be identified and omitted from the calculation. This avoids false merges for high motion scenes. Both the J-value

and J_t-value are computed independently, the J-value is used in seed determination while the J_t-value is used in the seed overlap determination.

This section has described the JSEG algorithm with little deviation from its original publication or proposed methodology. Despite its success in segmenting complex scenes, the technique has several shortcomings that must be addressed if it is to be successful in uncontrolled environments such as the one presented within the context of this thesis. One of the major shortcomings and improvements brought to this technique has already been described in the previous section. The use of adaptive clustering avoids misrepresentations in J-images by classifying colours exhibiting subtle colour gradients into common groups. Other improvements relating to colour, tracking and over-segmentation follow in the subsequent sections.

4.3 Soft-Classification Maps

The addition of a non-parametric clustering algorithm partially solves the problem of segmentation in the presence of strong colour gradients. Ultimately following a clustering procedure, every pixel is only given a single hard classification. In their list of improvements to the original JSEG algorithm, Wang *et al.* [28] introduced the concept of soft-classifications. Using the fact that a pixel can be represented as a mixture of clusters, the authors propose the means with which a pixel's membership to a specific class can be measured.

The membership value $\mu_{z,i}$ for a pixel z, characterized by the colour vector I_k, to a specific class i is described by the following:

$$\mu_{z,i} = \frac{P(w_i)P(I_k \mid w_i, \theta_i)}{\sum_{j=1}^{C} P(w_j)P(I_k \mid w_j, \theta_j)} \tag{4.18}$$

The cluster classes are assumed to follow Gaussian distributions represented by w_i, $i = 1,...,C$, for a total of C classes. In the above equation $P(w_i)$ represents the prior probability while $P(I_k \mid w_i, \theta_i)$ demonstrates the probability that pixel I_k belongs to the distribution w_i where $\theta_i = (u_i, \Sigma_i)$ represents the mean and variance matrix. The term $P(w_i)$ is computed as a ratio of the pixels belonging to class i and the total number of pixels. The second term $P(I_k \mid w_i, \theta_i)$ is computed using a standard Gaussian model.

$$P(I_k \mid w_i, \theta_i) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(I_k - u_i)^T \Sigma_i^{-1} (I_k - u_i)\right\} \quad i = 1,...,C \tag{4.19}$$

The assumption that the cluster classes follow a Gaussian distribution goes against the initial work done by the non-parametric clustering algorithm. The classes are known to not have a Gaussian distribution due to the way they were created. This oversight on the part of Wang *et al.* [28] is addressed within this work by way of a non-parametric representation of the term $P(I_k \mid w_i, \theta_i)$. Swain and Ballard [38] have demonstrated how histograms can be used to represent colour distributions and to localize objects. This methodology is also used here. A three dimensional RGB colour histogram of every cluster is created and by back-projecting these values into the image, a pixel's probability can be computed. A histogram back-projection process consists of replacing individual pixel colour vectors with their normalized histogram bin value. In other

words, after constructing a histogram based on the pixels of a given cluster, all image pixels are replaced with the value they index into the histogram. The result of a histogram back-projection is a probabilistic representation of the image given by the distribution found in the histogram. This allows soft classification maps to be computed without compromising the initial assumption of non-parametric data.

With the addition of soft-classification maps, the functions relating to class mean and variance for J-value computations are adapted as follows:

$$m_i = \frac{\sum_{z \in Z} z \cdot \mu_{z,i}}{\sum_{z \in Z} \mu_{z,i}} \qquad i = 1,\ldots,C \tag{4.20}$$

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z} (\mu_{z,i} \cdot \|z - m_i\|^2) \tag{4.21}$$

Here the value $\mu_{z,i}$ represents the weighted membership the pixel z has with class i. This adaptation of the JSEG equations allows the procedure to take into account the varying degree of membership a pixel may have with the different colour cluster distributions. It also prevents JSEG from falsely identifying smoothly varying colour regions as smooth edges.

4.4 Joint-Criteria Region Merging

Both the original and modified JSEG approaches unfortunately suffer from an over-segmentation problem. Its original authors [3] have proposed a simple merging algorithm which iteratively attempts to bring together two regions having the closest

corresponding histograms. JSEG however is not the first segmentation algorithm to provide over-segmented results and the issue of region merging has been explored extensively within other contexts [18]-[23]. We choose to adopt an algorithm that uses a joint space merging criterion introduced by Hernandez *et al.* [23]. This technique not only relies on colour information but also on the number of edge pixels between two candidates. As such, it prevents the accidental merging of regions with similar colour attributes having a strong edge in between them. This merging process is performed on any two regions whose adjacency is present throughout a set of frames.

The first step in performing the merge operation is to formulate a Region Adjacency Graph (RAG) [22]. The graph nodes represent regions within the image while edge costs are assigned according to the similarity between two adjacent regions. Once the RAG is constructed, regions having the highest similarity can be merged and provoke an update of the graph. The process is iterative until the similarity criterion achieves a set threshold or the desired number of regions has been obtained.

The similarity criterion used stems from Hernandez *et al.* [23] and is based on both a colour homogeneity and edge integrity measure. The colour homogeneity criterion is defined as follows for two adjacent regions i and j defined in a K-partitioned image:

$$\delta^{H}(R_K^i, R_K^j) = \frac{\left\|R_K^i\right\| \cdot \left\|R_K^j\right\|}{\left\|R_K^i\right\| + \left\|R_K^j\right\|} \left[\mu(R_K^i) - \mu(R_K^j)\right]^2 \tag{4.22}$$

Here, $\mu(R_K^i)$, $\mu(R_K^j)$ and $\left\|R_K^i\right\|$, $\left\|R_K^j\right\|$ represent the mean RGB colour values and sizes of the regions i and j respectively. The colour distances are weighted with a size

difference measure; smaller regions with a large colour distance will be characterized with a large value of $\delta^H(R_K^i, R_K^j)$.

The edge integrity criterion is based on the ratio of strong edge pixels and regular edge pixels found along the boundary of two adjacent regions. In order to compute this ratio, a gradient image is first created using Wang's [19] morphological method. A threshold is found based on the median value of the gradient image. Any pixels found to have a value higher than the threshold are considered strong boundary pixels. The criterion can now be summarized as follows:

$$\delta^{\varepsilon}(R_K^i, R_K^j) = \frac{\left\| \varepsilon_S^{i,j} \right\|}{\left\| \varepsilon_B^{i,j} \right\|} \tag{4.23}$$

The terms $\varepsilon_S^{i,j}$ and $\varepsilon_B^{i,j}$ represent the number of strong and the total number of boundary pixels respectively. Regions having a strong boundary ratio will end up having a larger value of $\delta^{\varepsilon}(R_K^i, R_K^j)$.

In order to produce a single similarity criterion for the merging procedure, both the homogeneity and edge integrity criteria must be evaluated. Since their scales are not known, [23] suggests using a rank based procedure where the final similarity is given by:

$$W = \alpha R^H + (1-\alpha) R^{\varepsilon} \tag{4.24}$$

Here R^H and R^{ε} are the respective ranks of the criteria given above for the same two adjacent regions. Ranks correspond to the sorted position in which two regions' value of $\delta^H(R_K^i, R_K^j)$ and $\delta^{\varepsilon}(R_K^i, R_K^j)$ are placed. α is a weight parameter used to impart importance on either of the former criteria. Hernandez *et al.* [23] suggest setting the

weight factor to 0 for a few cycles in order to remove false background regions having little to no boundary pixels. They then set this factor based on the ratio of small regions present in the image. Figure 4.2 depicts the process of merging regions together using a weight factor of 0.5. The region represented by node 2 is merged with node 1 since it has the lowest value of W.

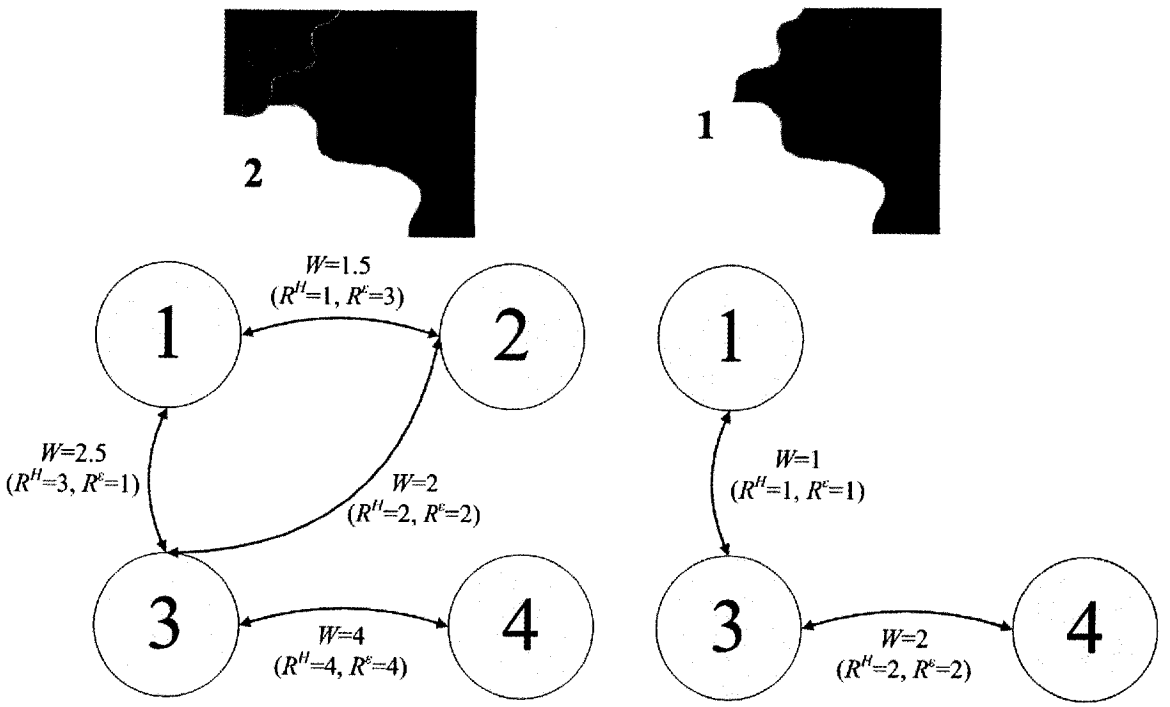


Figure 4.2 - Sample Region Merging Process

4.5 Region Tracking

At the start of this thesis, within the literature review a distinction was made between sequential and video block segmentation. That distinction is revisited here in order to describe the tracking algorithm developed for this framework. Sequential segmentation manipulates only a single frame at a time. While this manipulation can

utilize any amount of information previously compiled from preceding frames, the determination is performed only on the current frame. Video block segmentation on the other hand will manipulate multiple side-by-side frames at once. Region determination is consequently performed on all these frames simultaneously. This set of frames, sometimes called a video stack, has certain advantages when it comes to tracking. Region creation on a given frame can be influenced by its temporal neighbours and allows the segmentation to be adapted across the video stack. This advantage comes at a hefty memory and computational cost since all video stack frames must first be buffered. The following sections describe the hybrid strategy used in this work in order to track regions throughout a video using these two techniques.

4.5.1 Intra-Video Stack Tracking

A combination of parallel and sequential tracking is used within this work. The authors of JSEG have adapted their algorithm in order to allow for parallel segmentation through the introduction of the J_t-value as described in section 4.2. Their adaptation however requires that the entire video be segmented at once in order to be successful and is often not feasible due to memory constraints and video sizes. For this reason the segmented video is first separated in a series of video stacks. Stack sizes are determined by a number factors including available memory, processing time, video complexity, length, etc. Since video stacks must be buffered prior to processing this approach precludes the use of the system without incurring some delay.

As demonstrated in Figure 4.3 each video is split in sets of video stacks, the size of which can be manipulated by an operator and depends on the available memory and

computing power. The tracking and region determination done within a video stack is the same as the one proposed by in JSEG. That is to say, the J-value and J_t-value are used in order to produce regions and to find correspondences between frames. The tracking done in between stacks is described in the next section.

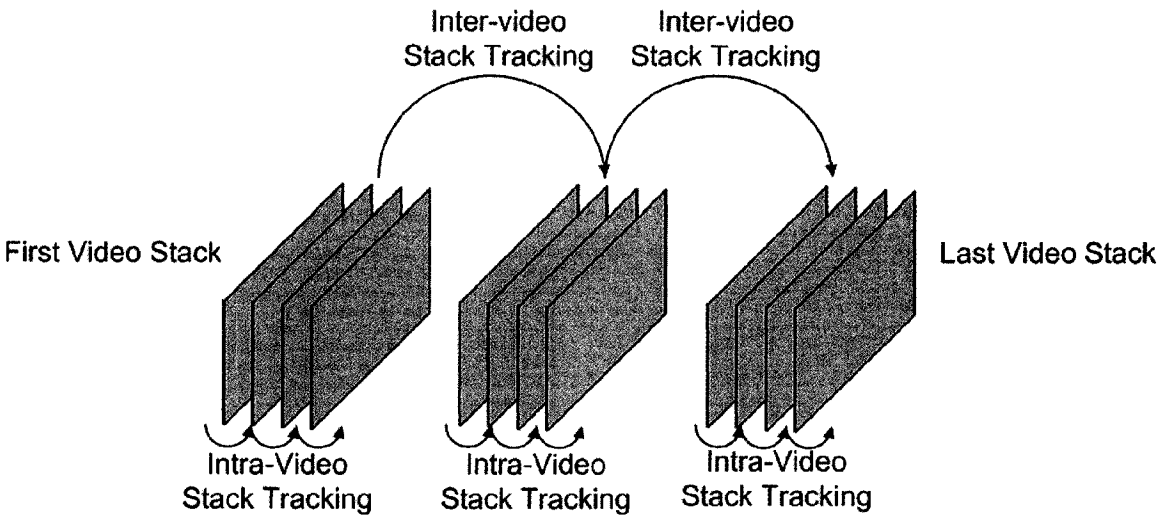


Figure 4.3 - Video Stack Tracking

4.5.2 Inter-Video Stack Tracking

The inter-video stack tracking algorithm proposed in this thesis is strongly based on region overlaps between two consecutive video frames. This means that the motions exhibited by the objects in the video must be captured with an appropriate frame rate in order to allow regions to have an overlay between frames. Multiple techniques have used overlap in the past as strong correspondence indicator for region tracking [53]-[56]. The tracking correspondence indicator used within this work stems from the research produced by Withers *et al.* [56]. In their work, the authors have tried to identify regions

correspondences between frames regardless of splitting, merging and non-uniform changes to the regions. This tracking methodology lends itself well to the segmentation technique presented here because despite the relative stability of regions between frames, subtle changes in the scene may often cause the same type of region behaviour described within Withers *et al.*'s research.

The criteria used to find a correspondence between regions of two subsequent frames depends highly on both distance and pixel overlap. In this case pixel overlap is defined as the number of pixels one region has in common with another between two successive frames. The authors of [56] define the overlap-ratio, $R_{i,j}(t)$, as the correspondence measure between region i and j. It is given by the following equation.

$$R_{i,j}(t) = \frac{V_{i,j}(t)}{D_{i,j}(t)} \tag{4.25}$$

Here the term $D_{i,j}(t)$ is a distance measure between regions i and j, and will vary along the interval [1, ∞]. The term $V_{i,j}(t)$ is an overlap measure between regions i and j, and will vary along the interval [0, 1]. The result, $R_{i,j}(t)$, will consequently be found in the range of [0, 1], where 1 indicates a perfect match and values closer to 0 correspond to regions far apart or with little overlap. The term $D_{i,j}(t)$ is defined as the fraction of the Euclidian distance between regions i and j with respects to the minimal distance involving these regions. Formally, the term $D_{i,j}(t)$ is expressed as follows:

$$D_{i,j}(t) = \frac{d_{i,j}(t)}{\min\limits_{1 \le l \le m, 1 \le k \le n} (d_{l,j}(t), d_{i,k}(t))} \tag{4.26}$$

Here the function $d_{i,j}(t)$ denotes the Euclidian distance between the centers of mass. The assumption is made that the frame at time t has at most m regions, while the frame at $t+1$has at most n regions. Finally the overlap term $V_{i,j}(t)$, is computed similarly, and corresponds to the fraction between the number of overlapping pixels and the smallest area size of the regions taken into consideration.

$$V_{i,j}(t) = \frac{B_{i,j}(t)}{\min(A_i(t), A_j(t+1))} \tag{4.27}$$

The function $B_{i,j}(t)$ represents the number of overlapping pixels while the functions $A_i(t)$ and $A_j(t+1)$ represent the area size of regions i and j respectively. The values from equations (4.25) and (4.26) approach 1 whenever the regions in question have similar spatial coordinates and overlapping areas respectively.

Using the value provided by $R_{i,j}(t)$, each region of the frame $t+1$ can be matched to a certain degree with its counterpart in the frame t. Regions that may have undergone a splitting or merging will still have a very large overlap-ratio with their ancestors. This is due to the fact that little motion occurs between frames and so position and pixel overlap remains somewhat constant. By applying a threshold to the overlap-ratio, eq. (4.25), final correspondence can be achieved.

4.6 Chapter Summary

The technique proposed in this chapter aims to segment human targets in unconstrained environments. The algorithm focuses on the JSEG implementation first

introduced by Deng *et al.* [3] but also introduces major modifications to this algorithm. Specifically, a better data manipulation by way of non-parametric clustering was described along with its role in the creation of soft-classification maps for homogeneous texture-colour region identification. The over-segmentation shortcoming of the original JSEG technique was addressed using joint-criteria region merging and a tracking algorithm is proposed to extend the technique's applicability to videos.

Chapter 5 Experimentation

This chapter discusses the experimental setup used in this work and offers an in-depth overview of the results provided in several test cases. These test cases are aimed at determining the motion capture capabilities of the overall framework and its ability to function in several complex environments. Results for each step of the technique are also compared with Deng's JSEG algorithm [3] in order to clearly demonstrate the improvements brought to the original technique.

5.1 Experimental Setup

This section looks at the experimental setup used in the acquisition and processing of video for the motion capture system. A multi-camera setup is used in order to provide the basis with which calibration, acquisition, 3D reconstruction and motion analysis can later be performed. The infrastructure introduced within this chapter is beyond the scope of the present work and was performed by a colleague [57]. Camera specifications, video compression and computing facilities are also discussed within this section.

The goal of using a multi-camera setup in an infrastructure similar to the one shown in Figure 5.1 is that it allows the capture of several key positions within a predefined workspace. Using a motion capture technique, such as the one described in this thesis, along with calibration and 3D reconstruction algorithms a comprehensive analysis and replay of motions can be performed. The infrastructure is composed of 8 Flea2™ Firewire 1394b cameras mounted on a structure covering a workspace of

approximately 2.5m x 2.5m x 2.5m. The cameras are fixed to the structure in stereo pairs covering the left, right, back and top views of a target. Figure 5.2 shows a diagram of the camera positions.

Figure 5.1 - Infrastructure Design

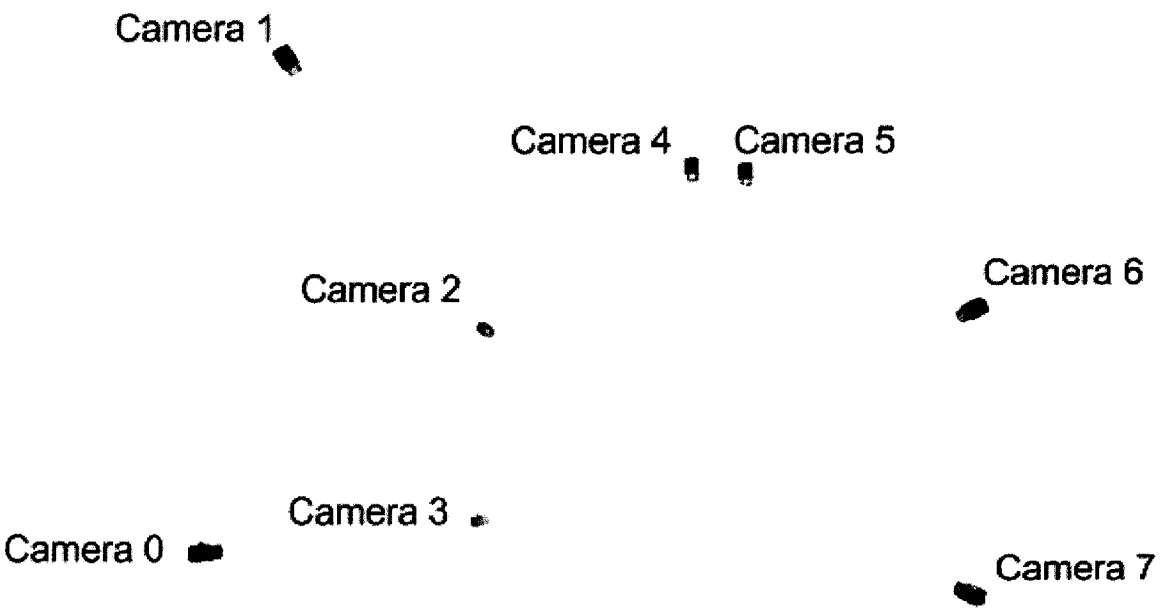


Figure 5.2 - Infrastructure Camera Setup

Acquiring 8 camera video feeds along with other meta-data such as registration information requires an excessive amount of computing power and bandwidth. A total of three computers hosting Pentium IV 3.0 GHz processors and running Windows XP were used in the acquisition process. In order to alleviate the hefty bandwidth requirements, the videos are captured with a frame size of 320x240 at 30 frames per second and compressed using the XviD codec. These same computers also host the motion capture software developed within this thesis.

5.1.1 Software Design

The goal in the design of the software was twofold: to allow the segmentation of videos into semantic regions and to provide the means with which an operator may select, track and capture motion from various regions. The segmentation process is kept separate from the interactions made by the human operators attempting to identify and interpret captured data. This section looks at the design of these two components and the working of the overall software.

The segmentation process requires considerable computing resources and in order not to interfere with user interactions it is kept in a separate process altogether. The process is implemented in C++ and compiled within a Visual Studio 2005 environment. The OpenCV vision library provides the underlining support for image manipulations. Communication with the process is done using command line arguments; its output is a set of segmented images stored on disk using a lossless compression. All parameters are provided to the process through a configuration file specified at runtime.

The motion capture interface, seen at Figure 5.3, is also implemented in the Visual Studio 2005 environment using Windows MFC components and the OpenCV library for image manipulations. The GUI provides the means with which operators may monitor the segmentation process and allows them to select groups of regions for further analysis. The selection is performed by clicking on segmented region from the initial frame. When multiple regions are selected they can be combined together by the operator in order to form separate groups of interest. The groups can be made to represent various semantic components of an image. As seen in Figure 5.3, three groups are identified in red, green and blue representing the pianist's head, torso and arm respectively. The groups can either viewed independently or as a complete ensemble. This allows operators to view the evolution of motion as the system processes the information provided by the segmentation process. The motion capture can also be stored to disk using a variety of options provided by the software. Corrections or reconfiguration of groups can be made on the fly as the information is processed. The application also provides an interface that allows the user to configure the various segmentation parameters; this interface is displayed in Figure 5.4.

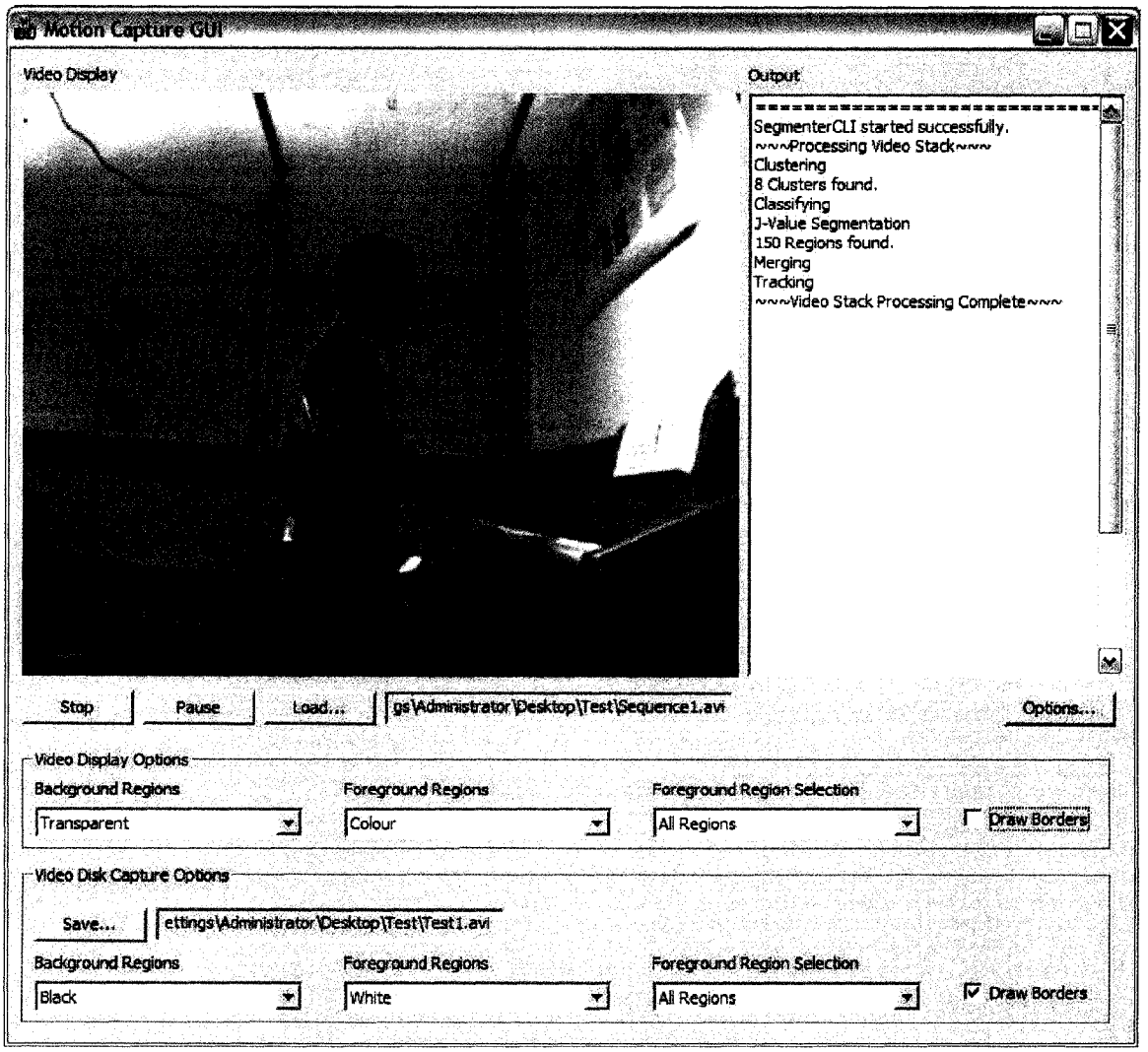


Figure 5.3 - Motion Capture Interface

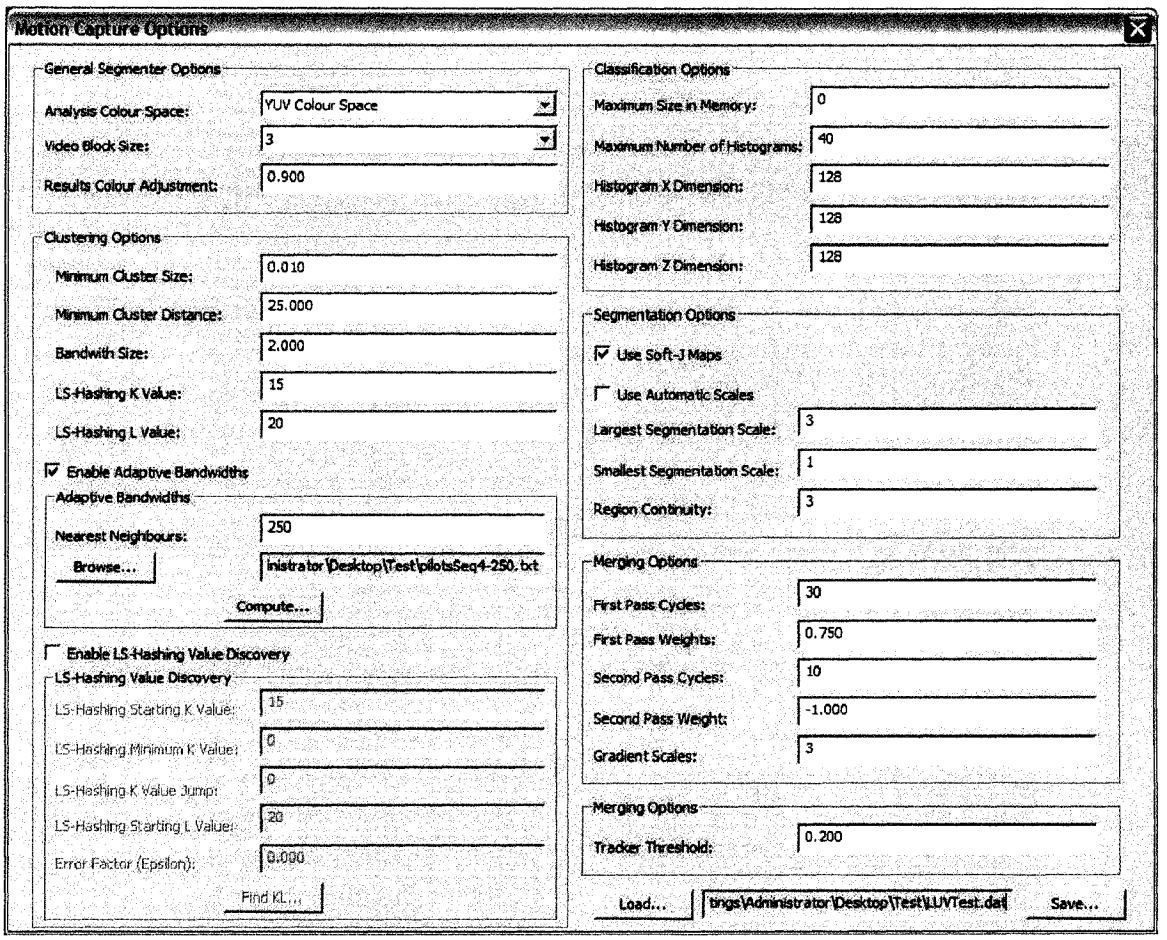


Figure 5.4 - Motion Capture Configuration Interface

5.1.2 Test Environments

One of the objectives in this thesis is to allow humans to perform in their usual day-to-day environments. These environments are numerous and can vary widely in complexity. This section looks at the videos captured in a subset of these environments and describes some of the segmentation and motion capture challenges. These videos are used to test the functionality and limitations of the technique proposed in this thesis.

The first environment examined in this section consists of the Vision, Imaging, Video and Autonomous Systems (VIVA) Research Laboratory found at the University of Ottawa. This research laboratory is by no means considered a day-to-day environment for human performers or musicians. The characteristics of this environment allow the algorithms to be tested in a simpler and more controlled setting. The room in question generally has simpler backgrounds and fewer textures. The lighting conditions in this environment can also be adjusted, background motion can be avoided and the performers' movements can be controlled. As seen in Figure 5.5, despite the simpler setting some lighting effects are still visible. Shadows still play an important role and colour contrast issues can also be observed.

Figure 5.5 - Laboratory Environment Examples

The second environment is more common for human performers. The home environment consists of video sequences taken by musicians in a more familiar practice setting. As can be seen in Figure 5.6 the complexity of the scenes is significantly greater than the laboratory environment. The background has richer textures, the foreground has a changing text overlay and the lighting conditions are less than ideal. Shadows play a significant role in the segmentation process and lighting reflections, particularly along the piano, provide an interesting challenge. Despite the added complexity, the scenes have a decent colour contrast for the more prominent image features. These videos are a good example of a day-to-day environments used by pianists. The motions exhibited by the performers are not controlled and are generally more pronounced than those within the laboratory setting.

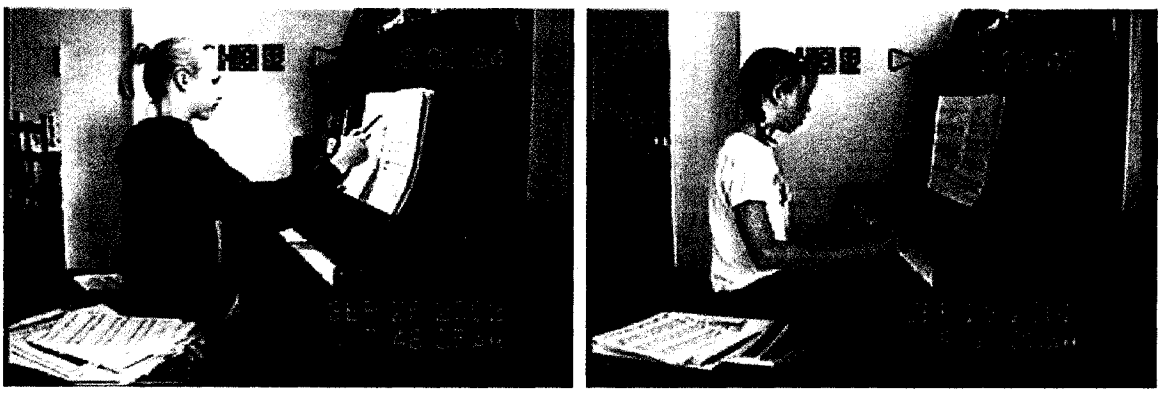


Figure 5.6 - Home Environment Examples

The last environment used to test the algorithm proposed in this thesis has the highest level of complexity. The studio environment is taken from the University of Ottawa at the Music Faculty's Piano Pedagogy Laboratory. This laboratory is used for recitals, practice, recordings and the research of piano pedagogy. This studio is also

considered to be a common environment for a performer or musician. Figure 5.7 depicts several scenes taken from the studio. The number of colours, textures and lighting effects make this environment by far the most challenging in terms of segmentation and motion capture. The goal behind using this type of environment for testing is to identify the limitations of the technique proposed within this work. In these videos a complex combination of indoor and outdoor lighting introduces a significant number of shadows and contributes to a weaker colour contrast. Background motion and specular lighting effects are also very common. The performers' movements are left uncontrolled and adjustments on the lighting are very limited.

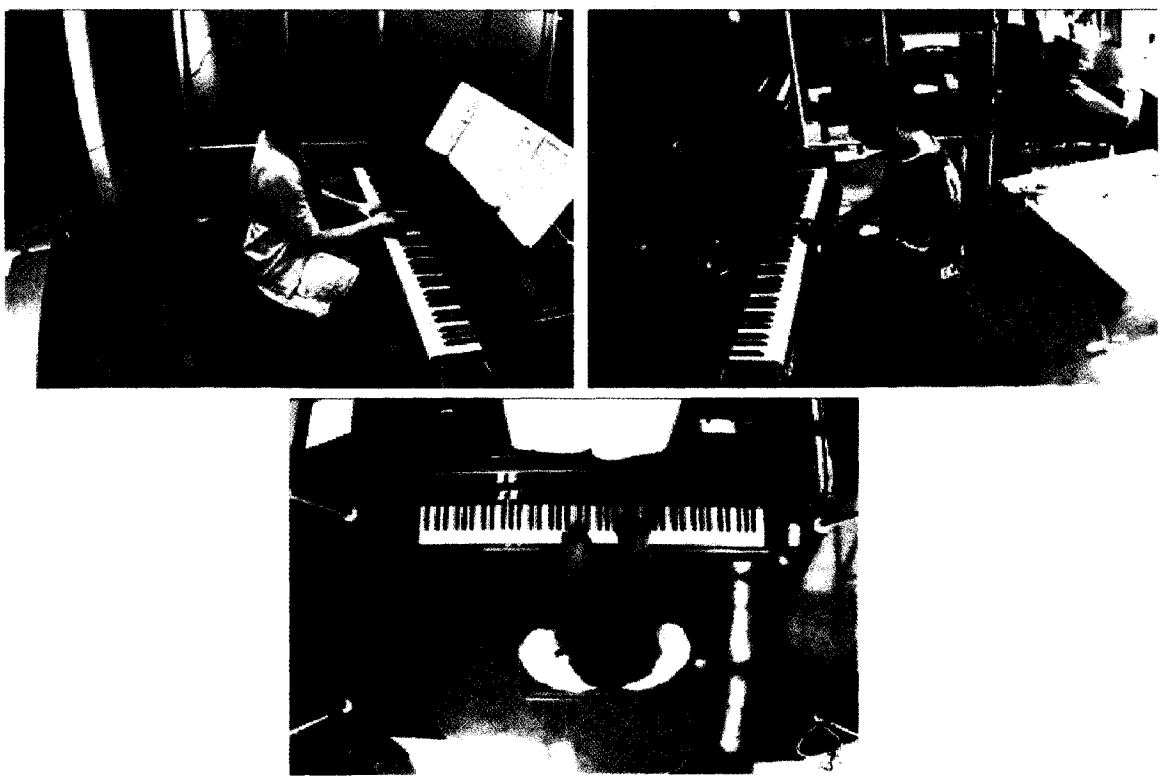


Figure 5.7 - Studio Environment Examples

5.2 Algorithm Analysis

This section provides an analysis on the intermediary results achieved by the technique suggested in this thesis. There are a total of five important steps used to achieve motion capture; these steps have already been introduced in Chapter 4. The results presented here offer an insight on how each of these steps contributes to the overall approach. A subjective and qualitative comparison will also be provided between the results from the individual steps introduced within this work and their counterparts within Deng's JSEG algorithm [3].

5.2.1 Clustering Results

The Fast Adaptive Mean-Shift (FAMS) algorithm previously covered in section 4.1 is the algorithm selected within this work to cluster the image L*u*v* colour data prior to the application of the segmentation technique. As explained earlier, the FAMS algorithm is an improvement over its predecessor, the Mean-Shift algorithm, through the use of adaptive bandwidths and Locality-Sensitive Hashing. This section looks at the impact these two improvements have on the overall clustering of unconstrained images and also compares the results with the original JSEG K-means clustering algorithm.

5.2.1.1 Impact of Locality-Sensitive Hashing Parameters

In the description of the clustering algorithm an optimization technique called Locality-Sensitive Hashing was explained. This optimization aims to expedite computations relating to neighbourhood queries performed during the Mean-Shift

computation. A more precise description of the algorithm is given in section 4.1. The parameters used in this technique change the granularity of the groups created to speed-up the neighbourhood queries. Since the groups are created based on a set of random inequalities, they do not always contain the necessary data points required for the precise computation of neighbourhood queries. This error is well documented in Georgescu's work [29]. However, the impact of these parameters should still be investigated in the context of the segmentation and motion capture framework presented here.

The parameters in question are the values of K and L, representing the number of inequalities per group and the total number of groups respectively. In the worst case scenario, both these parameters would be set to 0, meaning that all data points would have to be queried for each computation of the Mean-Shift. The processing requirements would far exceed any practical application. According to Georgescu *et al.* [29], as the value of K increases, the average size of cells will decrease thus reducing the likeliness that all the data required in an appropriate neighbourhood query will be present. This in turn would mean that more labels will be incorrectly assigned, thus introducing error into the clustering process. Similarly if the value of L increases then the intersection of cells will decrease and their union will increase. A larger union will require more queries thus a longer computation time. These phenomena are illustrated in Figure 5.8.

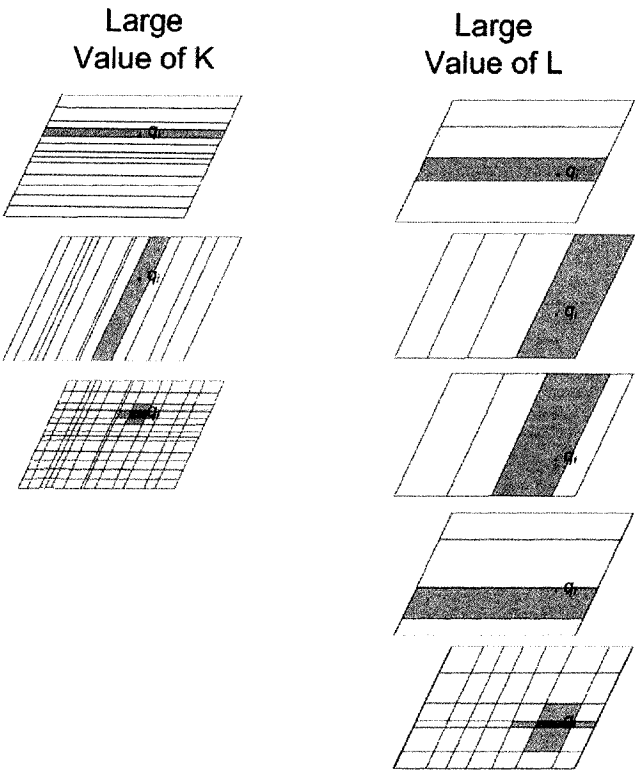


Figure 5.8 - Impact of K and L Parameters in Locality-Sensitive Hashing

In the left column the reduced cell size is noticeable with a large value of K. In the right column large intersections are observed due to the value of L. The relationship between the values of K, L, the error and the computation time has been analyzed and is demonstrated in [29]. Georgescu *et al.* concluded that the error was quickly mitigated with increasing values of L and that the relationship between K and L in order to maintain a minimal error followed a nonlinear polynomial curve. By enforcing this curve and estimating computation time for various (K, L) pairs an optimal tessellation can be obtained. Table 5.1 shows a sample frame from several videos, their clustered counterparts, their respective optimal (K, L) pairs as well as their total computation time. Note that in the following tables the cost associated to computing adaptive bandwidths is

not included; the impact of bandwidth selection is discussed in the next section. The clusters are drawn in greyscale where each cluster is represented by its own colour.

Table 5.1 - Clustering Results Using Optimal (K, L) Pair

Sequence Name	Initial Frame	Clustered Frame	Optimal (K, L) Pair	Computation Time (seconds)
Laboratory			(24, 16)	972
Home			(24, 10)	884
Studio			(20, 8)	999

The computation time is quite long; most of it is attributed to the cost of finding the optimal (K, L) pair. Even though the computation should not necessarily be redone for each frame since the data does not drastically change, it can perhaps be avoided altogether by providing a (K, L) pair that is suboptimal but sufficient for the purpose of this work. Table 5.2 demonstrates several results of (K, L) pairs for the various sequences. The total processing time is significantly reduced while the overall quality of

the segmentation remains very similar. This shorter processing time can be an advantage to the proposed clustering technique.

Table 5.2 - Clustering Results Using Suboptimal (K, L) Pair

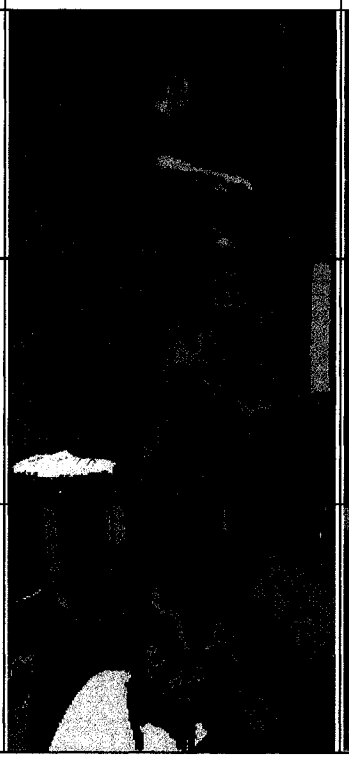
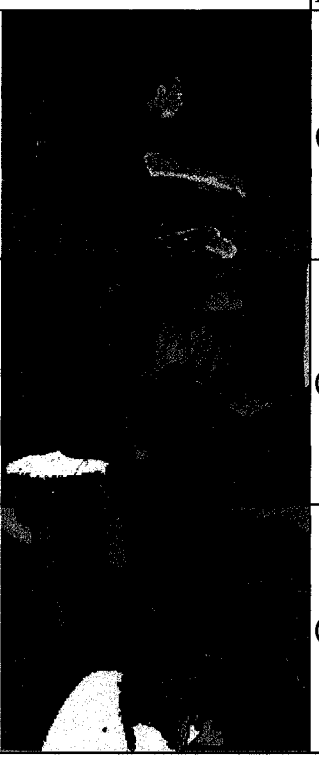
Sequence Name	Clustered Frame with Optimal Pair	Clustered Frame with Suboptimal Pair	Sub-optimal (K, L) Pair	Computation Time (seconds)
Laboratory			(22, 13)	58
Home			(25, 15)	87
Studio			(15, 5)	71

Table 5.2 clearly shows that the impact of (K, L) pairs on the formation of clusters is negligible at best. The values used in Table 5.2 were arbitrarily selected with an L value that is larger than K. If a sub-optimal pair is used then no guarantee can be provided on the level of clustering error. However, should the value of L be sufficiently elevated with respects to K any error should not severely impact the final segmentation. Also, since a sub-optimal pair is used time fluctuations in the clustering can be expected,

these fluctuations however pale in comparison to time required to compute the optimal pair. The range of values deemed appropriate vary from sequence to sequence and depend on the manner in which bandwidth selection is computed, in general values ranging from 5 to 30 provide adequate results. In this work we propose using sub-optimal pairs as a mean of speeding-up the clustering process at the cost of processing time fluctuations and possible minor errors in the clustering.

5.2.1.2 Impact of Adaptive Bandwidths

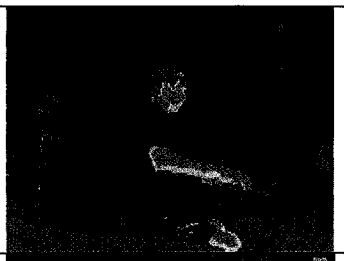
The other improvement introduced by Georgescu *et al.* [29] in the FAMS algorithm consisted of adaptive bandwidths. By estimating the pilot density of a given point using a k-nearest neighbour algorithm, Georgescu *et al.* [29] were able to compute the kernel bandwidth that would yield the optimal Mean-Shift convergence. Bandwidth selection can impact results significantly. A bandwidth that is too small will fail to converge to a mode and result in multiple tiny groups of labelled data points. A bandwidth that is too large may diverge away from the appropriate mode due to nearby distributions thus resulting in misclassifications. Table 5.3 gives an overview of the computation time involved in determining the bandwidth for each data point; the results from the application of adaptive bandwidths are seen in Table 5.4. The table was created using the optimal (K, L) pair and a k-nearest neighbour algorithm where k = 250.

Table 5.3 - Adaptive Bandwidth Computation Times

Sequence Name	Computation Time (seconds)
Laboratory	349
Home	357
Studio	282

After inspecting the bandwidth sizes for the various sequences, they were found to vary between 2 and 25 data points. Larger bandwidths were applied in locations with little variance while smaller bandwidths were applied in locations having larger variance. In order to circumvent the long processing time of adaptive bandwidths this research proposes that fixed bandwidth sizes be used. However the problem of non-convergence and incorrect convergence arises with bandwidths that are too small and too big respectively. The process of converting incorrect convergence would require an in-depth analysis on every point and thus negate any improvement. Merging points that have failed to converge to a mode can be done effectively using simple colour analysis. Table 5.4 demonstrates results of the clustering process using a smaller fixed bandwidth and then correcting for tiny groups of non-converged data points. The correction developed within this work is done by a simple iterative colour merging processing. The results are created using a fixed bandwidth size of 3 with the same suboptimal (K, L) pair depicted in Table 5.2. The shorter computation time associated with the fixed bandwidths depicted in Table 5.4 gives merit to the proposed technique and its corrective cluster merging process. Based on the results and the computation time, it is safe to conclude that utilizing non-optimal parameters for the clustering technique does not severely impact the quality of the clustering. In fact the clustering process is sped up greatly by removing the processes associated to the discovery of optimal parameters.

Table 5.4 - Clustering Results Using Fixed Bandwidths

Sequence Name	Clustered Frame with Adaptive Bandwidths	Clustered Frame with Fixed Bandwidth	Computation Time (seconds)
Laboratory			11
Home			27
Studio			46

5.2.1.3 Comparing FAMS with K-Means Clustering

Within this section the FAMS algorithm is compared with the original K-means clustering technique proposed by Deng *et al.* [3] in their implementation of JSEG. The parameters used in this comparison are suboptimal but allow FAMS to execute within a reasonable amount of time. The comparisons will look at scenes of different complexity and will provide insights to the qualitative results. Figure 5.9 demonstrates the results for several sequences in the laboratory environment.

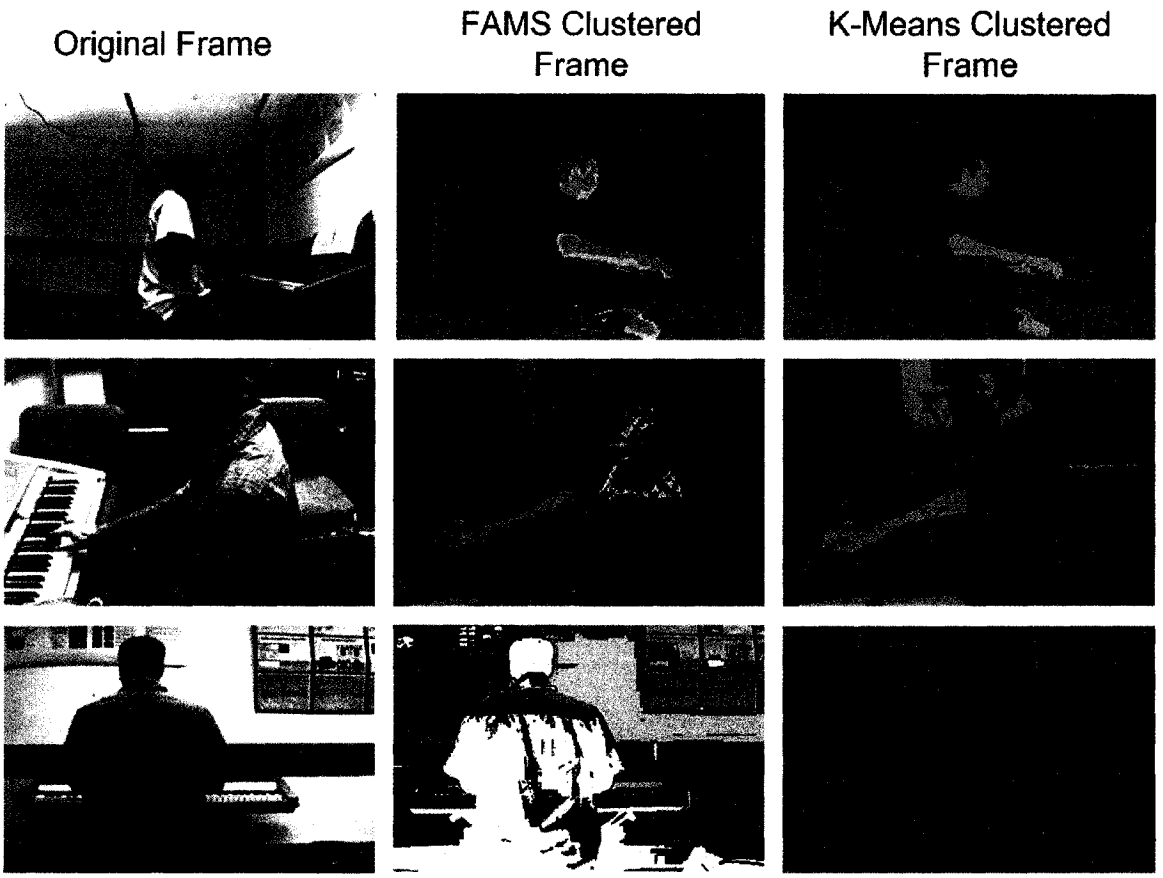


Figure 5.9 - FAMS and K-Means Clustering Comparison in Laboratory Environment

In many cases we find that the K-means clustering algorithm does not succeed in properly capturing some of the more subtle colour features in an image. Since K-means imposes Gaussian-like statistics on colour distributions, many data points end up being classified to a cluster they don't belong to. This phenomenon is observed in the first frame where background shadows along the infrastructure have been added to a distribution different than the wall's distribution. It is also observed on the second frame where the majority of the pianist's head has been merged with surrounding features. Another shortcoming of the K-means algorithm is the fact that it relies on thresholds in order to obtain an appropriate number of clusters, when this threshold is not precisely

tweaked to the frame in question, clusters may end up merged together thus removing many of the image details. The third image shows this observation clearly; FAMS succeeds in producing multiple clusters despite their relative colour distances while K-means only produces two clusters. Since the FAMS algorithm does not rely on spatial data or thresholds to cluster, sets of points may converge to a mode that is different from their neighbours. This observation is seen on the pianist's arms, legs and face in the first and second images. While this type of convergence may add complexity and make interpretation of the results more difficult, it can be dealt with using soft-classification maps as explained in the next section.

As a scene increases in complexity, FAMS' advantage over the K-means algorithm becomes more apparent. Nearby pixels which may seem similar in an image may in fact converge to different modes allowing for a more precise clustering to occur. Figure 5.10 demonstrates how the Gaussian clusters of the K-means algorithm tend to agglomerate all points having similar colour while the FAMS is better able to distinguish colours belonging to different objects. Figure 5.10 distinctly shows a better clustering with FAMS along the pianist's face and hands in the first image, and succeeds in properly distinguishing between the musician's hair, face, left arm, right arm and torso within the second image.

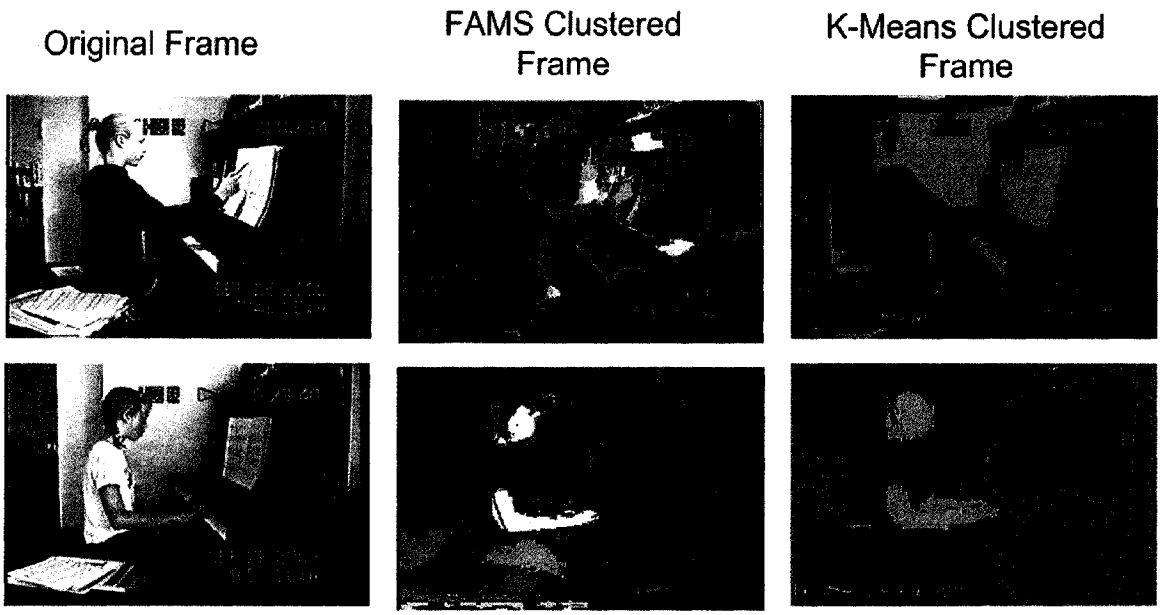


Figure 5.10 - FAMS and K-Means Clustering Comparison in Home Environment

In the more complex studio setting, in Figure 5.11, the same observations can be made. Distinction among colour groups is better achieved with the FAMS algorithm. The improvements in Figure 5.11 are more subtle due the high complexity in the scenes. However, FAMS achieves a less noisy clustering, were smaller and irrelevant background colour distributions do not degrade the overall clustering. In the third image a better distinction of the pianist's arms and head can be observed.

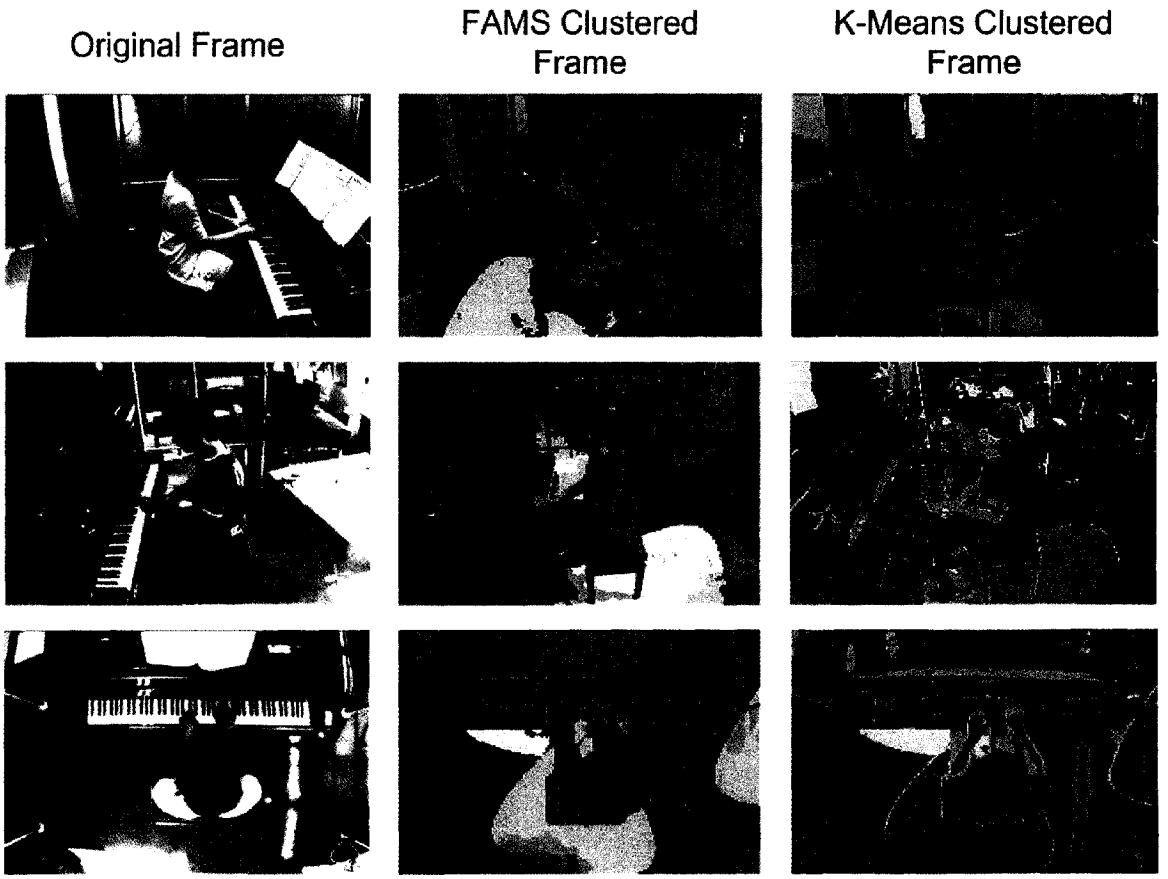


Figure 5.11 - FAMS and K-Means Clustering in Studio Environment

5.2.2 Soft-Classification Results

With the addition of soft-classification maps, membership values can be associated to each label. This association allows a probabilistic representation of the labelled images and consequently a more precise J-value computation. This section demonstrates some results using this type of representation and compares their effects within the J-value segmentation algorithm.

Figure 5.12 depicts the initial clustered image of a Laboratory scene using suboptimal FAMS parameters. Beneath the labelled image the probabilistic representation for a subset of the clusters is given. These clusters represent the majority

of the colours found within the image; the greyscale value of a pixel depicts its membership to the corresponding cluster. A bright pixel represents a strong membership to the cluster, while a dark pixel will only have a weak membership. These representations were created by back-projecting the histogram of a given label into the image. That is to say, the normalized bin value of a given pixel replaces its colour vector, thus giving the probability that the pixel belongs to the label's colour distribution.

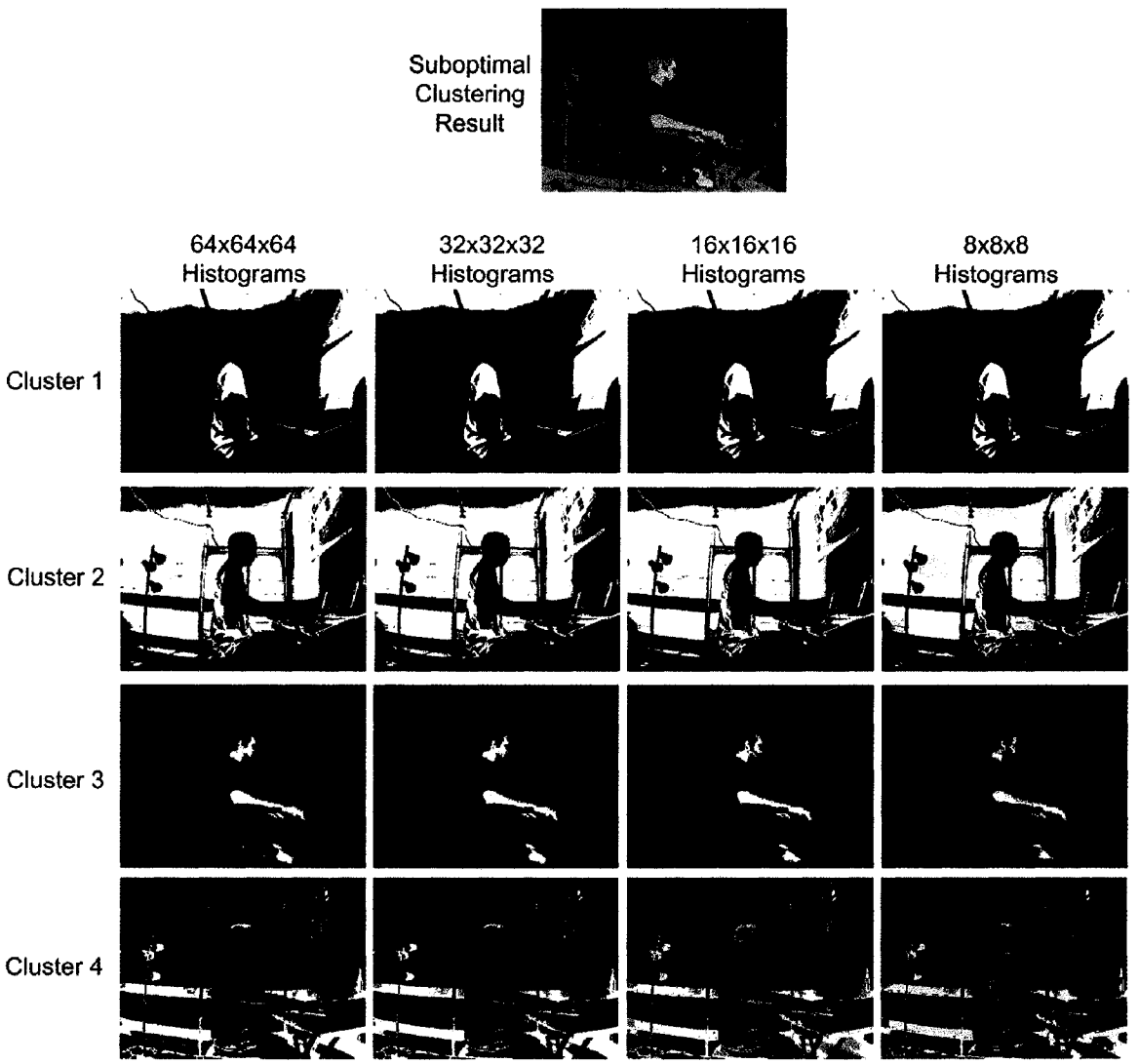


Figure 5.12 - Effect of Histogram Sub-sampling on the Probabilistic Representation of Clusters

Histogram sub-sampling is an important factor in the creation of the soft-classification maps. If the histograms maintain a high number of bins then the non-parametric distributions of the individual clusters will exhibit a minimum of overlap. It is only by generalizing the distributions via histogram sub-sampling that the soft-classification maps can impart membership values on the clusters. As the histograms become coarser, the overlap between distributions is increased yielding classification maps that exhibit a more varied range of values. If the histograms become too coarse however the non-parametric distributions are forced to conform to a set of bins that may misrepresent the distribution. In this case 3D histograms of RGB colours of size 16x16x16 were found to be sufficient in order to provide soft-classification maps without overly misconstruing the cluster distributions.

The application of soft-classification maps to the computation of J-value images improves results. Figure 5.13 shows an example of how these new classification maps impact the J-value images. The goal of computing J-values is to identify homogenous colour-texture regions. Image portions having low degree homogeneity are coloured in white, while dark portions indicated a high degree of homogeneity.

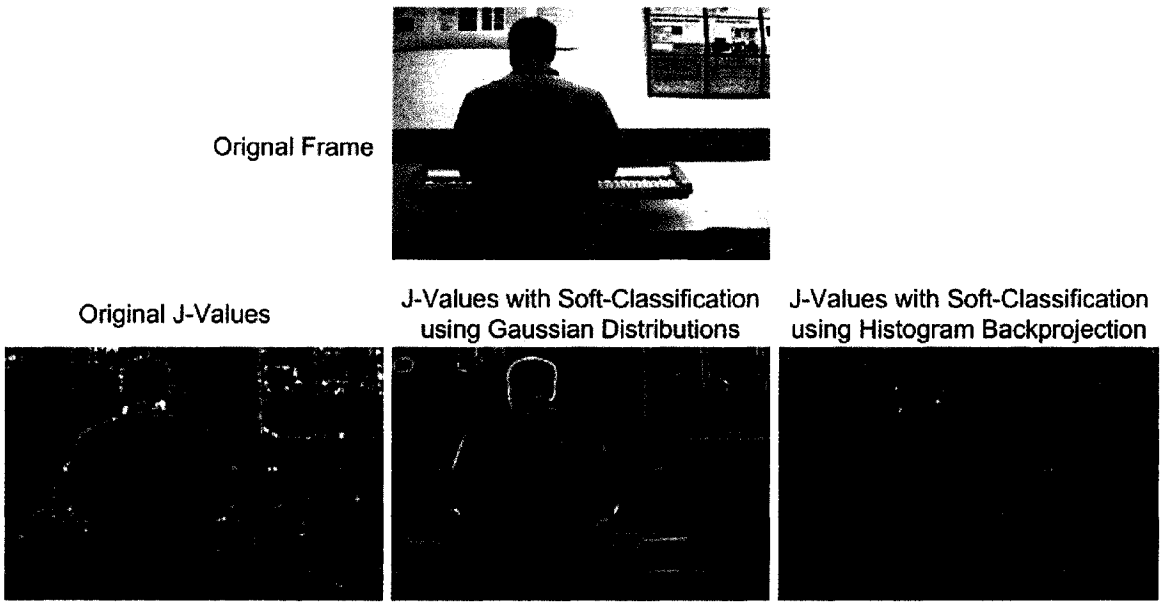


Figure 5.13 - Impact of Soft-Classification Maps on *J*-Value Computations

In the left column the image has several small non-homogeneous portions which in fact can be considered noise. Due to the highly textured scene many of the small subtle colour variances are included in the J-value computations. Lighting changes, colour gradients, small insignificant edges have a large impact in the way the homogenous regions are found and subsequently segmented. Soft-classifications help in removing the noise introduced by these components by allowing labels to have membership values. Even if a pixel is given a certain label, if its membership value is spread out among multiple classes it will likely result in a low J-value. In other words, the soft-classification of the clusters allows for a softening of gradient edges between two very similar clusters. The middle column shows the impact the technique proposed by Wang *et al.* [28] has on J-value computations. The right column shows the results of the improved soft-classification proposed in this work. While both these techniques succeed in their goal of removing non-homogeneous noise, the original soft-classification

by Wang *et al.* [28] has a tendency of over attenuating the values. This attenuation is observed along the musician's shoulders, where the J-value edges have been completely removed. The flexibility of being able to choose the amount of histogram sub-sampling in the proposed technique allows the algorithm to more accurately represent clusters and thus makes it less likely to attenuate key J-value elements.

5.2.3 *J*-Value Segmentation Results

The core JSEG algorithm has not been significantly modified in this work; J-values are computed and regions are produced in the same manner Deng *et al.* [3] have proposed. The FAMS and soft-classification additions described earlier are introduced in order to improve the results of the JSEG algorithm which are now observed in this section. Once again a comparison between the improved technique and the original segmentation algorithm will be provided.

For the results demonstrated here, the kernel and scaling values are the same as suggested by Deng *et al.* [3]. These parameters are reviewed in detail in Appendix C. The only remaining parameter that is left to be tweaked is at which scales the segmentation should occur. A larger scale will result in the discovery of larger and less pronounced homogenous colour-texture regions while at a smaller scale, more defined edges are found. The scales used will modify the size of the kernel and consequently the number of neighbourhood pixels considered in J-value computations. The final J-value will reflect the homogeneity of the neighbourhood pixels taken into consideration. Regions are iteratively split by re-computing J-values within the defined area at a smaller scale and determining if a new set of seeds are present within the region. The

number of scales used will determine how coarse the segmentation will be. Deng *et al.* [3] define an upper bound on the largest scale based on image size. Beyond the upper bound, JSEG is said to no longer provide relevant segmentation information. For the sequences used in this work, that bound is set at scale 3. This means that the *J*-value segmentation will perform an initial region determination with a kernel up-sampled by a factor of 3. These initial regions will be refined by applying a second kernel of scale 2 and repeating the region determination process. Results between the application of 2 and 3 scales are also compared.

The differences in the segmentation results from the improved algorithm and the original segmentation algorithm can sometimes be quite subtle. The improvements are aimed at incorporating sections of an image that exhibit some kind of smooth colour gradient. In Figure 5.14 the segmentation results for a controlled laboratory environment are given. In the first sequence the improved algorithm can clearly be seen from the manner in which the musician's legs and arms have been identified. The background is also better segmented. Since shadows are better incorporated within the clustering process the number of regions is kept small. In the second sequence the major improvement can be seen in the segmentation of the pianist's face. The segmentation of the pianist's right arm is slightly more complex, this is in part due to the manner in which the clustering has occurred. The third sequence shows the most improvement. This sequence was only given 2 clusters when done using a K-means algorithm. The added clusters from the FAMS algorithm vastly improve the segmentation. The musician's head is clearly identified and his body is better segmented from the other surrounding sections. By adding another scale to the segmentation process, regions created at larger

scales are iteratively split based on the J-values computed within them. Little improvement was achieved by adding another scale to the segmentation. The majority of the image details were captured by refining the regions using a kernel of scale 2. At scale 1, regions were split due to small variations in colour-textures and contributed to over-segmenting the image. While JSEG demonstrates the ability to clearly segment semantic image components, the overall results show clear signs of over-segmentation.

Figure 5.14 - Segmentation Results in Laboratory Environment

In the home environment, Figure 5.15, scene complexity is increased significantly. However, the improvements still succeed in producing a better segmentation. In the first sequence, the musician's hand and legs are segmented successfully, while in the original algorithm they are missed altogether. In the second sequence, the musician's torso is clearly defined as are her legs and a distinction between

her left and right arm. The added complexity does contribute to some error in the segmentation. In particular some bleeding effect can be observed. Sections having very similar colour properties to their surroundings tend to bleed into another portion of the image. This can be seen along the pianist's arms and legs in the second image, where regions are slightly misconstrued and include nearby image sections.

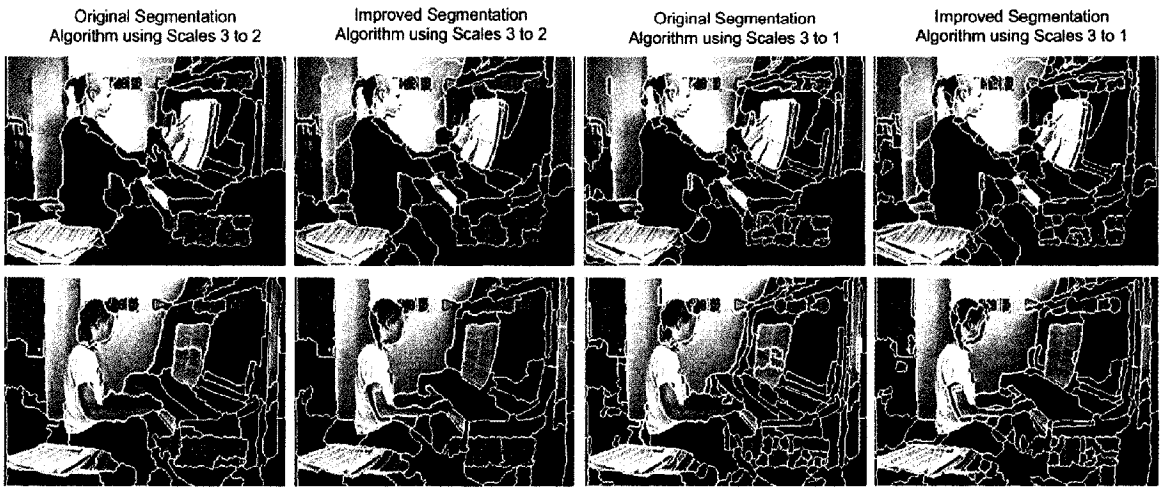


Figure 5.15 - Segmentation Results in Home Environment

In the studio environment, seen in Figure 5.16, the improvements become even more subtle. The increasing complexity of the scene reaches the limit of JSEG's ability to properly create seeds of semantic regions. Some improvements can still be observed, in particular the musician's head and the background floor within the first sequence. The musician's head and arms are also clearly segmented in the third sequence; a significant advantage over the original algorithm. However, due to the high number of textures and edges in the second sequence, it is difficult to come to any type of conclusion.

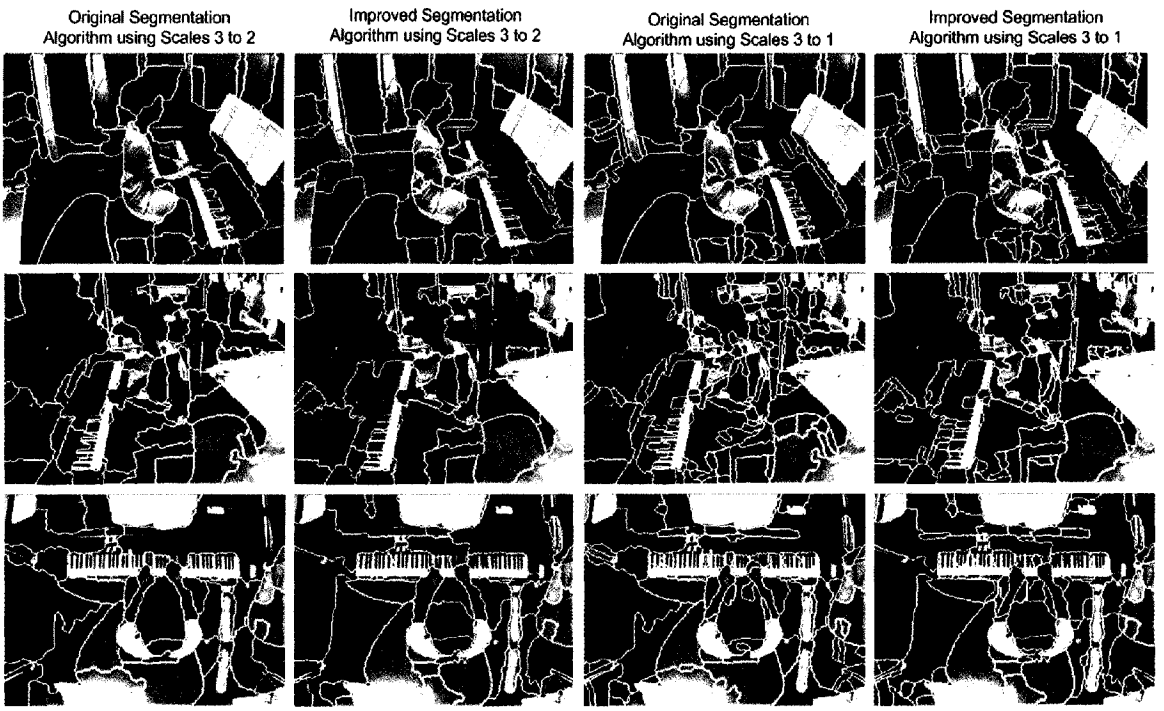


Figure 5.16 - Segmentation Results in Studio Environment

These results demonstrate a definite advantage provided by the FAMS and soft-classification improvements introduced in this work. Only in highly complex scenes does the advantage become less obvious. As mentioned before the lack of improvement can be explained by the fact that JSEG's ability in clearly distinguishing regions in very complex scenes such as the Studio environment remains limited. By extending the segmentation to a smaller scale the refinement allows regions to be split in such a way that semantic image regions are better identified. Regardless of the scale however, a region merging process must be done in order to properly finalize the segmentation.

5.2.4 Merging Results

From the results in the previous section the need for merging similar regions together is apparent. In the original segmentation algorithm a histogram comparison technique was used. Proposed by Hernandez *et al.* [23] and adopted in this work is an iterative joint-criterion merging technique involving both colour and edge information. This section examines how a weight parameter establishing the criteria was chosen as well as a how the new algorithm compares to the original.

The algorithm used in this work must rely on a weight parameter, α, shown in equation (4.24), in order to determine how much importance should be attributed to either the edge or color criteria computed between regions. The other important factor that must be accounted for is the number of regions that should be merged. The impact the weight selection has on the entire merging process is demonstrated in Figure 5.17. In these cases the same total number of regions was merged, thus allowing a more objective view of exactly how the weight parameter changes the final result. The initial segmentation of the frame as well as the number of regions merged is purposefully exaggerated in order to provide better insight as to the effects of the merging. When the weight parameter, α, is low and favours edge data, only regions having a weak boundary are merged together. In the frame where $\alpha = 0.0$ the segmentation quickly merges together regions having larger colour differences but where their boundaries produce a more subtle colour gradient. This is observed along the musician's face and arms. Other regions having stronger boundaries but very similar colour properties are simply left alone; this lack of merging is clearly seen in many aspects of the background. In the frame where only colour data is considered ($\alpha = 1.0$) regions having a distinct border

such as the legs quickly merge to the background due to their surrounding colour properties. When observing the mixture of weights that consider both edge and colour information, a more elevated weight tends to produce better results. In the case seen in Figure 5.17, $\alpha = 0.75$ is clearly the ideal weight.

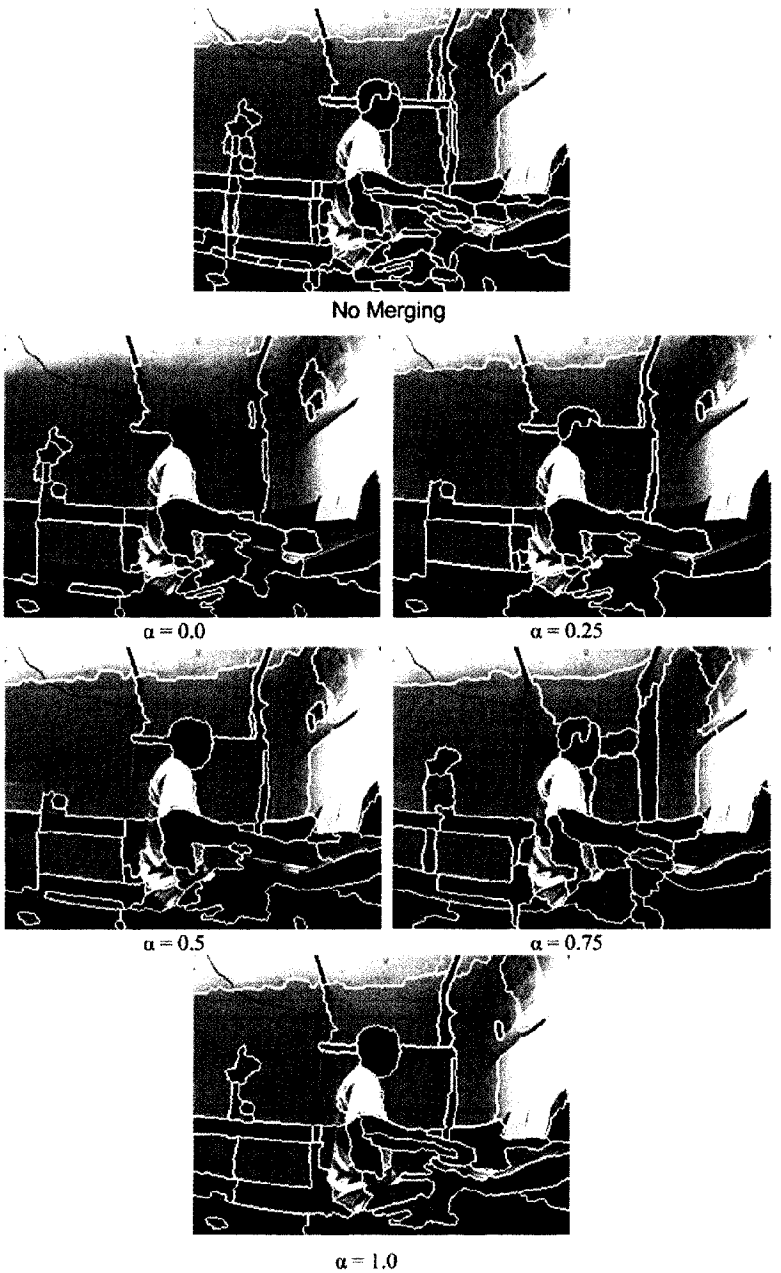


Figure 5.17 - Impact of Weight Selection on Merging Process

Without exaggerating both the segmentation and the merging processes, results presented in Figure 5.18 attempts to compare the algorithm by Hernandez *et al.* [23] to the original merging technique used by Deng *et al.* [3]. The original technique only manages to merge together regions having similar colour properties. The merging process is stopped when the difference in colour surpasses a threshold. In the attempts done below, the threshold had to be set at a very low level or the result would be under-segmented. In most cases, no merging at all gave the best results with the original JSEG. The improved merging process provides a clear advantage for reducing the number of regions. The results demonstrated here were created by manually selecting a weight and number of merging iterations that would provide the best segmentation. This process also allows for a greater flexibility by reducing parameter tweaking in the various segmentation stages prior to the merging.

Figure 5.18 - Merging Result Comparison

In the first frame there is a significant reduction in background regions with the improved approach. Not all of these regions have similar colour properties but they do no have strong boundaries between them. In the second frame the original merging process removes any region of significance with the exception of the pianist's arms and head. Finally, in the third sequence the pianist's arms and head are maintained as a separate region.

5.2.5 Tracking Results

The last remaining process to be analyzed is the tracking algorithm that permits the motion capture of specific body segments in the image. As explained before, the tracking is done using frame blocks and the merge of two separate algorithms to track the regions contained within a block and those contained between blocks. This block-based representation is a new concept proposed in this thesis. The intra-video stack tracking algorithm is the same as the original technique proposed by Deng *et al.* [3]. The inter-video stack algorithm is based on the research done by Withers *et al.* [56] and allows for tracking sequences that cannot be completely stored in memory or analyzed as one large video stack. The algorithm proposed by Withers *et al.* [56] was originally intended for the tracking of cell merging and splitting but is applied here in the context of tracking segmented image regions between video stacks. The results of both these algorithms are observed in this section.

Figure 5.19 and Figure 5.20 show the results of the intra-video stack tracking technique. In these cases blocks were kept at a moderate size with 5 and 4 frames respectively. Since the clustering, soft classification, segmentation and merging techniques all treat the frames as one single large volume the results do not change significantly from one frame to the next. In order for a region to be considered when segmentation occurs, it must be found within each frame of the video stack. The correspondence between regions is hence done implicitly.

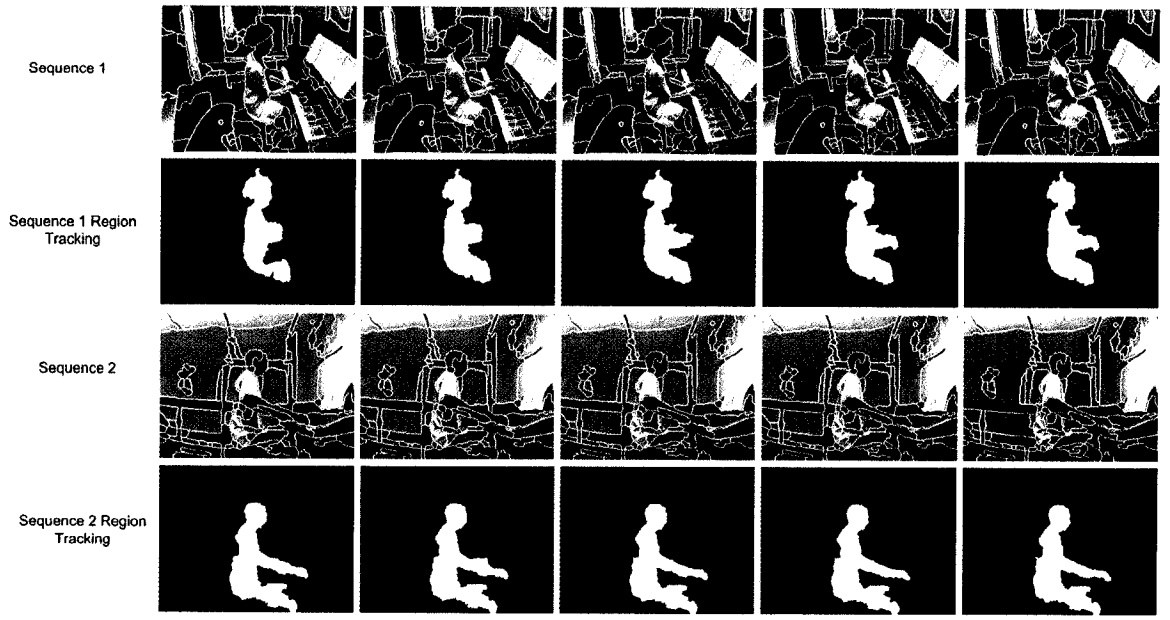


Figure 5.19 - Intra-Block Tracking Results

Figure 5.20 - Intra-Block Tracking Result Comparison

As can be seen, the regions found within a block are very consistent from one frame to the next, allowing for easy video stack tracking. When the tracking process operates on images segmented using Deng *et al.*'s original technique [3], a number of superfluous regions are created. The original technique also fails to map regions with the same level of accuracy provided by the improved technique. Both Figure 5.21 and Figure 5.22 show example results of the inter-video stack tracking algorithm. These results were

generated using a smaller block size of 4 and a correspondence factor, $R_{i,j}$ from equation (4.25), of 0.6. Since the videos depicted in Figure 5.21 and Figure 5.22 have less motion, a smaller block size of 4 frames was selected in order to shorten the processing time. The overall quality of the segmentation does not drastically change between blocks, allowing a high degree of correspondence between regions of different video blocks. The correspondence factor must be tweaked based on the total number of regions. When the number of regions increases, so does the number of potential correspondences between blocks. In order to avoid false correspondences, the factor must be increased in order to provide a more discriminating tracking.

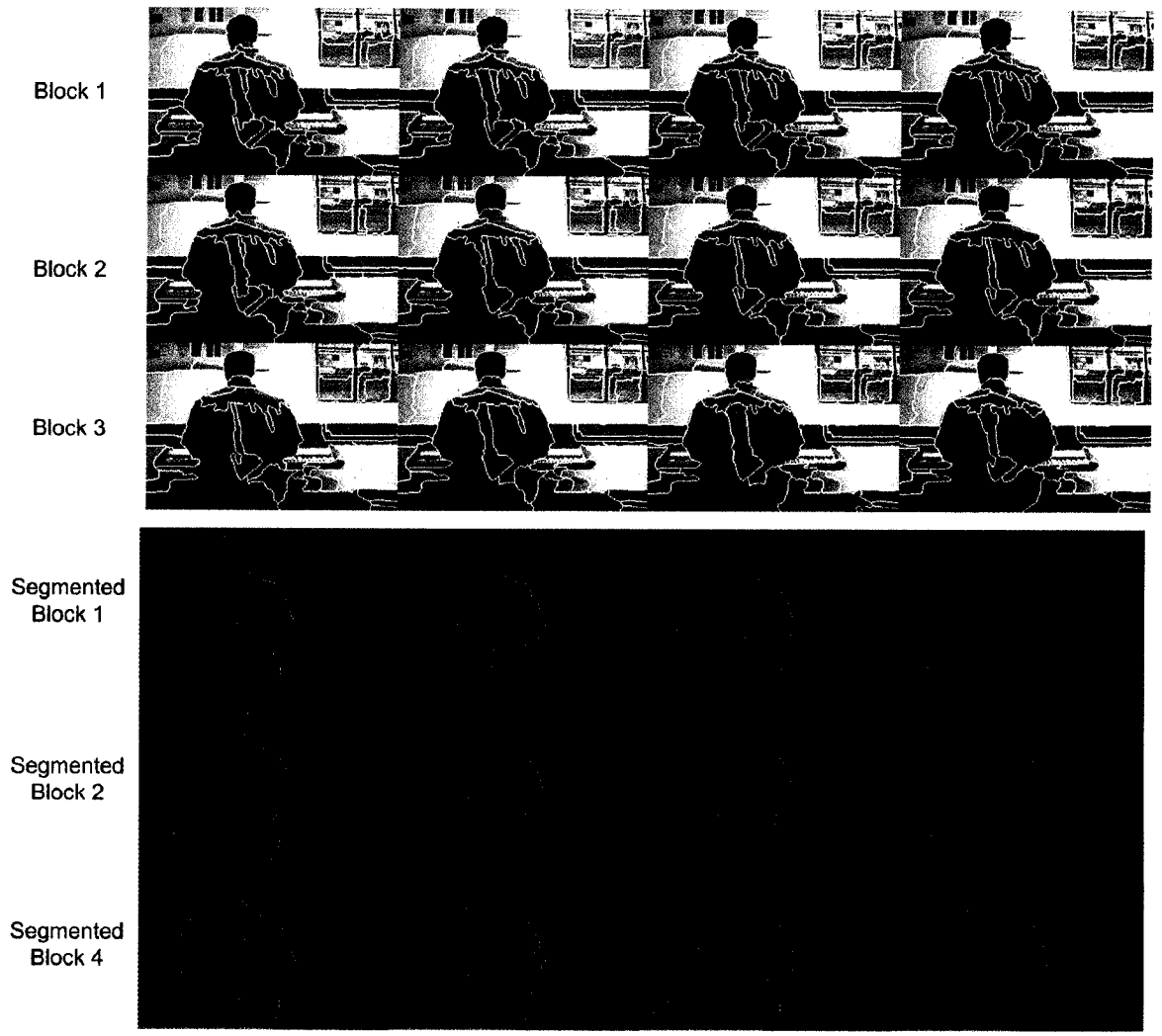


Figure 5.21 - Inter-Video Stack Tracking in Low Motion Scene

Also conveyed in these figures is the algorithm's ability to not only segment a musician but to also allow for the identification of various image sections. In Figure 5.21 both the pianist's head and torso are identified, while in Figure 5.22 the head, right arm and torso are captured. There are also some visible region deformations, in particular around the musician's torso and head in Figure 5.21 and Figure 5.22 respectively. These deformations are due to the algorithm's inability to correctly separate surrounding colours from those on the pianist.

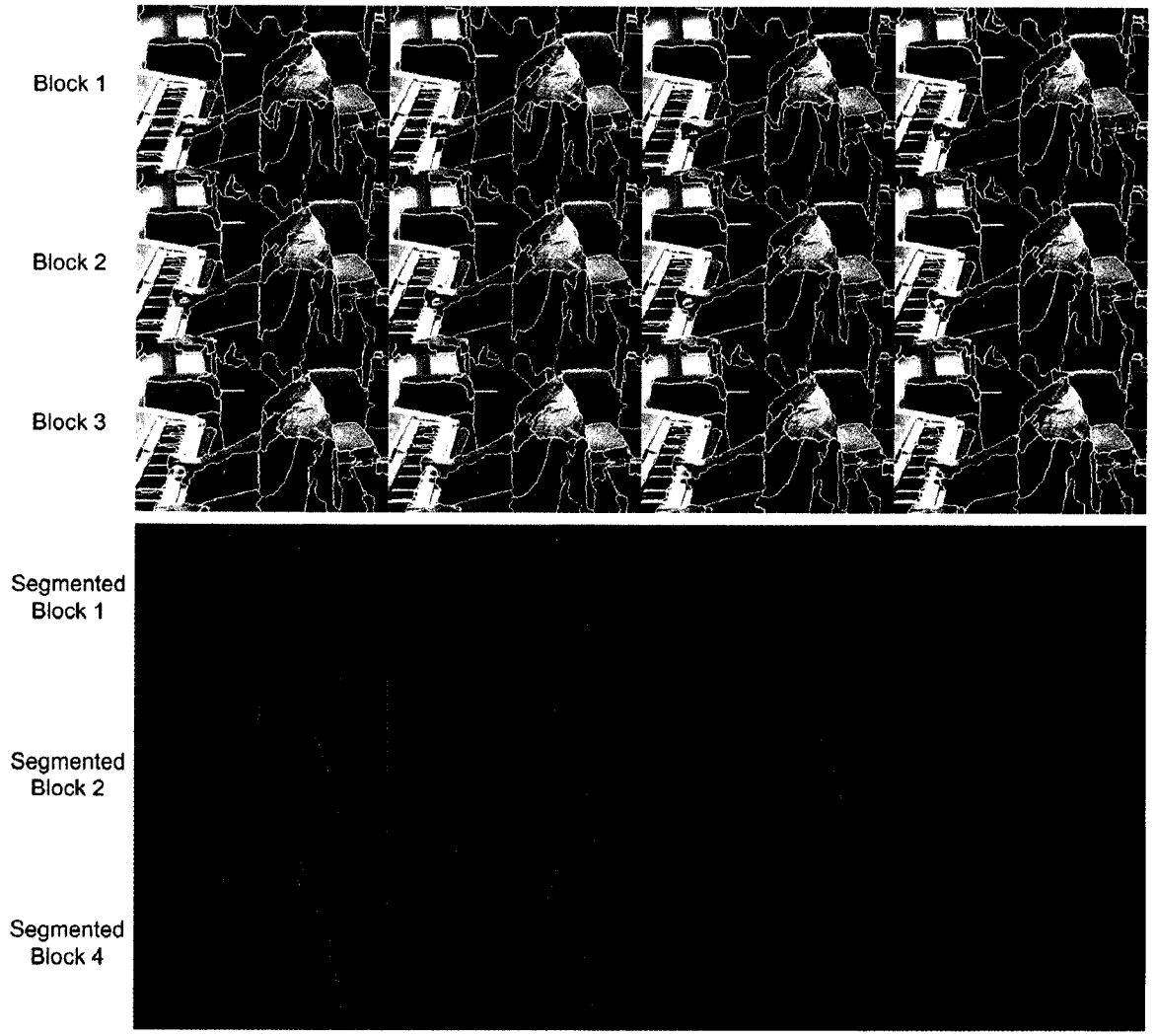


Figure 5.22 - Inter-Video Stack Tracking in Medium Motion Scene

5.3 Experimental Results

The following section of this chapter gives complete results for several videos of different scenes. These videos exhibit different harsh conditions against which the algorithm can be tested with. The laboratory environment provides a scene in which lighting and complexity are controlled while the home and studio environments provide scenes with more varied lighting and a larger number of textures respectively. A look at

both the segmentation and motion capture aspects for the technique will be given. The block sizes used to produce each result is reflected by the number of images per row.

5.3.1 Laboratory Environment Results

The laboratory environment is characterized by a more uniform lighting scheme and a background that is composed of a simpler set of textures. The musician has a clear contrast between himself and the background and no moving components other than the pianist can be observed. The first sequence is used in the segmentation process while the second sequence demonstrates the motion capture abilities of the technique.

Figure 5.23 depicts several frames of the segmented musician in the laboratory environment. The sequence covers several seconds of activity, but in the interest of brevity only sets of frames covering the first couple of seconds are given. The initial set of groups to be segmented throughout the video includes the pianist's head, torso and left arm. The groups are created by a human operator selecting the appropriate segmented regions out of the first frame. From the outset the pianist's general form is clearly segmented. Only the hand exhibits some segmentation fault by incorporating nearby background regions. The cause of this fault, seen in frames 60 to 83, is due to the false merging of the region represented by the musician's watch and the background. There is no pronounced edge between these two regions and the colours are strikingly similar. As the musician lowers his hand, the watch no longer intersects with the troublesome background section and immediately yields better segmentation results. Alternatively, less merging could have been used, thus increasing the total number of regions and processing required, yet improving the results.

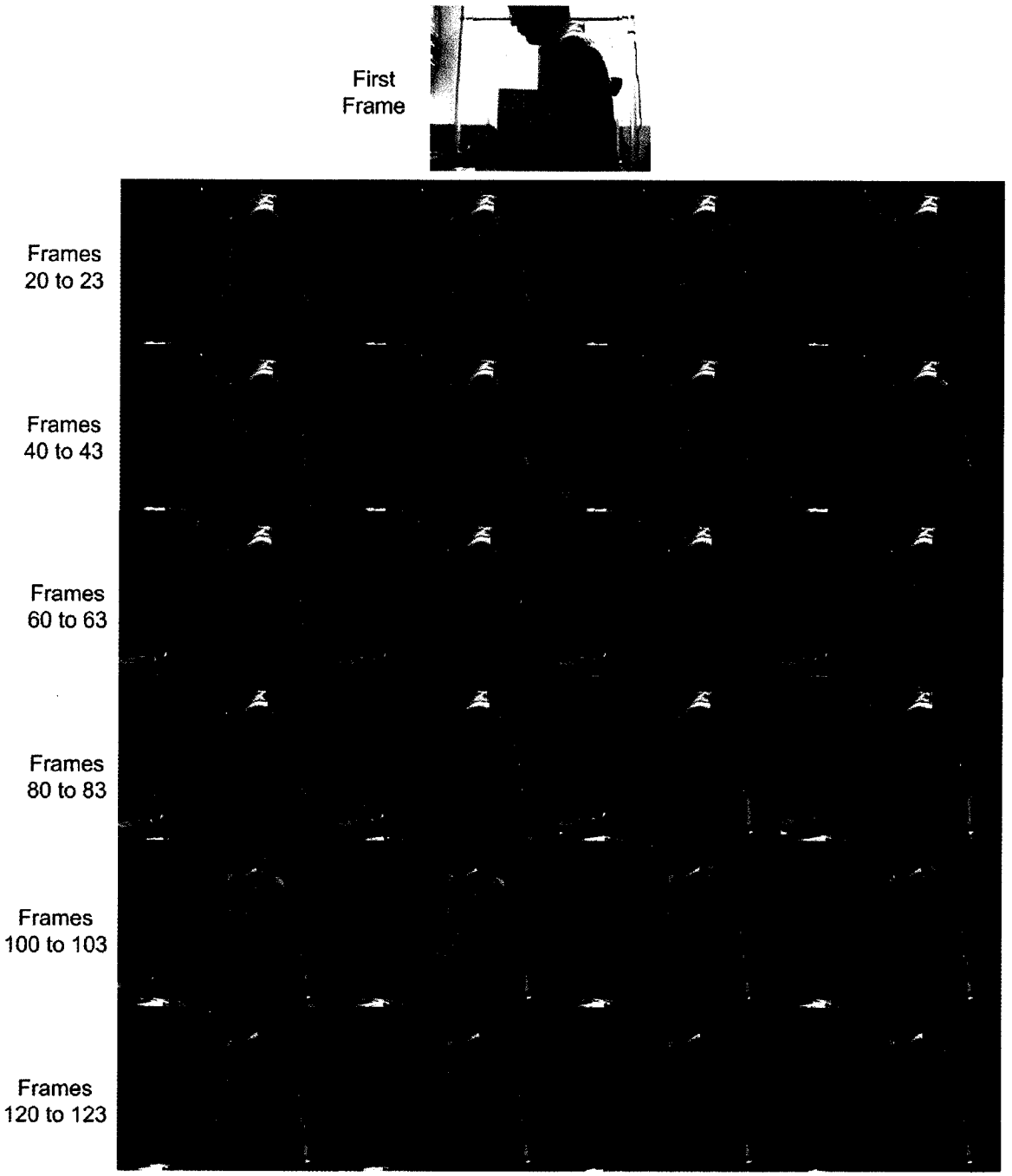


Figure 5.23 - Final Laboratory Segmentation Results

In Figure 5.24, motion capture results in the laboratory environment can be observed. The motion capture attempts to provide insight on the movement of the individual region sets of the segmented sequence. In the sequence the musician's head,

torso and arm are individually tracked. Once again these groups were selected by a human operator after the segmentation of the first frame. In this laboratory sequence lighting is slightly less uniform and creates some amount of confusion with the pianist's clothing. The end result of this confusion is small un-segmented portion of the musician's back. This error however is considered to be small and assumed not to significantly impact any future analysis on the pianist's motions. In fact, despite colour changes to the pianist's shirt due to shadow, and the colour blending between the wall and the shirt, the system succeeds in maintaining a proper segmentation of the musician's torso.

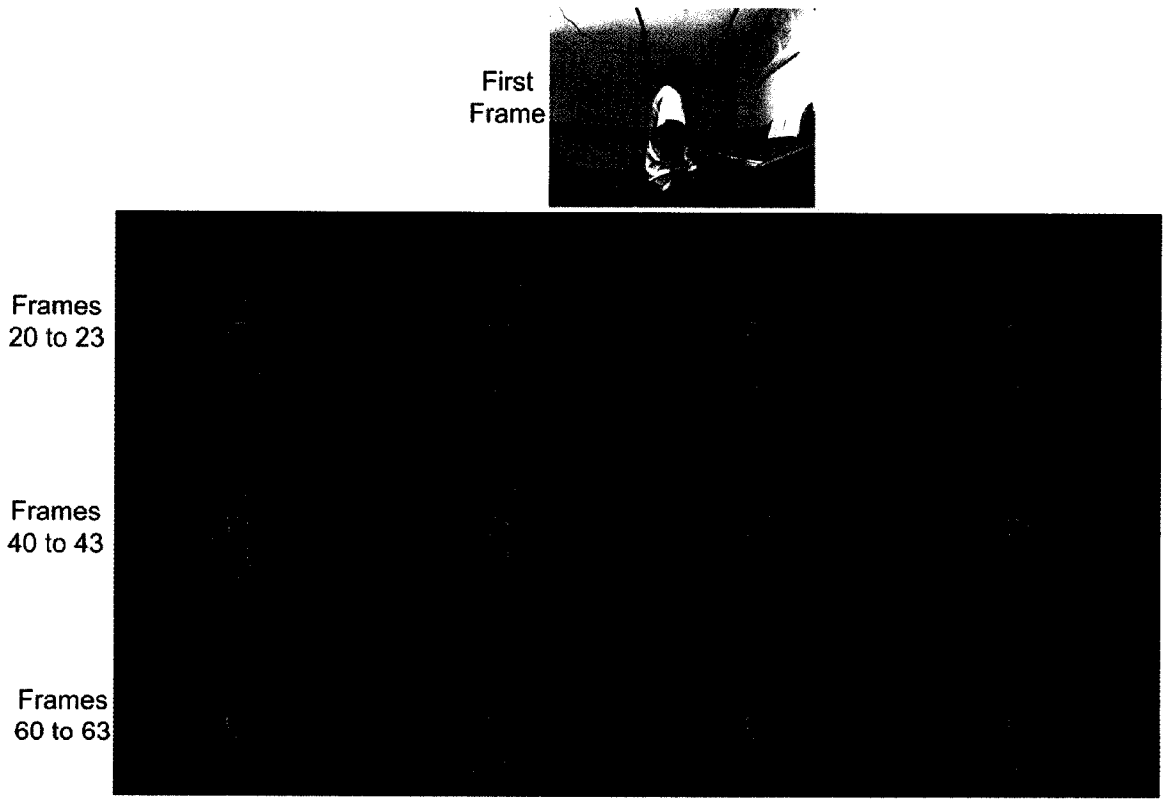


Figure 5.24 - Final Laboratory Motion Capture Results

Both the segmentation and motion capture results are very good within the laboratory environment. With the exception of small inconsistencies the system performs very well. As mentioned before however, the environment is simplistic and the motions of the pianist are small and easy to track. The results presented here clearly indicate that within these types of well lit and controlled environments the system succeeds admirably.

5.3.2 Home Environment Results

The home environment videos were provided by piano students and taken with little regard to the quality and complexity of the overall scene. The sequences are rich in texture and complex lighting effects. The movements exhibited by the musicians are also far more pronounced than the ones performed by the musicians in the laboratory environment. Once again the results of segmentation and motion capture will be given.

The home environment seen in Figure 5.25 is clearly more complex that any of the laboratory sequence. The number of textures and lighting effects is considerably higher; complex colour behaviours including light refractions from the piano and wall can be observed. The goal of the segmentation was to incorporate as much of the musician's body as possible. The head, arm, torso and leg regions were all selected as the segmentation targets. In the initial results the musician's hand cannot clearly be defined with respects to the background music sheet, this creates some error in the final representation. This error is also present when the hand approaches the off-white keys of the piano. It is only when the musician's hand departs from the music sheet and piano keys that it can clearly be segmented. This fault in segmentation is attributed to the clustering algorithm's inability to properly identify the hand pixels as being apart from

the colour distributions of the piano keys and music sheet. One of the reasons why this may have occurred is due to the small number of pixels representing the hand. The lack in pixels translates into a lack of colour components that would have otherwise formed a more distinct colour distribution. This problem could potentially be remedied using very large block sizes in the tracking algorithm, thus increasing the number of pixels belonging to the hand. This solution however would have required considerable resources in order to buffer and process such a large number of data points. Hand segmentation can also be improved by zooming cameras onto regions of interest, expanding the coverage and number of pixels, of specific parts of the body.

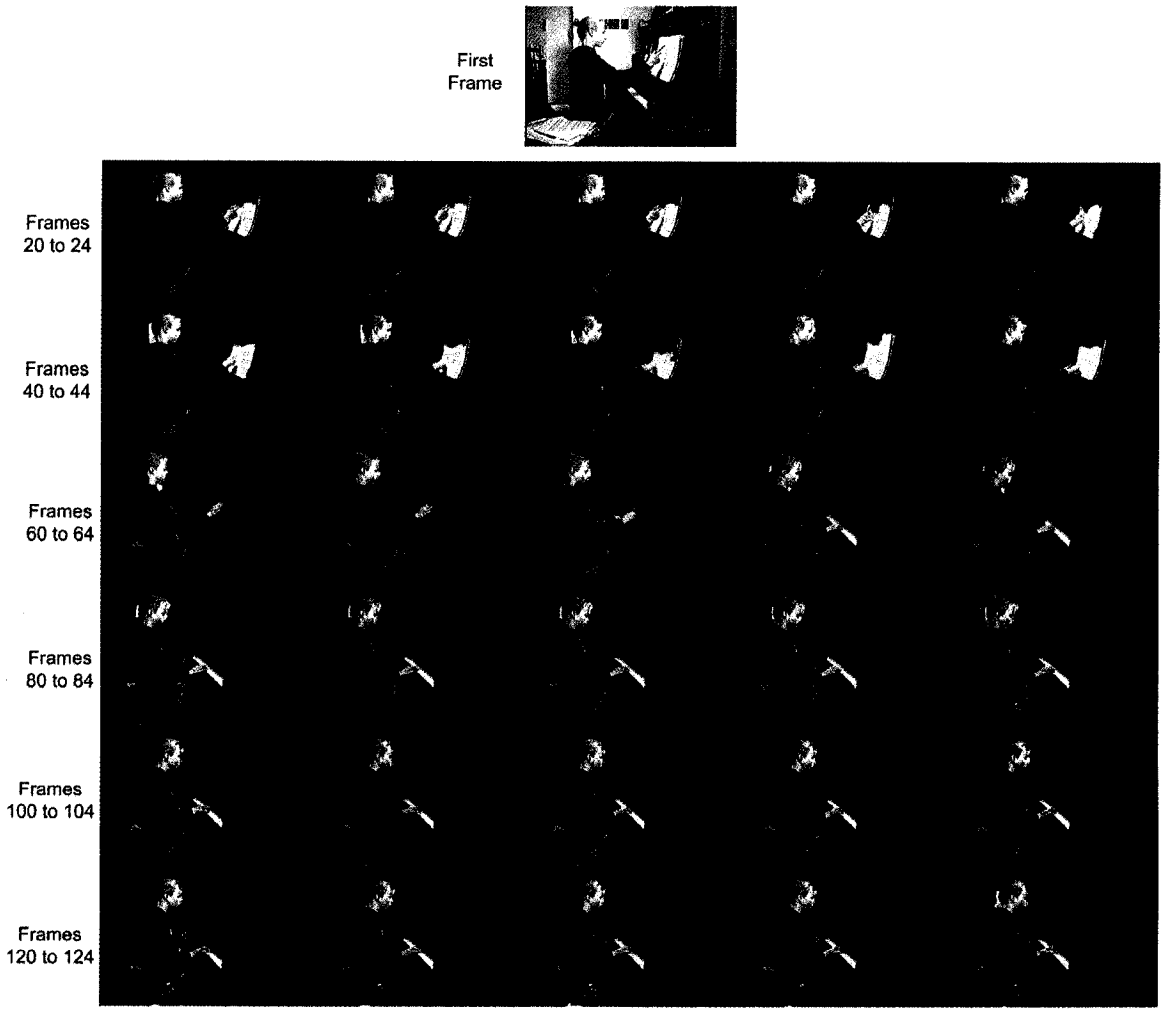


Figure 5.25 - Final Home Segmentation Results

In the home motion capture sequence, the regions selected for capture exhibit rougher segmented edges and less stability. In Figure 5.26, the pianist's head, torso, arms and legs were selected by a human operator out of the initial frame as the motion capture targets. In the sequence the head and torso are fairly well segmented and tracked. With the exception of small irregularities due to the lack of contrast in the background, these targets represent the pianist's body posture very well. The arms and legs on the other hand pose a more serious challenge. These targets have a more complex colour representation and are positioned closed to other similarly complex components such as

the piano. While J-value segmentation typically succeeds well in situation of high texture complexity, in this case it fails to adequately provide well rounded regions for the targets. The arms and legs are still identified and tracked without any significant issues, but their regions remain somewhat misconstrued due to the overall local scene complexity surrounding them.

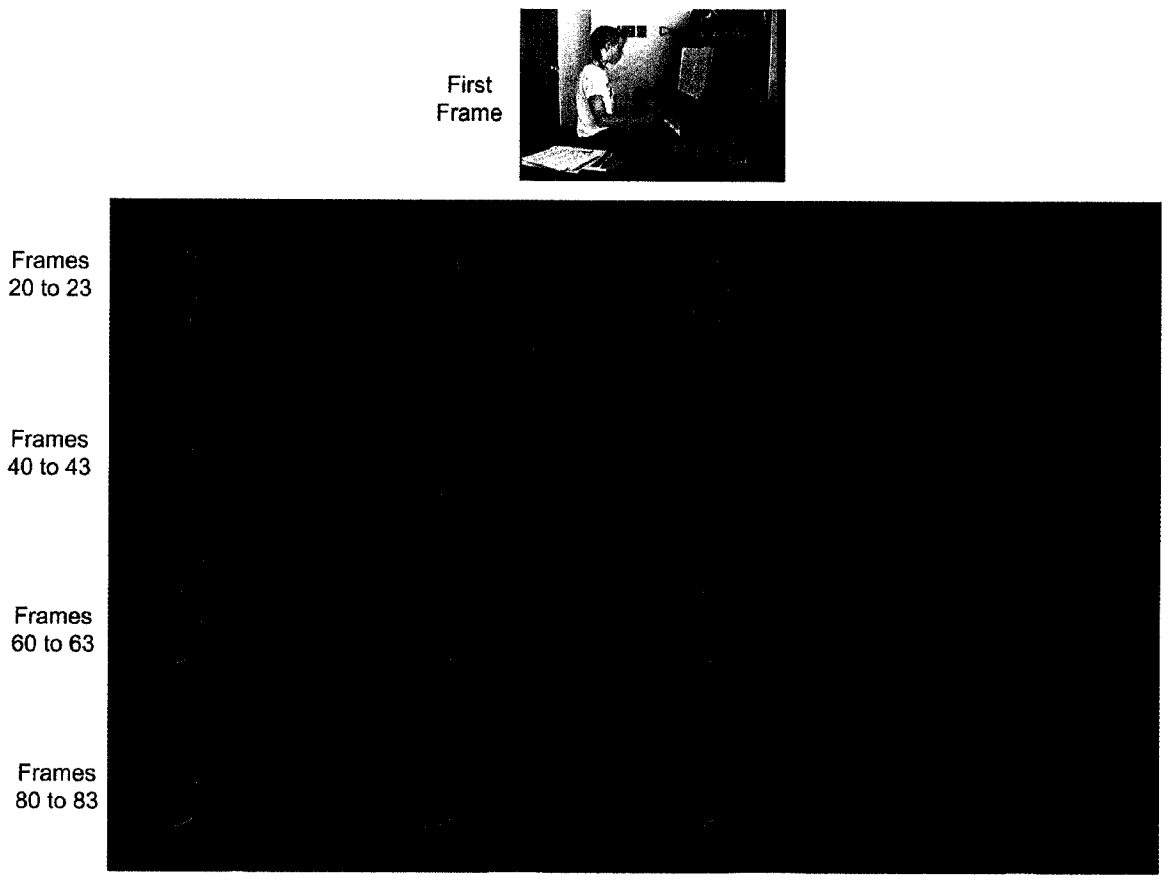


Figure 5.26 - Final Home Motion Capture Results

This section has provided both segmentation and motion capture results for the home environment. As can be observed the overall system scales very well with the added complexity provided by this environment. While some difficulties are encountered

with respects to less discriminating image components the overall results still provide valuable insight as to the overall body posture of the pianist.

5.3.3 Studio Environment Results

The final environment used to the test the system presented within this thesis has the highest level of complexity. The number of textures, lighting effects and motions make for a very difficult scene to segment. The studio environment is used for doing both music recordings and performances; it can be subjected to several different lighting conditions. These complexities are used in order to test the limitations of the segmentation and motion capture algorithms proposed in this work.

The first studio sequence tested for segmentation is seen in Figure 5.27. The sequence has several colourful and rich textures as well as non-uniform lighting, particularly along the background of the scene. In the segmentation attempt, the musician's head, torso, legs and arms were identified for segmentation. In this sequence, the musician's hair was found to be rather difficult to separate from the dark background section. For this reason, it was omitted from the segmentation altogether. Its colour similarities simply did not allow the clustering algorithm to differentiate among the two image components. Similar to the home environment the pianist's hands were difficult to segment apart from the piano keys. The reasons for this lack of distinction are also the same; the insufficient size of the features and the local scene complexity made it difficult to produce two separate regions. In fact throughout the segmentation the hands and arms exhibited unreliable detection. Other components of the target, such as the neck and legs, often included some background components. These misconstrued regions were the

result of small lighting changes and shadow effects that dampened the J-values between image sections. This resulted in a single region on some frames and two separate regions on others. Despite these shortcomings, the algorithm maintains a vast majority of the musician, including the torso, head and right arm throughout the sequence. In frames 120 to 154 the algorithm's ability to handle partial occlusions can be observed. The pianist slowly turns her head away from the camera, obscuring some of the facial features. However, since the face does not completely disappear from view it is successfully segmented throughout the movement.

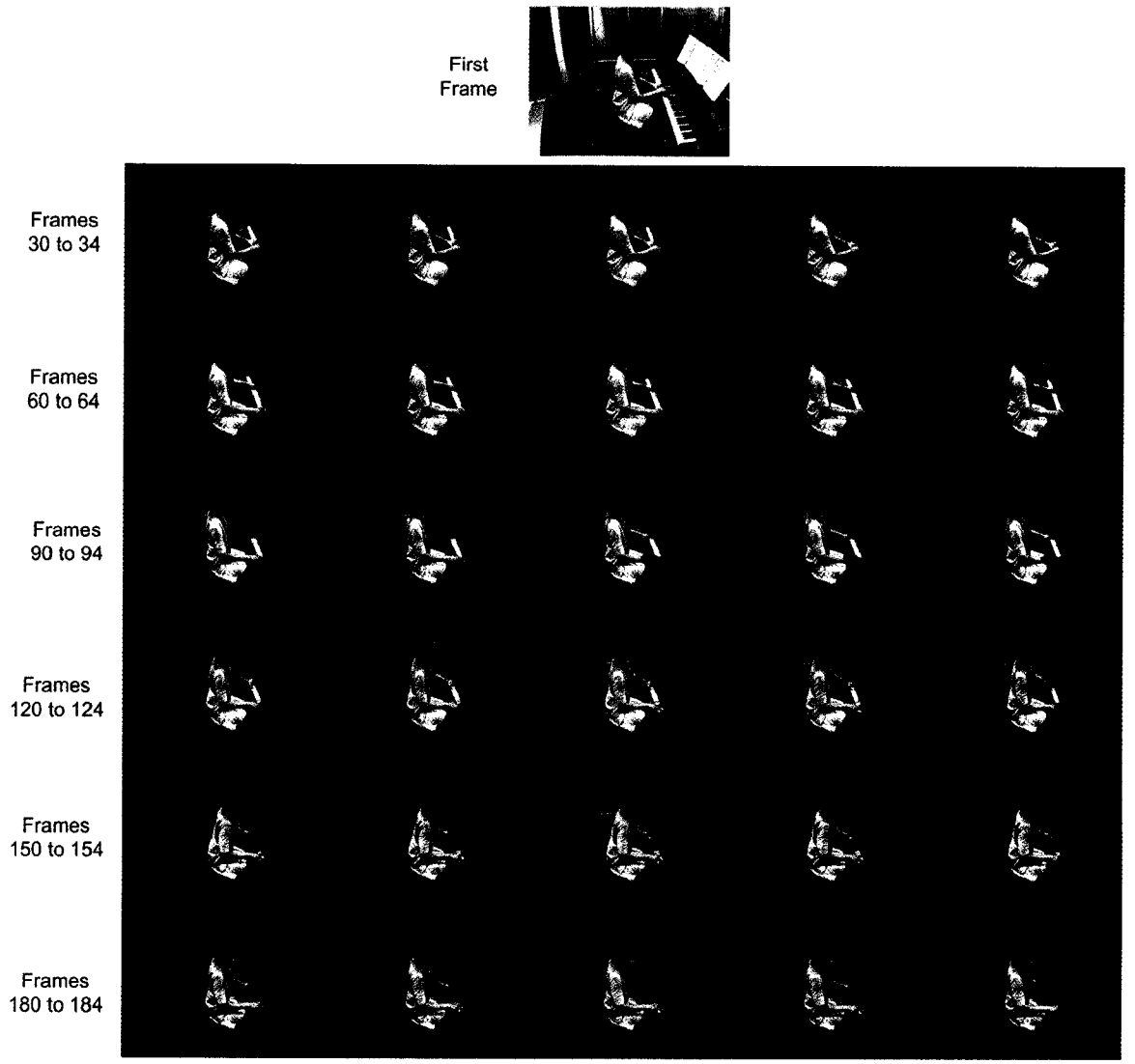


Figure 5.27 - Final Studio Segmentation Results

In the second studio sequence the motion capture aspect of the technique is evaluated. The scene's complexity surpasses that of the original studio video. The scene has several darker textures, small background motions and a complex combination of outdoor and indoor lighting. The motion capture targets in this case were the pianist's head, arms, torso and legs. As can be observed in Figure 5.28, the legs and torso can generally be tracked without too many region inconsistencies. The arms and head

however, often exhibit radical deformations due to changes in local colours and scene complexity. Throughout the entire sequence, the musician's head is misconstrued with nearby image sections having either similar colours or whose texture cannot easily be segmented into a coherent region. In several instances the arms are under-segmented due to the fact that regions are lost in the tracking process. This loss is a direct result of the lack of stability in the region creation process from one video stack to the next. Since each stack must deal with a wide range of colours and motions the segmentation results can vary quite a bit and impede proper inter-video stack tracking. In this case the algorithm does not necessarily provide the best representation of the motions exhibited by the smaller scene components. For more exact results in such a complex environment the target regions should be constrained to image parts that are more prominent.

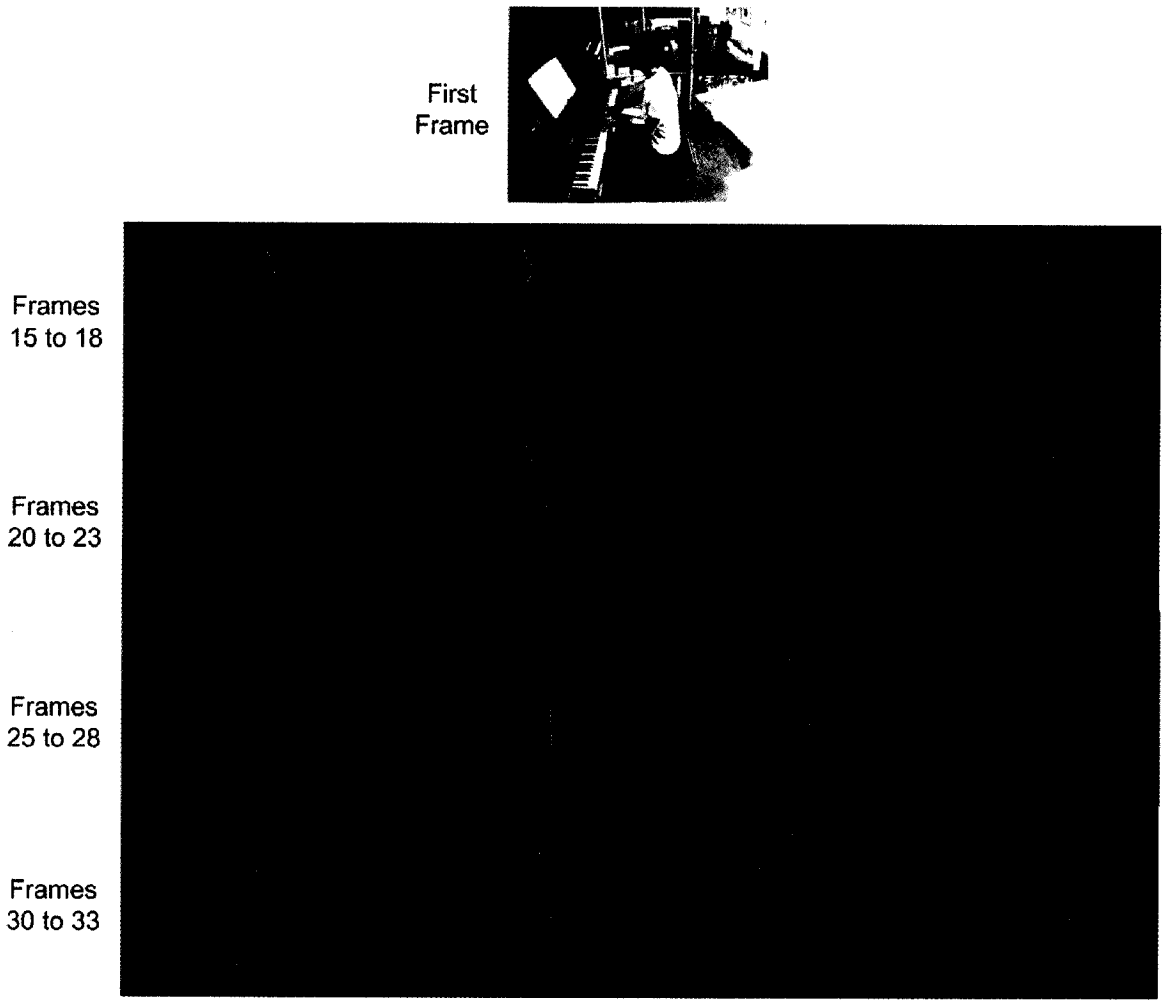


Figure 5.28 - Final Studio Motion Capture Results

The studio environment provides several challenges to the technique presented here. Despite the richly coloured scenes, the algorithm still scales well to the added complexity. Smaller image components however are made very difficult to identify and track, especially when nearby image sections are of high complexity. In such cases the field of view of the cameras should be constrained and only take into consideration the more prominent features of a sequence.

5.4 Experimental Results in Other Contexts

The following section examines the applicability of the technique described here in other contexts besides piano playing. The goal of this analysis is to further strengthen the underlining objective in order to provide a framework appropriate to unconstrained environments. In order to achieve this, the technique has been tested against a video that is outside the stated test cases and in an entirely different environment.

The selected video sequence for this test depicts a professional hockey match and demonstrates the framework's adaptability and scalability towards environments that exhibit a much higher level of motion. This context also poses several tracking challenges stemming from the frequent interactions between players. For this situation, background subtraction methods would outright fail due to the motion of the camera viewpoint and individuals in the background. Figure 5.29 depicts the final results of the technique.

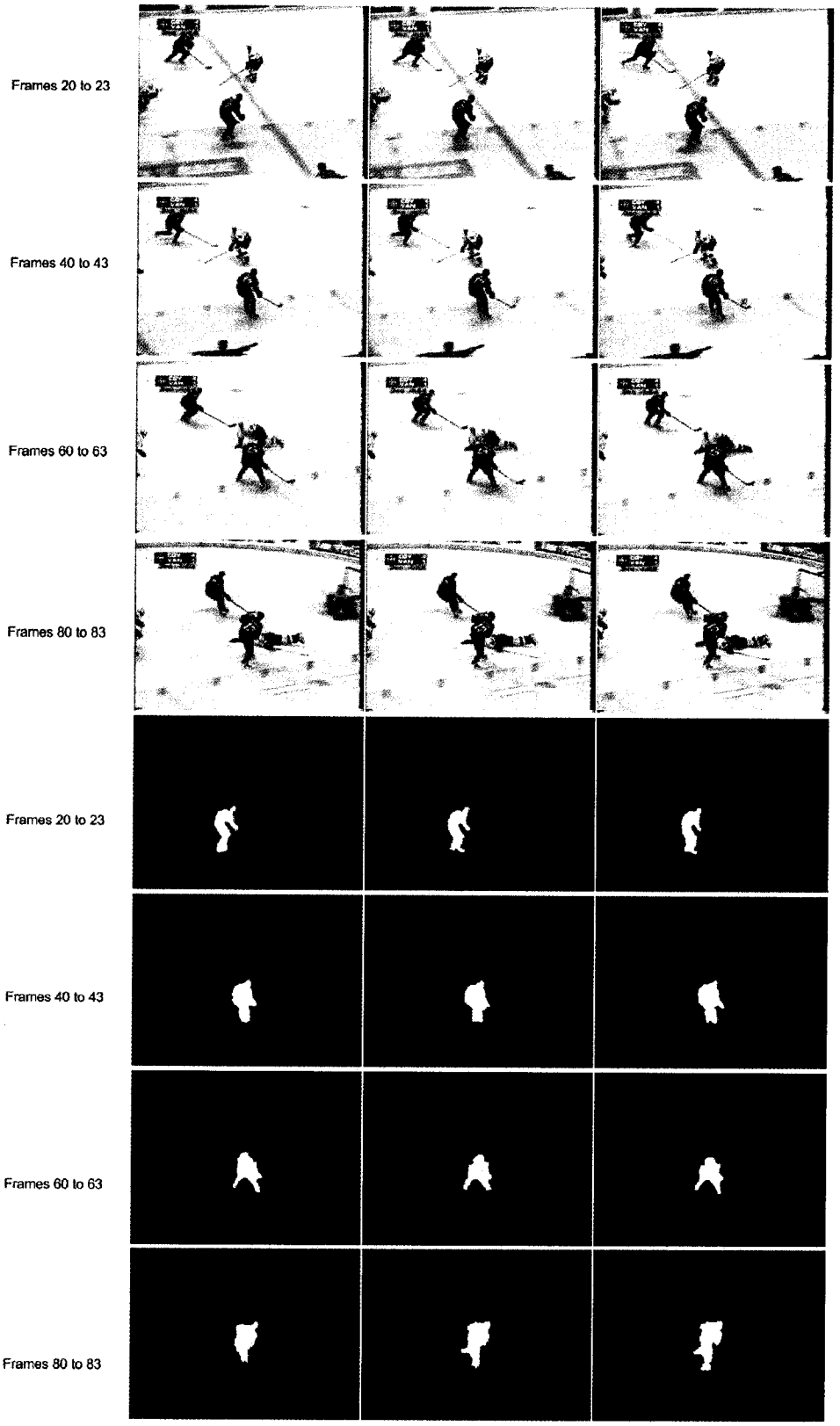


Figure 5.29 - Results From Video in a Different Context

The results from this test clearly show just how well the technique presented within this thesis can be applied to various contexts. The segmentation and tracking of the selected hockey player is good despite the high speed at which both the individual and the camera is moving. Interactions with other nearby players aren't enough to deter the tracking of the appropriate regions. There is however some loss of detail within the final results; this is mostly due to the fact that the target of interest occupies a very small portion of the overall view. Overall, the application of this technique in a context other than piano playing was successful.

5.5 Chapter Summary

The results demonstrated in this chapter clearly show how the technique explained within this thesis can be used in order to segment and capture motion from performing musicians in different environments without markers or specific dress code. These results also show how the algorithm scales well in light of several added scene complexities. It is only when the scenes become overwhelmingly complicated that smaller areas of interest can no longer reliably be tracked or segmented. The comprehensive analysis of the algorithm has also showed how the improvements introduced throughout this thesis do in fact help Deng *et al.*'s [3] original JSEG algorithm surpass its initial shortcomings and make it applicable for motion capture.

Chapter 6 Conclusion

The final chapter will cover three sections. The first section summarizes the research and infrastructure presented within this work. The second section reviews the contributions of the overall work. The final section discusses and provides insight on possible future work.

6.1 Summary

This thesis presented an approach for a motion capture system that uses only passive vision technologies. The goal of this system is to provide the means with which human performance evaluations can be done without the need for cumbersome and interfering technologies. The research was presented in the context of piano playing, where every year professionals succumb to serious repetitive stress injuries. The capture of motions in this context must be done without imposing constraints on the musician or his environment. Such constraints could only impede a musician's true performance and serve to invalidate the captured data.

A review of the traditional motion capture systems was presented in Chapter 2. Many of these traditional techniques used cumbersome devices that had to be worn by performers in order to acquire data. Infra-red and magnetic trackers attached to individuals only inhibit their natural motion and are not appropriate for the context presented here. Passive technologies were also reviewed. The application of thresholds, background modelling, contour, statistical, and region-based methods were all found to either be inadequate for complex environments or relied on assumptions with regards to

the movement. As such the vast majority of existing techniques could not work well in the unconstrained setting required here.

A subset of the techniques reviewed was tested in the typical environments in which musicians performed with increasing complexity. While some of the techniques showed promise, such as the mixture of Gaussians and Continuous Adaptive Mean-Shift (CAMSHIFT), their application to more complicated environments or to a more varied set of motions was limited. Even with several improvements to the CAMSHIFT algorithm, the technique would not be able to handle the type of scenes used in this work without requiring a serious reworking of its founding principles.

The region-based algorithm, JSEG, developed by Deng *et al.* [3] used a novel colour-texture homogeneity measure with which semantic image regions could be identified. This technique became the starting point for much of the research presented here. Applied to their original algorithm were the improvements suggested by Wang *et al.* [28]. In addition to these improvements a non-parametric description of colour clusters, a merging technique based on watershed segmentation and a block-based tracking algorithm were also introduced.

These improvements, described in Chapter 4, were found to give a significant advantage over the original JSEG algorithm. They extended the applicability of the original algorithm from simple textured environments to the complex scenes surrounding musicians during repetitions. The improvements also allowed for individual components of a scene to be tracked, thus allowing individual motions from a musician to be observed. The final study on of several different scenes demonstrates the flexibility of the proposed algorithm as well as its scalability in light of additional scene complexities.

6.2 Contributions

This thesis introduces a new method for capturing and tracking key performance motions within complex environments using a purely non-invasive technique. The algorithm utilizes a region-based segmentation algorithm in order to identify semantic image components. This segmentation makes use of a local homogeneity criterion in order to produce these semantic regions. The algorithm also allows tracking by finding correspondences between image frames using a temporal homogeneity criterion.

The original algorithm [3] was improved using a non-parametric clustering technique as well as an equivalent non-parametric representation of the clusters in order to provide the underlining data set for the segmentation. The use of these techniques allows for more dynamic image data. A flaw in the original algorithm that produced an over-segmentation of the images was corrected using a joint edge and colour criterion for merging similar regions. Likewise, since the original algorithm did not allow for online region tracking, a block-based algorithm is applied in order to modify the existing methodology. This inter-video stack tracking algorithm finds region correspondences between blocks by taking into account various region transformations that may have occurred.

Many of the additions to the original algorithm stem from previous works in various domains of computer vision. The combination of these techniques provides a new and innovative way of performing segmentation and motion capture. In the case of the FAMS algorithm, some modifications were done in the manner in which optimization parameters were applied. A new way of using soft-classification maps based on histograms was proposed by this work and shown to be successful. The use of multiple

video tracking algorithms in a block-based configuration was also presented within this thesis and proven to perform well.

These additions were found to improve the overall segmentation of complex images as seen in Chapter 5. A better overall identification and tracking of human targets was achieved as demonstrated by experimental comparison with state-of-the-art segmentation and tracking techniques. Individual region sets were tracked throughout a sequence in order to allow motion capture of human performance.

6.3 Future Work

The algorithm proposed in this work is successful in providing the means with which a performer's motions can be captured. The system however, is dedicated to capturing human performances. The fact that humans will always be the targets of interest can be an advantage. Skeletal and kinematic models could be used to assist the segmentation and tracking process. By fitting a model to the visual data, a refinement on the image regions can be performed in order to improve the overall results. The restrictions imposed by these models would also help the tracking process by limiting its search area when finding region correspondences in subsequent frames. The system is also limited to the two-dimensional case. In a performance a full three-dimensional view of the movement would provide a better and more informed analysis. Already the physical infrastructure for a multi-view acquisition of human performance has been built. The segmentation algorithm could benefit from the additional information provided by the 3D coordinates of points across multiple cameras. This of course would require the construction of dense depth maps for each point of view. Both the clustering and

segmentation algorithms could be expanded in order to take advantage of the 3D colour distributions in order to provide a far more discriminative segmentation among the components in a scene. This would also eliminate misconstrued regions due to colour-texture similarity with other background regions.

References

[1] A. Sundaresan, and R. Chellappa, "Markerless Motion Capture using Multiple Cameras," *Proc. of the Computer Vision for Interactive and Intelligent Environment*, pp. 15-26, Lexington, Kentucky, Nov. 2005

[2] D. Russell, "Establishing a Biomechanical Basis for Injury Preventative Piano Padagogy", *Recherche en Éducation Musicale*, no. 24, pp. 105-118, Aug. 2006.

[3] Y. Deng, and B. S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *in IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800-810, Aug. 2001.

[4] B. Delaney, "On the Trail of the Shadow Woman: The Mystery of Motion Capture," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 14-19, Sept. 1998.

[5] S. Yabukami, H. Kikuchi, M. Yamaguchi, K. I. Arai, K. Takahashi, A. Itagaki, and N. Wako, "Motion Capture System of Magnetic Markers Using Three-Axial Magnetic Field Sensor," *in IEEE Trans. on Magnetics*, vol. 36, no. 5, pp. 3646-3648, Sept. 2000.

[6] S. Hashi, M. Toyoda, S. Yabukami, K. Ishiyama, Y. Okazaki, and K. I. Arai, "Wireless Magnetic Motion Capture System—Compensatory Tracking of

Positional Error Caused by Mutual Inductance," *in IEEE Trans. on Magnetics*, vol. 43, no. 6, pp. 2364-2366, June 2007.

[7] Vicon Peak, *Vicon Motion Capture System*, Lake Forest, CA, 2005. http://www.vicon.com.

[8] S. Drouin, P. Hébert, and M. Parizeau, "Simultaneous Tracking and Estimation of a Skeletal Model for Monitoring Human Motion," *Proc. of the 16th Conference on Vision Interface*, pp. 81-88, Halifax, NS, June 2003.

[9] N. Otsu, "A Threshold Selection Method from Gray Level Histograms," *in IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62-66, Mar. 1979.

[10] C. Stauffer, and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 246-252, Ft. Collins, CO, June 1999.

[11] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.

[12] T. Horprasert, D. Hardwood, and L. S. Davis, "A Robust Background Subtraction and Shadow Detection," in *Proc. of the 4th Asian Conference on Computer Vision*, vol. 1, pp. 983-988, Taipei, Taiwan, Jan. 2000.

[13] S. Atev, O. Masoud, and N. Papanikolopoulos, "Practical Mixtures of Gaussians with Brightness Monitoring," in *Proc. of the 7th IEEE International Conference on Intelligent Transportation Systems*, pp. 423-428, Washington, DC, Oct. 2004.

[14] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," in *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, 1987.

[15] C. Gu, and M.-C. Lee, "Semiautomatic Segmentation and Tracking of Semantic Video Objects," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 572-584, Sept. 1998.

[16] N. Peterfreund, "Robust Tracking of Position and Velocity with Kalman Snakes," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 564-569, June 1999.

[17] S. Sun, D.R. Haynor, and Y. Kim, "Semiautomatic Video Object Segmentation Using VSnakes," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 75-82, Jan. 2003.

[18] L. Vincent, and P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583-598, June 1991.

[19] D. Wang, "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539-546, Sept. 1998.

[20] S.-Y. Shien, Y.-W. Huang, and L.-G. Chen, "Predictive Watershed: A Fast Watershed Algorithm for Video Segmentation," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 5, pp. 453-461, May 2003.

[21] Y.-P. Tsai, C.-C. Lai, Y.-P. Hung, and Z.-C. Shih, "A Bayesian Approach to Video Object Segmentation via Merging 3-D Watershed Volumes," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 175-180, Jan. 2005.

[22] K. Haris, S. N. Efstratiadis, N. Maglaveras, and A. K. Katsaggelos, "Hybrid Image Segmentation using Watershed and Fast Region Merging," in *IEEE Trans. on Image Processing*, vol. 7, no. 12, pp. 1684-1699, Dec. 1998.

[23] S. E. Hernandez, and K. E Barner, "Joint Region Merging Criteria for Watershed-Based Image Segmentation," in *Proc. of the 2000 International Conference on Image Processing*, vol. 2, pp. 108-111, Vancouver, BC, Sept. 2000.

[24] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E Rogowitz, "Adaptive Perceptual Color-Texture Image Segmentation," *in IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1524-1536, Oct. 2005.

[25] T. N. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," in *IEEE Trans. on Signal Processing*, vol. 40, no. 4, pp. 901-914, Apr. 1992.

[26] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, "Peer Group Filtering and Perceptual Color Image Quantization," in *Proc. of the IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 21-24, Orlando, FL, June 1999.

[27] C. Qiu Xiao, L. Jian Cheng, and Z. Cheng Hu, "Multispectral Satellite Imagery Segmentation Using A Simplified JSEG Approach," in *Proc. of SPIE on Applications of Digital Image Processing XXVII*, vol. 5558, pp. 853-861, Denver, CO, Nov. 2004.

[28] Y. Wang, J. Yang, and P. Ningsong, "Synergism in Color Image Segmentation," in *Proc. of the 8th Pacific Rim International Conference on Artificial Intelligence*, vol. 3157, pp. 751-759, Auckland, New Zealand, Aug. 2004.

[29] B. Georgescu, I. Shimshoni, and P. Meer, "Mean Shift Based Clustering in High Dimensions: A Texture Classification Example," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 456-463, Nice, France, 2003.

[30] F. Bayoumi, M. Fouad, and S. Shaheen, "Based Skin Human Detection in Natural and Complex Scenes," in *Proc. of the 46th IEEE International Midwest Symposium on Circuits and Systems*, vol. 2, pp. 568-571, Cairo, Egypt, Dec. 2003.

[31] M.J. Jones, and J. M. Rehg, "Statistical Color Models with Application to Skin Detection," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 274-280, Ft. Collins, CO, June 1999.

[32] L. Sigal, S. Scarloff, and V. Athitsos, "Skin Color-Based Video Segmentation under Time-Varying Illumination," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 832-877, July 2004.

[33] J. Yang, W. Lu, and A. Waibel, "Skin-Color Modeling and Adaptation," in *Proc. of the 3rd Asian Conference on Computer Vision*, vol. 2, pp. 687-694, Hong Kong, 1998.

[34] W. Du, and H. Li, "Vision Based Gesture Recognition System with Single Camera," in *Proc. of the 5th International Conference on Signal Processing*, vol. 2, pp. 1351-1357, Beijing, China, Aug. 2000.

[35] D. Comaniciu, and P. Meer, "Robust analysis of feature spaces: color image segmentation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 750-755, San Juan, Porto Rico, June 1997.

[36] G. R. Bradski, "Computer Vision Face Tracking for Use in a Perceptual User Interface," in *Intel Technology Journal*, vol. 2, no. 2, 1998.

[37] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces," in *Proc. of the Pan-Sydney Area Workshop on Visual Information Processing*, vol. 10, pp. 3-7 , Darlinghurst, Australia, 2004.

[38] M. J. Swain, and D. H. Ballard, "Indexing Via Color Histograms," in *Proc. of the 3rd International Conference on Computer Vision*, pp. 390-393, Osaka, Japan, 1990.

[39] B. Schiele, and J.L. Crowley, "Recognition without Correspondence using Multidimensional Receptive Field Histograms," in *International Journal of Computer Vision*, vol. 36, no. 1, pp. 31-52, 2000.

[40] F. Pelisson, D. Hall, O. Riff, and J.L. Crowley, "Brand Identification Using Gaussian Derivative Histograms," in *Proc. of the 3rd International Conference on Computer Vision Systems*, pp. 429-501, Graz, Austria, 2003.

[41] K. A. McCrae, D. W. Ruck, S. K. Rogers, and M. E. Oxley, "Color Image Segmentation," in *Proc. of SPIE, Applications of Artificial Neural Networks*, vol. 2243, pp. 306-315, Orlando, FL, Apr. 1994.

[42] L. Xiong, D. Li, H. Hu, and G. Jin, "Segmenting the Color Image in a Simple Background by ANN Method," in *Proc. of SPIE, Symposium on Multispectral Image Processing*, vol. 3545, pp. 470-473, Wuhan, China, Oct. 1998.

[43] S. N. Krjukov, T. O. Semenkova, V. A. Pavlova, and B. I. Arnt, "Backpropagation Neural Network for Adaptive Color Image Segmentation," in *Proc. of SPIE, Applications of Artificial Neural Networks in Image Processing II*, vol. 3030, pp. 70-74, San Jose, CA, Feb. 1997.

[44] A. Doulamis, N. Doulamis, K. Ntalianis, and S. Kollias, "An Efficient Fully Unsupervised Video Object Segmentation Scheme Using an Adaptive Neural-Network Classifier Architecture," in *IEEE Trans. on Neural Networks*, vol. 14, no. 3, pp. 616-630, May 2003.

[45] S.-J. Lee, C.-S. Ouyang, and S.-H. Du, "A Neuro-Fuzzy Approach for Segmentation of Human Objects in Image Sequences," in *IEEE Trans. on Systems, Man and Cybernetics*, vol. 33, no. 3, pp. 420-437, June 2003.

[46] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[47] C. Toklu, A. M. Tekalp, and A. T. Erdem, "Semi-Automatic Video Object Segmentation in the Presence of Occlusion," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 4, pp. 624-629, June 2000.

[48] J. Shi, and J. Malik, "Normalized Cuts and Image Segmentation," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[49] M. Allmen, and C. R. Dyer, "Computing Spatiotemporal Relations for Dynamic Perceptual Organization," in *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 58, no. 3, pp. 338-350, Nov. 1993.

[50] J. Shi, and J. Malik, "Motion Segmentation and Tracking using Normalized Cuts," in *Proc. of the IEEE International Conference on Computer Vision*, pp. 1154-1160, Bombay, India, Jan. 1998.

[51] D. DeMenthon, "Spatio-Temporal Segmentation of Video by Hierarchical Mean Shift Analysis," University of Maryland, Tech. Rep., 2002.

[52] M. Côté, P. Payeur, and G. Comeau, "Comparative Study of Adaptive Segmentation Techniques for Gesture Analysis in Unconstrained Environments", in *Proc. of the IEEE International Workshop on Imaging Systems and Techniques*, pp. 28-33, Minori, Italy, Apr. 2006.

[53] D. Comaniciu, R. Visvanathan, and P. Meer, "Kernel-Based Object Tracking," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 2003.

[54] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Video Object Segmentation Using Bayes-Based Temporal Tracking and Trajectory-Based Region Merging," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 782-795, June 2004.

[55] J. Badenas, and P. Filiberto, "Segmentation Based on Image-Tracking in Image Sequences for Traffic Monitoring," in *Proc. of the 14th International Conference on Pattern Recognition*, vol. 2, pp. 999-1001, Brisbane, Australia, Aug. 1998.

[56] J. A. Withers, and K. A. Robbins, "Tracking Cell Splits and Merges," in *Proc. of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 117-122, San Antonio, TX, Apr. 1996.

[57] S. Bériault, P. Payeur, and G. Comeau, "Flexible Multi-Camera Network Calibration for Human Gesture Monitoring", in *Proc. of the IEEE workshop on Robotic and Sensors Environments*, Ottawa, ON, Canada, Oct. 2007.

Appendix A K-Means Clustering

The K-means algorithm is a process that attempts to group data based on its attributes into k partitions. The algorithm has a wide range of applications; it can, for example, be applied to reduce the number of colours in an image. In this case, the data would be the various colour pixels whose attributes can be represented as an RGB or other colour space vector. Ultimately the algorithm attempts to minimize the total variance found within each partition. This concept is represented in equation (A.1).

$$V = \sum_{i=1}^{k} \sum_{x_j \in P_i} \left| x_j - \mu_i \right|^2 \qquad \text{(A.1)}$$

Here V represents the total variance found among all partitions. It can also be seen as a measure of how well each data point x_j fits into its own partition P_i. Each partition is described by its mean vector μ_i.

The K-Means algorithm does suffer from two major shortcomings; the value of k is not necessarily known for a data set and even if the number of partitions was known their centers still need to be determined. To resolve the latter of the two problems, an iterative refinement process known as Lloyd's algorithm can be used. This refinement works by first assuming the number of k partitions required. The partition centers are randomized and each data point is associated to the partition whose center is the closest. Once each data point has been associated to a center, each partition center is re-computed as the mean of the points associated to it. With the new means computed, the process is

repeated until a minimum in the total variance has been reached. This refinement process can be quite slow for large data sets.

In Deng *et al.*'s Peer Group Filtering [26] technique, the problem of selecting a parameter k is resolved by applying Lloyd's algorithm multiple times for different values of k. As the number of partitions increases the total variance to which Lloyd's algorithm converges towards should diminish. The number of partitions stops increasing when the total variance hits a threshold set by an operator. This threshold is highly dependent on image content. The K-means algorithm also does not fair very well in scenes where colours do not cluster circularly around the partition centers.

Appendix B Kernel Density Estimation

Kernel density estimators are a means with which data distributions can be estimated without requiring that they fit into pre-defined parametric representations. The motivation behind kernel density estimators comes from the use of histograms and their shortcomings. This appendix looks at how histograms can be used in order to estimate distributions and how kernel density estimators are in fact a generalization and improvement of histograms.

Histograms are an easy and efficient way of representing data. Their representation however depends on the size as well as the start and end points of the bins. As Figure B.1 demonstrates, by simply changing the bin size of a histogram its representation can significantly change. In the top portion of the figure the random data seems to follow a single mode distribution, however by changing the sampling of the histogram, the distribution does not seem to follow the same kind of parameterization.

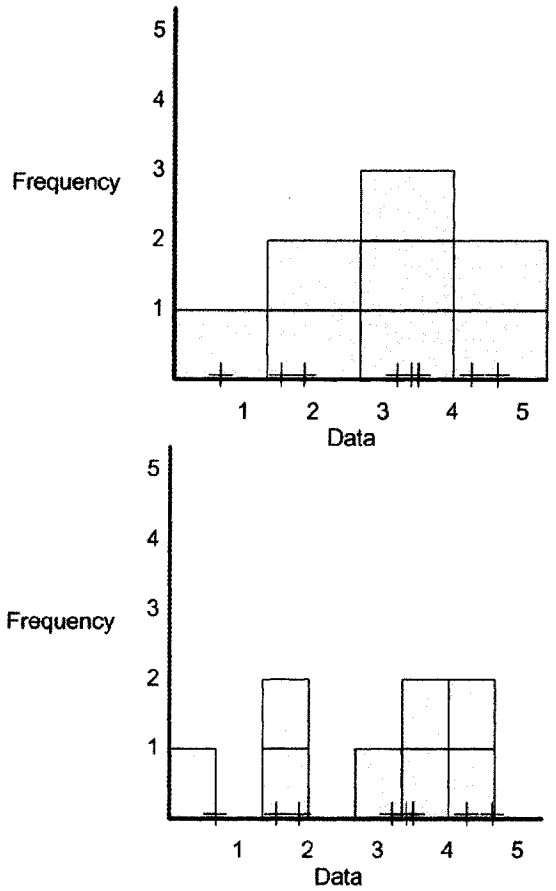


Figure B.1 - Histogram Representation of Data Distributions

In order to alleviate some of the problems with the histogram representation, kernel density estimators can be used. These estimators center a kernel function on each data point, the response at these points is added in order to have a more appropriate representation. Figure B.2 shows the result of using a block kernel estimation whose width and height are equivalent to the bin size used in the first histogram of Figure B.1. The representation remains discontinuous because the kernel function is discontinuous as well. By selecting a continuous kernel function a smooth distribution curve can be obtained. While the kernel density estimators solve the problem of histogram end points, they do not necessarily provide any insight on how large their bandwidth should be.

There are plenty of techniques available for determining the appropriate bandwidth for a kernel density estimator, for reasons of brevity they are not discussed here.

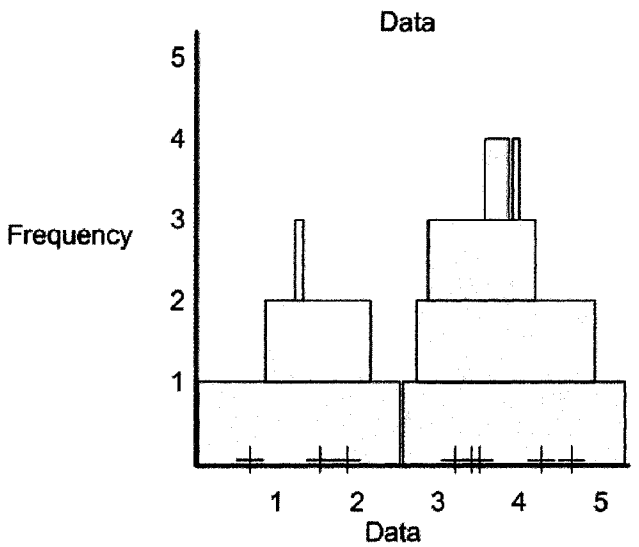


Figure B.2 - Data Representation Using a Discontinuous Kernel Estimator

From a mathematical stand point a kernel density estimator can be described using the equation (B.1).

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x(i)}{h}\right) \qquad \text{(B.1)}$$

Here the density $\hat{f}(x)$ is estimated for n points. The kernel function is represented by K and is considered to have a bandwidth of h. If the kernel function is found to integrate to 1, then the same conclusion can be applied to the density estimate. A Gaussian curve is typically the most common type of kernel function.

Appendix C Examples of Homogenous Colour-Texture Maps

The technique used in this thesis relies on colour-texture maps in order to identify regions of an image that are homogenous. These regions are identified by computing every pixel's J-value. A large value indicates a pixel of low homogeneity, while a low value indicates a pixel of high homogeneity with its local neighbourhood. This section will demonstrate how these values are computed for different scales as well as give simple examples.

The formulas used in the computation of a J-value are covered in section 4.1 with equations (4.8) to (4.12). Figure C.1 shows an example of several class distributions and their resulting J-value computation. Each example is formed from a set of 3 classes with different types of spatial distribution. In example 1 each class is clearly divided; in example 2 all the classes are uniformly distributed; in example 3 only 2 of the 3 classes are distributed. A distribution of multiple classes over an area is typical of the colour behaviour exhibited by textures. From the results shown below, it is clear that class maps having several yet uniformly distributed labels still exhibit a low J-value. This property is what makes the algorithm able to identify both colour and texture homogenous regions.

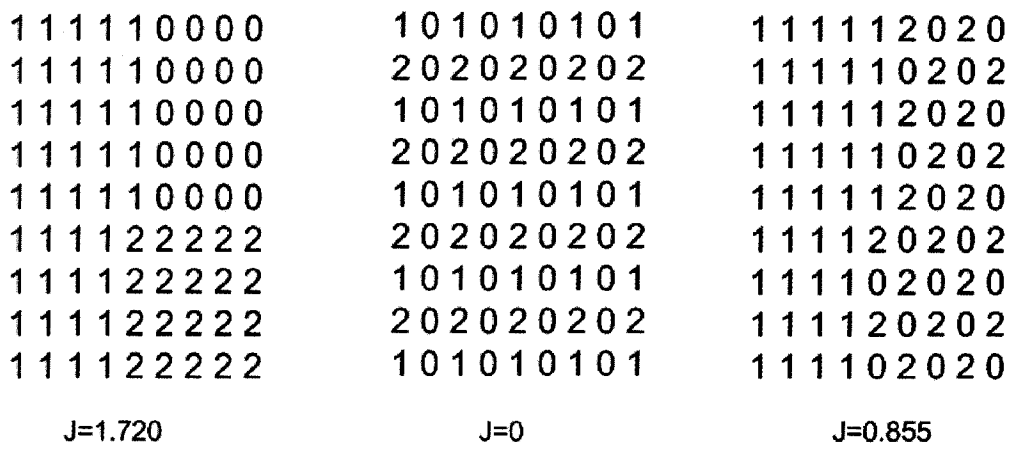


Figure C.1 - Example of *J*-Value Computations for Various Class Maps

In order to have a mapping of the J-values over an entire image, each pixel on which the computation is to occur needs to define a neighbourhood area. This area is represented by a circular kernel mask. The size of the mask will determine the scale at which colour-texture edges will be found. The base kernel is a 9x9 mask, as seen in Figure C.2; it is up-sampled in order to take into consideration larger edges. As the kernel becomes larger it also takes into consideration a larger neighbourhood of pixels. The homogeneity of this larger neighbourhood will be reflected in the final *J*-value. Regions are determined at the largest kernel scale and then refined by applying a smaller kernel scale within these regions. The smaller neighbourhood of pixels being considered within a region will dictate if the region will be split into smaller ones. Figure C.3 shows an example of the J-value computation for a simple image having both subtle and more gradient edges. At a smaller scale the sharper edges are clearly distinguished, while at a larger scale, the colour gradient is better observed. The shape of the kernel, its up-sampling behaviour, and the *J*-value thresholds applied in the region determination process are given by the original authors Deng *et al.* [3].

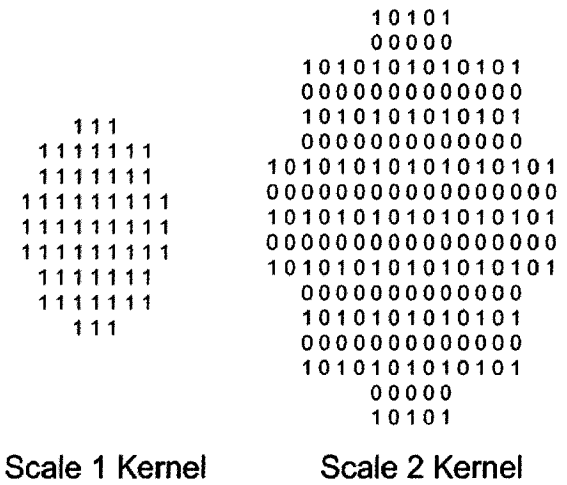


Figure C.2 - *J*-Value Kernel Masks

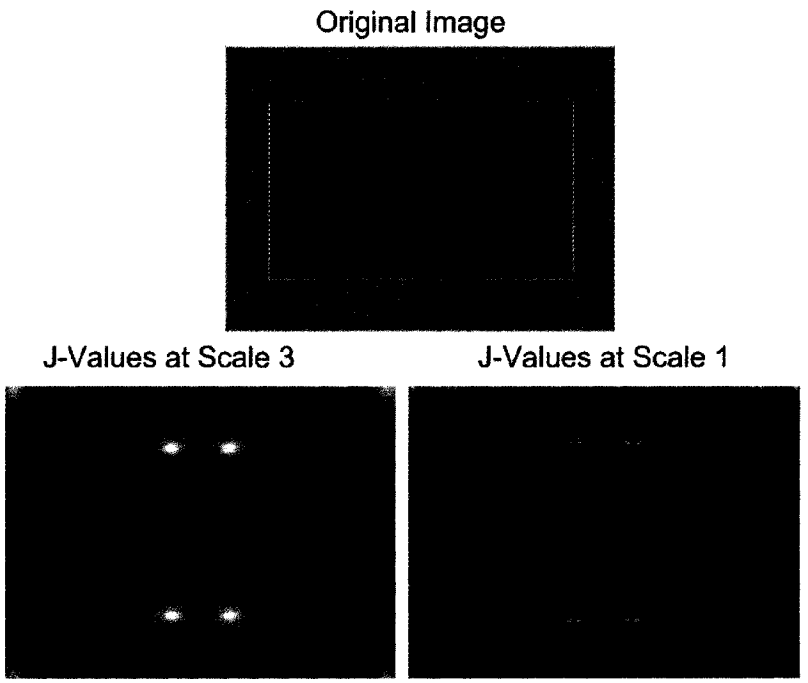


Figure C.3 - *J*-Value Representation of a Simplified Image