University of Leeds
School of Computing

Visual Signature
For Large Scale Tracking

Dima J. Al Damen
Supervised by Dr. Andy Bulpitt
MSc in Distributed Multimedia Systems

The candidate confirms that the work submitted is her own and the appropriate credit has been given where reference has been made to the work of others.

(Signature of student) ________________________

Abstract

Visual Surveillance is no longer confined to tracking people. Identification of people being tracked across several cameras forms the basis of future surveillance systems. Such systems could be installed in public areas like city centres, airports, underground stations, etc.

The optimal solution is to find a set of features that can uniquely identify each tracked pedestrian. This collection of features, called the visual signature, can then be used to identify each person during one session of the system. HP labs in Bristol are currently trying to develop a large-scale tracker using the idea of the visual signature. Their tracker is based on the existence of such a visual signature.

The main research concern is what information needs to be extracted from the person to formulate the basis for such a signature. Colour information stands as a high-probability candidate. Colour histograms and Gaussian mixture models can be used to represent colour spaces. Techniques are implemented and compared for the purpose of finding the visual signature.

Since colour histograms lack any geometric relations, the project proposed a new scheme to increase the performance of the histograms. The three-regions scheme, which divides the body into three regions horizontally, tries to overcome this disadvantage. The experiments demonstrate that this histogram-based signature is capable of both identifying the same person and differentiating amongst people.

The project includes a thorough background review in the field. The findings are based on a series of experiments and results. In addition, future directions are explained for deciding on the best representation of the visual signature. The project's deliverables also include a Java library to be used in similar researches.

Acknowledgements

I wish to express my sincere gratitude to my supervisor Dr. Andy Bulpitt for his constant support and encouragement. As this project was a new field to me, his support and follow up were essential and highly appreciated.

I would also like to extend my appreciation to the computer vision group members headed by Professor David Hogg, in particular Dr. Derek Magee, Mr. Christopher Needham and Professor Roger Boyle. Appreciation is also extended to my assessor Dr. Nick Efford for his valuable guidance.

This project could not have been carried out without the opportunity granted by HP labs in Bristol. Mr. Colin Low and his colleagues gave an ideal chance to challenge my knowledge against a real-life application system with lots of implications on our daily lives.

Finally, I owe a great deal of thanks for my friends and colleagues, including those who participated in my experiments' sample.

The last to mention, but no doubt the most important, is my family. Rima, Rawan, Ahmad and Jamal, for their support, encouragement, love and understanding.

Table of Contents

1. Introduction

The importance of tracking multiple people is increasing nowadays with the wide availability of networked cameras. Computer Vision researchers aim towards an automatic and real-time analysis of human behaviour using such cameras. Other target applications include visual surveillance in public places like car parks and shopping malls. (Orwell 1999, p. 14)

Such systems were made possible by many factors. The powerful computers that can process colour images (Cheng 2001, p.2259) and digital cameras are some of them. Yet, a plug-and-play software package for visual surveillance is still far from being available in the near future. Several technical and methodological difficulties hinder its introduction to the market. In their paper "Intensity independent color models and visual tracking", Korhonen et. al. state, "visual tracking is a wide and complex problem. The application possibilities are diverse, and so are the circumstances". (2000, p.600)

It is agreed upon that a "satisfactory solution for tracking multiple people in generic video is not yet available". (Lu & Tan 2001, p.137) Lots of research has been done recently in the area, yet all was directed towards specific conditions and environmental settings. Robust systems that fit in external circumstances and changes beyond control are a necessity to achieve human visual tracking. (Korhonen 2000, p.600)

1.1. The need for the visual signature – Problem Statement

One of the main requirements towards a plug-and-play human visual tracker is finding the identity of the human being tracked. HP laboratories produced a proposal towards building such a system. Yet, this proposal is based on the assumption that a visual signature of the human can be obtained.

In his paper, "Visual Sensing for Large-Scale Tracking", Colin Low defines the visual signature as "a finite bitstring of some kind". (2002, p.2) The paper elaborates by stating, "the tracking problem becomes trivial if each camera in a system generates a unique signature for each unique object presented to it, and each unique object results in the same unique signature being generated by each camera". (p.2) Yet, producing such a signature is far from trivial. It is understood that this signature cannot be regarded as unique, and an acceptable level of confusion can be dealt with. (p.3)

The paper concludes by stating that producing the signature is "a formidable task… various researchers claim to have solved significant parts of the problem". (p.22) This research tries to collect, understand and analyse the available research techniques that might lead to the production of such a signature.

In creating the signature, it is clear that the signature uniqueness is maintained for the same person whenever he or she is within the system tracking area. (Krumm 2000, p.3) The signature need not be maintained for the same person across multiple sessions of the system. Other systems, in which the signature is expected to be unique for the person across all sessions, have different requirements.

The idea of the visual signature was introduced in other researches and software companies. The FaceIt ARGUS product developed by Identix Inc. generates a multi-session unique signature that identifies people based on the shape-model of their faces. Such signatures are viewpoint dependent and are not suitable for large-scale whole body tracking. These security-oriented systems find their application in controlled-environments with a limited number of people, like a company building or complex.

Microsoft Research Vision Technology Group expressed their need for such a signature. In a research entitled "Multi-camera multi-person tracking for EasyLiving", the group states that knowing the identity of people in a room is a vital necessity for their intelligent software products. (Krumm 2000, p.1) The group tries to utilize the identity and location of people in producing an intelligent living room that reacts towards the activities of people in the room. To identify the people in the room, a collection of techniques was utilized. One of these is an active badge; it is an electronic wireless device that can locate people in the tracked area. Special receivers can catch signals from these badges and identify uniquely the person wearing the badge. (p.2)

It should be clearly stated that producing the visual signature is an object recognition technique. Thus, finding the object and segmenting it from the background scene is beyond the scope of the project. It is assumed that other techniques are to be used to segment the foreground from the background, and divide the foreground into areas that correspond to the number of people tracked in the scene. Several techniques are provided in the literature to support this technique. They might still need further enhancement when it comes into occlusion and condensed areas, yet this is another area of research.

No further assumptions will be taken in the search for techniques to produce the required visual signature. In considering a particular solution and implementing it, assumptions will be stated for that particular implementation. Yet, this review is general and not confined to any particular conditions or circumstances.

1.2. Minimum Requirements

It was very difficult to predict the findings of this project. Experiments might only show that the implemented technique is not of any usage in creating the visual signature. After a committee meeting of the School of Computing, the minimum requirements of this project were amended and confined to the following:

1. Clear understanding of the current state-of-art in real-time tracking
2. Understanding of the current systems available in the School of Computing to help in producing such a signature.

The original minimum requirements could be found in appendix B. The first minimum requirement was met through the background review found in chapter two. A case study of the two trackers is available in chapter three and satisfied the second minimum requirement. The dropped requirement was related to proposing a solution to the visual signature dilemma. Proposing a solution could be a tricky requirement. Yet, it was clear from the beginning that the project is not only a literature review. Some techniques are going to be implemented and tested.

1.3. Fields of Research

The project brings together the fields of computer vision, artificial intelligence and image analysis. The literature reviewed belonged to the following areas: computer vision, visual surveillance, image retrieval databases, object recognition, colour information and people tracking systems.

It should be mentioned that the majority of the useful researches were found in the image retrieval database and object recognition literature. It is obvious that the problem of the project is close to the problem being currently studied in these fields. The search for occurrences of one object in a collection of images is directly related to the need for the visual signature.

1.4. Major Achievements

The project not only experimented the widely used techniques in the literature but also new suggestions were proposed. The project had a good collection of background readings. The two existing trackers were studied and analysed. The blob tracker was modified for the usage of the project. Four cycles of experimentation and evaluation were carried out and several conclusions were drawn from these. The project concluded that the three regions scheme RGB colour histogram is the best technique to be used for visual signatures in controlled environments. The three regions scheme proposed in this project was explained in section 5.2.1.

The project findings and suggestions were discussed in two focus groups held by the computer vision group in the university and in HP labs in Bristol. Both groups said the results are worth pursuing, and their suggestions could be found in this report.

Not only the project had some good results that are worth pursuing, but it also resulted in a Java library to be used for subsequent research. These Java classes are added now to the library of the computer vision group, which was confined to C and C++ classes. This will enable future researchers to utilize the power of Java in their researches.

Moreover, the results of the first and second experiments were presented in a paper conference to the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance in France. The paper could be found in appendix D.

1.5. Report Structure

The report is structured in seven different sections. These sections are followed by a set of appendices. The project report aims at explaining how the project was carried out without going into excessive technical detail. The report's expected readers are researchers in the topic who wish to pursue similar or related experiments.

The chapter following the introduction discusses the literature reviewed. The chapter, entitled "in search for the visual signature", is structured according to the topics discussed. The literature chapter forms the basis for any later design decisions. This is because the project's goal is to implement promising techniques derived from the literature reading.

Chapter three targets the second minimum requirement of this project. The chapter explains the two trackers developed by the computer vision group and justifies the choice of the tracker to be used in the experiments.

The fourth chapter restates the requirements for the developed library, the design decisions and the methodologies followed in both the research and the implementation. Because the research was carried out in cycles, an iterative research methodology was followed. Action Research enables utilizing the results of pervious experimentation cycles in the following ones. OOA&D was used to enable iterative development of the library and experimentation code.

Experiments were discussed and evaluated in the fifth chapter. Analysis, design and implementation phases of four different experiments could be found there. Each experiment has its own evaluation and each experiment is expected to affect the design decisions of the next experiment. Due to the fact that this project is research-based, experiment results formulate the basics for the project's findings.

The Project has more than its findings only. Project management, which is described in the sixth chapter, involves the way the project duration was utilized to meet the objectives. The key challenges of the project were stated, and the techniques to face these challenges are explained.

In researches, future directions have a considerable importance. Hence, chapter seven proposes further experiments and other techniques to be implemented. It also contains a summary about the project's progress. References and bibliography could be found at the end of the chapter.

The report has eight appendices. The first appendix is a reflective piece on the flaws and successes of the project. The second appendix is a copy of the objectives form. The third contains the notes of both the supervisor and the assessor given to the student along with the interim report. The fourth appendix is the paper submitted to the IEEE workshop. A full copy of the paper, which summarized the project's findings for two out of the four experiments, could be found there.

As the project resulted in a Java library that can be used for later researches, the library design documents and final documentation could be found in Appendix E. Examples of the tables representing detailed results of the experiments are provided in appendix F. The last two appendices index the figures and tables included in this report.

2. In search for the Visual Signature

The problem of identifying the visual signature was re-stated by Orwell et. al. in their paper, “Optimal color quantization for real-time object recognition” as follows: “the goal of the process is to determine which (if any) previously observed signal refer to the same physical object as the currently observed signal”. (2001, p. 404) This requirement was utilized in image retrieval systems. Retrieving images that contain a particular object/person from a large database of images falls within the thriving class of applications for such a visual signature. Yet, walking pedestrians make the problem more complex as they constitute non-rigid objects, and do not have a fixed external shape and colour combination. (Lu & Tan 2001, p.137)

The aim of this project is to study the available techniques in the literature that might help in producing the required visual signature. The project would then elaborate towards proposing and testing solutions that the literature supports. Although the search for the visual signature has just started and previous researches were not aiming at finding the visual signature, one should utilize the previous researches in similar areas.

2.1. Colour Information as a basis of the visual signature

Colin Low’s paper proposed colour indexing as the basis on which identification can be done. This section reviews the general trends in the literature towards identifying an object visually, shedding some light on the strengths and weaknesses of colour-based techniques. The techniques can be divided into four categories: colour-based, 2D shape-based, 3D shape-based and edge-based techniques.

Colour-based techniques try to identify an object based on the colours of the area that the object occupies, i.e. the pixel colours. Colour-based is a new trend in object recognition. 2D shape-based techniques have been in use comparatively longer. They have experienced some enhancements with the introduction of snakes, PDMs (Point Distribution Models) and AAMs (Active Appearance Models). (Cootes 1998) 3D shape-based techniques are more complex techniques that try to re-generate the 3D shape of the object. They require complex facilities such as cameras with overlapping fields of view or stereo cameras. Edge-based techniques try to study the edges of the object based on the difference of colour between the inner of the object and its surroundings. Recent research carried out at the University of Southampton discussed the usage of gait information. Gait is believed to be the “most universal and complex of all human activities”. (Yoo 2002, p.35) Yet, at the same time, gait cannot be used as a per-frame signature. This is because several frames had to be analysed before gait information can be understood.

Before discussing and comparing these techniques, it is worth mentioning the problems that face any visual signature technique. A list of these problems is:

1. Scale: The signature provided should be the same regardless of whether the object is close to the camera or resides further in the scene.
2. Rotation: The signature should remain constant as the object is rotated around the axis perpendicular to the image’s plane.
3. Orientation: The signature should be reasonably immune to viewpoint changes.
4. Occlusion: The object should be still identified if partially occluded by other static or dynamic objects in the scene. The signature generation algorithm should be also capable of dealing with full occlusion so it does not experiment algorithm failure.

5. Shadows: The signature should not be affected by changes in the object's shadow.
6. Changes in Lighting direction and/or intensity: Regardless of the reason for lighting variations, the signature should still identify the same object under different lighting conditions.

Along with this list lies the problem of complexity. As the visual tracking systems are expected to run in real-time, they must be quick enough to ensure real-time performance. In addition, complex algorithms require more powerful and thus more expensive computer support, which limits the availability of the system.

Swain and Ballard noticed that colour has been neglected for a long time as a "recognition cue" because it is not an intrinsic characteristic of the object in comparison to shape-models or edges. (1990, p.390) It is widely discussed in the literature that colour-based techniques are much more immune to variations caused by the above problems. It is logical that the overall colour of the object does not change with scaling or rotation while the geometry of the object does. (Low 2002; Elgammal 2001, p.563; Raja 1998a, p.228) The literature also claims that colour-based techniques' performance supersedes their competitors in other cases. Colour is considered to remain relatively constant as the viewpoint changes. (Grove 1998, p.1442; Funt & Ginlayson 1995, p.522) Moreover, occlusion is a major problem in tracking condensed areas, as it can never be avoided. Compared with 2D shape descriptors and 3D models, colour is less sensitive to partial occlusion, (Grove 1998, p.1442; Elgammal 2001, p.563; Raja 1998a, p.228; Lu & Tan 2001, p.137; Funt & Ginlayson 1995, p. 522) though shape-descriptors could be designed to handle the reduction in the amount of object data retrieved. (Rothwell 1995, p.8) Colour-based techniques are also much easier and faster to implement for real-time systems. Grove et. al. explained that the "low computational cost of the algorithm… makes colour a desirable feature to exploit when appropriate". (1998, p.1442; Funt & Ginlayson 1995, p.523)

The major difficulty that faces colour-based techniques is that colour changes dramatically with lighting conditions. (Lee 2001, p. 1659) In addition Fieguth and Terzopoulos discuss that it is more vulnerable to noise from camera and background. (1997) Noise reduction is part of the segmentation process which is beyond the project's scope.

Despite these deficiencies, researchers began to use colour-based techniques quite widely since their official introduction by Swain and Ballard in 1990. Several techniques are studied towards decreasing these problems and utilizing colour-based techniques to the most. Bressan et. al. summarize that "when lighting conditions do not change severely…, the color distribution of an object is a simple kind of sensor data that have demonstrated to be an efficient signature of object-recognition". (2003, p.691-2)

It should also be mentioned that this trend towards colour adoption is not general. Several researchers criticized the use of colour. For example, Mason and Duric's experiments show that edge-based techniques are more accurate in object identification. Li et. al.'s results proved that 3D-models perform much better for multiple viewpoints. (2001, p.43)

The following table shows the performance of colour-based techniques under the problems mentioned above:

Variation	Colour
Scale	Unaffected
Rotation	Unaffected
Orientation	Performs quite well with slight viewpoint changes
Occlusion	Endures partial occlusion
Shadows	Endures deficiencies in background segmentation
Lighting changes	The worst technique as the colour changes dramatically under illumination changes
Complexity	The simplest & fastest technique available

Table 2.1 Independencies for Colour Information

Based on the above discussion, colour seems to be a good candidate towards object identification. Other auxiliary techniques can always be introduced to help in object identification. Some of these were introduced by Krumm (2000): stereo cameras can provide depth information which helps in segmenting and identification of people (p.4), active badges discussed in the previous chapter (p.2), omni directional cameras provide a wider coverage (p.3), cameras with overlapping fields of views can decrease occlusion (p.1). These auxiliary techniques can always be introduced to increase the signature uniqueness and help decrease confusion. Yet, they are much more expensive to use, and sometimes they are not feasible in public area locations.

There is no doubt that one cannot depend on colour information alone for a signature with very high matching results. Other features should be combined together with colour into a multi-feature vector. Colour is however expected to play a major role in such a feature vector.

2.2. Colour and Colour Spaces

In referring to the word 'colour', one should clearly differentiate between human understanding of colours and computerized modelling. Currently, colour information is stored in an artificial colour space. Different collections of colour spaces have been tested in the computer vision context. It is still debatable if any of these colour spaces is better than the others or closer to the human perception.

RGB Colour space:
RGB was used in most of the researches reviewed. (Krumm 2000; Elgammal 2001; Mason & Duric 2001; Lee 2001; Fieguth & Terzopoulos 1997; Agbinya & Rees 1999; Swain & Ballard 1990; Funt & Ginlayson 1995) Yet none of these researches provide any clear explanation of the use of RGB. Despite the wide coupling of the three components of the colour space (red, green and blue), it is still widely used, as most cameras provide a RGB signal. (Cheng 2001, p. 2260; McKenna 1999, p. 226)

To decrease the correlation between RGB components, while utilizing its availability, some normalized colour spaces were derived from RGB. For example normalized RG space was used by Lee (2001) and Finlayson (2002). Some researches state that after experimentation, intensity-independent colour models, such as HSI and normalized RG, produce better results. (Korhonen 2000, p.603)

HSI Colour Space:
HSI competes with RGB widely as it is designed to break the correlation between chromaticity and intensity. It was used in the reviewed literature, (McKenna 1999; Grove 1998; Lu & Tan 2001) and a thorough explanation of its advantages was provided.

HSI colour space divides the colour into three components. Hue represents the basic colour and dominant wavelength in the spectral. Saturation, on the other hand, represents the purity, sometimes referred to as the 'vividness' of the colour. Intensity represents the brightness of the colour. (McKenna 1999, p.226; Cheng 2001, p. 2262) Yet, the notion of intensity provided here is not the human-perception of intensity; it is just an average of RGB values. (McKenna 1999, p. 226)

Hue is believed to be very useful in separating objects with different colours. It is also "invariant to some types of highlights, shading and shadows". (Cheng 2001, p. 2275) However, HSI suffers a serious disadvantage; its non-removable singularity that creates discontinuities in the representation of the colour. Thus, "a slight change of input RGB values can cause a large jump in the transformed values". (Cheng 2001, p.2263)

Despite the frequent utilization of the HS 2D colour space in the literature, Lee et. al. show how the HS space changes its place as the brightness changes. (2001, p.1660) Their research concludes that the relationship between the HS space and the intensity is a second-order B-spline curve function. (p. 1664)

CIE Colour Space:
The utilization of the LUV colour space provided by the cameras was merely dependent on experimentation results. The researches that adopted this scheme claim that it is far better than the above two schemes. (Korhonen 2000; Orwell 2001; Chen & Lu 2002) CIE colour space was based on the LUV colour space but was oriented towards human perception of colour. Because the range of CIE colour space values exceeds 300, normalized CIE was developed to restrict the ranges to 0 through 1.

There is no agreement as to which colour spaces are more suitable for computer vision purposes, neither is there any standard understanding of the human colour perception. Thus, it is still left to the researchers to pick the colour space that satisfies their requirements. This choice is still purely based on their own experimentation results. Quantitative results are used in favouring one colour space over the others. For full table of advantages and disadvantages of different colour spaces refer to Cheng. (2001, p.2275) Using the physics-based models of the light is far from simple and thus is not feasible. (Raja 1998a, p.229)

2.3. Colour Histograms and Histogram Intersection

After selecting the colour space, one should proceed into defining the proper representation of the colour information. Swain and Ballard proposed the concept of the 'colour histogram', which has been used intensively since then. (Chang & Krumm 1999; Chen & Lu 2002; Cheng 2001; Elgammal 2001; Funt & Ginlayson 1995; Krumm 2000; Raja 1998b)

Before building the colour histogram, the colour space is usually divided through a step entitled 'quantization'. Krumm et. al. claim that quantization "reduces the effect of spatially varying illumination in the scheme". (2000, p. 6) In their research, they divided the RGB space into 4x4x4 colour cubes, which results in 64 bins. Mason & Duric also quantized the 24-bit colour into 12-bits. (2001)

Orwell et. al. criticised this quantization technique because it linearly sub samples the data and does not maximize the differences between objects. Thus, the "extreme bin values will be rarely used, and the commonly used values may be so common that they are unable to discriminate between different entities" (Orwell 2001, p.408). Histogram equalization and INFOMATCH algorithm are proposed to solve this problem. Such concerns are more applicable to object recognition. The differentiation between humans is usually through their clothes that constitute the majority of the object's area in the scene.

The choice of the number of bins affects the resulting density. Choosing a fine quantisation results in a "noisy" density and a large number of empty bins but reduces the effect of illumination. On the other hand, a very small number of bins will "smooth away" the density structure (Raja 1998a, p.229). Thus, there is a trade-off in choosing the number. Yet, most researchers agree that as the number of bins lies within a certain range, the result is somehow constant. The experiments of Gevers and Smeulders showed that the number of bins was of little influence on the recognition accuracy if the bin number lies between 32 and 256 bins for all colour spaces. (1999)

The basic idea of the colour histogram proposed by Swain and Ballard is to count the number of times each colour bin occurs in the targeted image. (1990, p.391) The colour histogram is very dominant in applications as it is invariant to translation and rotation about an axis perpendicular to the image plane. It is only slightly affected by camera motion, scaling and occlusion. (Swain & Ballard 1990, p.391; Lu & Tan 2001, p.137)

Several enhancements were proposed to improve colour histograms, one of which is colour co-occurrence histograms (CH). (Chang & Krumm 1999) "CH adds geometric information to the normal color histogram, which abstracts away all geometry". (p. 498) Despite the introduction of geometrical notation, CH still adapts to different viewpoints and scales by adjusting the distances used in CH, yet the algorithm to perform this is not clearly stated in Chang & Krumm's proposal. One example, to show the effects of lack of geometry, was proposed in Orwell et. al. Colour histograms cannot distinguish between "the Belgian and German flag: although they have a different spatial configuration, they contain the same colors in the same proportions". (2001, p.403)

Another suggestions that consider geometric properties involve dividing the target object into regions and considering the histogram of each region in addition to the spatial information between these regions. (Bressan 2003, 692) A similar suggestion was made earlier by Matas et. al., Colour Adjacency Graph (CAG) represents both the colour of the regions and their adjacency. (1995)

After obtaining the histogram, techniques should be found to compare and contrast two histograms. Histogram Intersection is the standard technique commonly used along with colour histograms. (Krumm 2000; Raja 1998a) Histogram Intersection was also proposed by Swain and Ballard to solve some problems in object recognition: distractions in the background, different image and histogram resolution, changes in viewpoint and occlusion. (1990, p. 391)

Given two histograms H and M (Model), each containing n buckets, then the histogram intersection value (X) is defined by:

$$X = \frac{\sum_{j=1}^{n} \min(H_j, M_j)}{\sum_{j=1}^{n} M_j}, \quad X \in [0,1]$$

Another major advantage of colour histograms and histogram intersection algorithms is their simplicity. Orwell et. al. proposed the implementation of both histogram creation and comparison using specialized hardware to ensure real-time performance. (2001, p.401) In presenting the disadvantage of this algorithm, Swain and Ballard cleared that the technique is robust to scale changes but not scale invariant. (1990, p.391) Dividing the bin counts by the total number of pixels ensures scale independency.

To decrease the illumination effect, Fieguth and Terzopoulos (1997) proposed a new method. The new technique is also believed to be insensitive to camera noise and scaling. It can be explained briefly as follows (p. 22):

If the object required to be tested is represented as (r_i, g_i, b_i) and the model object by (r_i^*, g_i^*, b_i^*):

$$t_r = \frac{r_i}{r_i^*};\ t_g = \frac{r_g}{r_g^*};\ t_b = \frac{r_b}{r_b^*}$$

$$X = \frac{\max(t_r, t_g, t_b)}{\min(t_r, t_g, t_b)}$$

X=1 implies a perfect fit and X increases as the fit becomes poorer.

Despite all the advantages of the histogram and histogram intersection techniques, Swain and Ballard clearly stated that it is "sensitive to changing light conditions" (1990, p. 392). To solve this, several suggestions were proposed. One of which is to store a new histogram for the person in every different part of the tracking area so it covers all the lighting conditions. (Krumm 2000, p.6; Bressan 2003, p.694) This suggestion only works for previously known people for whom the system has been trained.

Others proposed some sort of adaptation to the colour histogram. In their paper, Funt and Ginlayson proposed indexing the derivative (Laplacian or first directional derivative) rather than indexing the colour itself. (1995, p.523; Han & Ma 2002) This makes a histogram of the relative ratio of the object's colours rather than the actual colours themselves. The idea is based on the assumption that "the ratio of sensor responses across a color boundary remains unaffected by changes in the incident illumination". (Funt & Ginlayson 1995, p.523) Gevers and Smeulders criticized the colour ratio idea. This is because it assumes that points have the same surface normal. (1999, p.453) This assumption is affected by the change in the object's geometry.

2.4. GMMs

Histograms are tempting to use for data representation due to their simplicity. Yet, they are only effective when the number of bins is manageable and where sufficient data is available. (Raja 1998a, p. 229) Semi-parametric approaches like Gaussian Mixture Models could be more effective in such circumstances. GMMs are more suitable for higher-dimensional distributions. (Orwell 2001, p.403; Korhonen 2000, p.602) Although GMMs are a very appealing technique to decrease the complexity of a large data set, they fail to "discover true structure in cases where the partitions are clearly non-Gaussian". (Roberts 1998, p.1141)

Semi-parametric approaches decrease the number of information recorded about the underlying data points. Semi-parametric models only grow with the increased complexity of the data representation rather than the number of data points. (Bishop 1995, p.60) The GMM is then simply represented by the collection of weights, means and covariances of the Gaussian distributions. (Orwell 1999, p. 15) The optimisation of the GMM is most commonly achieved through the Expectation Maximisation (EM) algorithm.

Each GMM is represented by a collection of Gaussians. The GMM records the probability that each data point (i.e. pixel) in the space under investigation belongs to one of the Gaussians in the mixture model.

$$p(x|j) = \frac{v_j \cdot G(x, n_j, \Sigma_j)}{\sum_{i=1}^{n} v_i \cdot G(x, \mu_i, \Sigma_i)}$$

$$\text{where } G(x, \mu_j, \Sigma_j) = \frac{e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)}}{(2\pi)^{\frac{n}{2}} \sqrt{|\Sigma_j|}}$$

$p(x|j)$ represents the weighted probability that the data point x belongs to the Gaussian j. v_j represents the weight of that Gaussian within the mixture model. The weights of all Gaussians in the GMM should sum up to 1 always - $\sum_{j=1}^{n} v_j = 1$. n_j represents the mean vector for the Gaussian. The vector has the same dimension as the represented colour space. Σ_j represents the covariance matrix and is of size [dimension][dimension].

In finding the value of the Gaussian function $G(x, \mu_j, \Sigma_j)$, the distance of the data point to the Gaussian mean is considered. The 'Mahalanobis metric' is used in calculating the distance using the equation (Duda 1997):

$$\sqrt{(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

The EM algorithm tries to find the best representation of the data points using a Gaussian mixture model. The EM algorithm is a "two stage iterative scheme which computes the local maximum likelihood fit of an arbitrary number of Gaussians to a data set". (Magee 2000, p.91) The algorithm iterates to change the Gaussians' parameters to maximize their fit to the underlying data. Hence, EM algorithm tries to update the values of the means, the weights and the covariance matrices of the Gaussians in the GMM. The following equations represent the updating equations used by the algorithm. In the equations, m represents the number of data points. The EM algorithm does not specify the initialisation values for the means, weights and covariance matrices. This is left to the application-specific implementation.

$$v_j = \frac{\sum_{j=1}^{m} p(x|j)}{m}$$

$$\mu_j = \frac{\sum_{j=1}^{m} (x \cdot p(x|j))}{\sum_{j=1}^{m} p(x|j)}$$

$$\Sigma_j = \frac{\sum_{j=1}^{m} p(x|j)(x-\mu_j)(x-\mu_j)^T}{\sum_{j=1}^{m} p(x|j)}$$

The use of GMMs for colour space representation became common. (McKenna 1999; Grove 1998; Elgammal 2001; Raja 1998b) Elgammal recommends the usage of GMMs for regions with mixture of colours, "for example, people's clothing". (2001, p.563)

Despite the wide usage of GMMs, there is no standard way to compare two GMMs and derive the probability that they refer to the same object. Orwell states that “even when the mixtures are equally populated, it is not clear how to match up the two sets of Gaussians and then determine what variability is acceptable”. (1999, p.15) The other disadvantage of GMMs mentioned by Orwell is that the distributions are unstable if the data is clustered. The EM algorithm ensures the maximum likelihood but there is no guarantee for the individual Gaussians to converge in the same way under small perturbations to the total distribution. (p. 15)

Another technique is proposed to count for various lighting changes. It is called Adaptive Gaussian Mixture Model (AGMM). (McKenna 1999; Raja 1998b; Korhonen 2000; Lee 2001) Research clearly states that the number of components of the GMM is not affected by the lighting conditions. It is only the parameters of the model that require adaptation. (McKenna 1999, p.227)

The initialisation technique for AGMM was discussed widely in the literature and is still debatable. McKenna explains how any colour model can be used and will be adapted over time. (1999, p. 226) Another system is built interactively so that the user can pick the object to be tracked for initialisation purposes. (Agbinya & Rees 1999) This system requires that the objects be “cooperative, i.e. hold still, while the box contents are captured in the first frame”. (p. 297) In the case of our project, it is assumed that the objects added to the tracking area will be identified by the object segmentation process and thus, the initial colour model could be used for initialising the AGMM.

An obvious problem that faces any adaptive algorithm is the case of occlusion. The model might adapt in a wrong way if the object is occluded partially or fully. McKenna et. al. extend their proposal by a calculated measure that halts the adaptation algorithm if a sudden and large change of the model is encountered. (1999, p.227)

If O is the object being modelled, N is the number of pixels occupied by the object then L is:

$$L = \frac{\sum \log p(x|O)}{N}$$ for every x, a pixel data in O

L experiences a sudden drop in its value when occlusion occurs. A threshold value is required to stop adaptation as occlusion occurs.

In such a case, the original model is used until the object is visible again and L exceeds the threshold. (Agbinya & Rees 1999, p.299)

2.5. Multi-Camera Tracking

All previously mentioned techniques and algorithms were designed for a single-camera tracking system. The complexity added by introducing multi-camera networks is a thriving area of current research. Many researches claim that multi-cameras do not introduce any extra complexity if the adaptive mixture model is introduced. Unless special care is considered in building up the camera network topology, this is far from being feasible.

Only two reviewed researchers discussed the multi-camera condition. Agbinya and Rees stated the need for a “multi-camera handshake protocol”. (1999, p.302) The protocol should consider the viewpoints and lighting changes between each of the two cameras and how this might affect the colour model obtained. In addition, it utilizes the known camera network, i.e. as the object leaves one camera, it is expected to appear in another one. The camera the object is leaving will hand the object's information to the probable next camera(s). The protocol is very ideal but was not explained in more detail in the paper.

The other very challenging and clearly explained technique was proposed by Orwell et. al. in their paper "Optimal color quantization for real-time object recognition". The paper explains in detail the process of finding a zero-order or first-order transformation to map the UV colour space from one camera to the other. (2001, p.407) The proposed technique can even be programmed for automatic calculation of this transformation. However, the experimentation made on this technique was confined to the same viewpoint, i.e. the orientation with which the tracked object leaves the first camera is the same as that of the object entering the second camera. A different orientation was not tested.

Another issue that might be raised with multi-camera systems is the difference in manufacturing companies, which results in different colour characteristics. This might need a new modelling process for each camera specification. (Lee 2001, p. 1663) Calibration of the different camera settings is another option.

2.6. Limitations

Only one technique can never be sufficient to produce reasonably acceptable object recognition and identification. (Grove 1998, p.1442) One example is Microsoft's system for EasyLiving that utilizes all the possible techniques: active badges, cameras with multiple views, colour histograms, 2D shape-based and stereo cameras to produce a reliable tracker. Thus, any technique alone can never claim to be sufficient. The search nowadays focuses on the extent to which each individual technique can be relied on.

In addition, the reliability of any visual signature depends on the clarity and reliability of the pixels assigned to the object. Object segmentation techniques are currently far from producing exact object areas that are forwarded to the visual signature production unit.

It is still also unclear whether the visual signature can be merely based on the per-frame features of the object. The signature might require some feedback from the tracker to gauge the possibility of the object being in this position, based on previous positions and its velocity. Thus, the information provided by the tracker can help in reducing the number of models compared and relieving confusion.

Another question that remains open is the identity of the visual signature. One hopes that the result of a technique can produce a number of parameters that are combined to produce a visual signature. Yet, it might result that the visual signature is only a pointer to an adaptive colour histogram or AGMM.

3. Case Study: Trackers

3.1. The need for a tracker

In the project's specifications, it was clearly stated that the project is not concerned with the segmentation of the object from the background, because this problem is simply another project in its own. It was then decided to use an already available tracker for this purpose.

The systems available in the school of computing were analysed and studied. Two trackers had been developed in the computer vision group. The pedestrian tracker was developed by Dr. Adam Baumberg as his PhD thesis in 1995. The second tracker can be used for tracking general objects. Dr. Derek Magee designed it initially for car tracking.

Both trackers were tried. The following evaluation criteria were considered in deciding which tracker is to be used for this project. First, the required tracker should retrieve the largest possible pecentage of the pixels belonging to the pedestrian. If enough pixels were not retrieved for the object then the resulting representation of colour information would not be accurate. Not only is the tracker required to get the largest possible percentage of correct object pixels, but it is also required to minimize the number of pixels identified incorrectly as the object's pixels. Pixels from the background should be discarded. Shadow pixels and background noise attached to the object should also be discarded.

Second, the tracker is expected to retrieve a collection of pixels belonging to the person. Each pixel should be associated with its colour information. This tracker's code should enable retrieving these colour information for subsequent analysis. Third, if several objects were identified in the scene, the tracker should divide the pixels of the foreground into those objects.

3.2. Dr. Baumberg's Tracker – The Pedestrian Tracker

Initially, it was preferred to use the tracker produced by Dr. Baumberg in 1995. The pedestrian tracker was especially developed for tracking and recognizing walking humans. It is based on PDM landmark points. The tracker is capable of identifying the person by drawing a boundary around the external edge of the person.

The pedestrian tracker uses the adaptive background technique to segment the foreground from the background. Several PDMs are then initialised in the foreground and objects are tested against the pedestrian shape models. An edge is drawn around the pedestrian to mark that one person has been detected and is being tracked.

The tracker was implemented in C and runs on an IRIX machine. As figure 3.1 shows, the tracker draws the edges and produces a foreground picture showing the pixels belonging to each person. The tracker has a long list of initialisation parameters that control the performance. For the purpose of the project, the command level interface was studied to see how it could be utilized. The tracker requires an MPEG file with no audio track included. It performs better for 352x288 frame sizes and requires an initialisation stage for the background of 25 frames.

Figure 3.1 Adam Baumberg's Pedestrian Tracker - The Interface

Adam Baumberg's tracker has both advantages and disadvantages from the point of view of this project. The advantages are:

1. The tracker is specially designed for identifying people and pedestrians. Other moving objects are expected to be ignored. Yet, it was found that this is not quite true as the tracker is very good at fitting other objects as pedestrians like bicycles, trees and shadows.
2. The tracker has a wide variety of parameters that could be changed to fit different settings.
3. The tracker writes each frame of the movie under analysis into one image. When the frame information is analysed, one can go back to the frame's image and see what the image really looks like. This helps in figuring out the strengths and weaknesses of the analysis.
4. The tracker discards shadow pixels successfully because the shadow is not included in the shape-based model.

The disadvantages of the tracker could be summarized in the following points:

1. The complexity of the initialisation parameters. The tracker was initially configured for specific ground-plane mapping. The parameters for configuring a new mapping were not explained in the documentation, which made it difficult to change the settings according to each experiment's requirements.
2. The tracker works efficiently for small frame sizes - this results in fewer pixels per pedestrian. The author wished to use a bigger frame size to enable more detailed colour analysis. The tracker did not work properly for bigger frame sizes. The AVERAGE_WORLD_HEIGHT and some other parameters are not relative to the size of the frame and thus had to be amended. Even after amending the initial values, the tracker was capable of identifying pedestrians but failed to scale the PDM down properly. This is because scaling was relative to the small frame size. This problem could not be solved. Trials to figure out other parameters that had to be changed were unsuccessful.
3. The tracker's code is not available and thus only the current output of the tracker can be used. The tracker currently produces an image for each frame. The pixels belonging to each object are coloured differently. In addition, a file is created with the landmark points of each object. All these image files should be processed and used as the interface to the colour analysis code.

4. The major problem with the tracker is its failure to distinguish the exact pixels of the pedestrian. Although the tracker has very high tracking percentages and the same person is always identified and correctly tracked, yet as the person walks the silhouette tends to surround areas that are completely different from those really occupied by the pedestrian, as figure 3.2 shows. It was then decided that although this tracker is better in identifying people, it is not as good in deciding on the pixels belonging to the pedestrian, which is the prime requirement of this project.

Figure 3.2 The Pedestrian Tracker – Error in Pixel Identification

3.3. Dr. Magee's tracker – The Blob Tracker

The blob tracker developed by Dr. Derek Magee was primarily devoted for car tracking and traffic analysis. A more advanced version of the tracker was designed for baseball players tracking. This advanced version was studied. The tracker is a traditional motion tracker that does not specify any details about the exact shapes of the objects being tracked.

The tracker was developed in C++ under a Linux environment. It uses adaptive background for segmentation and works at different resolutions and frame rates. It also requires an MPEG movie where the audio track is excluded. A parameter file allows editing the initial parameters and settings of the tracker. Figure 3.3 shows the interface of the tracker.

Figure 3.3 The Blob Tracker's Interface

The following parameters were necessary for the purpose of the project:

Parameter Name	Required Values	Parameter Description
INIT_FRAMES	25	The number of frames required for the background initialisation
RES_X RES_Y	1 1	The resolution. Full resolution was considered in this project's analysis
GP_CALIBRATION_ANGLE	0 – 20	The tilt angle of the camera
MAX_REL_SIZE MIN_REL_SIZE	2.5 0.75	The maximum and minimum relative sizes of the blobs
BG_INIT_FRAMES	30	The initial number of frames before any tracking is done // Note: this parameter is available in the main program code rather than the header file

Table 3.1 Blob Tracker's relevant parameters

Advantages of the blob tracker:

1. The tracker does not depend on any ground-plane mapping. The only relevant parameter is the tilt angle of the camera mentioned in table 3.1, which facilitates the usage of this tracker for different settings.
2. Although the tracker does not specify the shape of tracked objects, it allows specifying the minimum and maximum sizes of the tracked blobs.
3. The tracker has the ability to merge blobs and a flag is available to allow that. This decreases the disadvantages of blob trackers where the same object is represented by more than one blob.
4. The code of the tracker is available and documented. Dr. Derek allowed editing the code for the purpose of the project.
5. The tracker can work on any frame size and under any resolution. This allows the usage of larger frames for more detailed analysis.

Disadvantages of the blob tracker:

1. The tracker does not produce any resulting information that is of relevance to the project. The tracker's output file only specifies the location of the object in each frame (the centre of mass of the object only).
2. Although the tracker enables merging of blobs, it is very likely to divide the object into two blobs. The specified minimum and maximum sizes of the blob play a major role in avoiding this. Figure 3.4 shows a possible case. The two rectangles indicate two objects being detected rather than one.

Figure 3.4 The Blob Tracker – Error in Merging the Pedestrian's Blobs

3. Because the tracker is not specially designed for pedestrians, it tracks all moving objects in the scene. To use the tracker efficiently, one should ensure the tracker's background does not have other moving objects apart from pedestrians.
4. This tracker does not produce images that belong to each frame for later analysis. The information for each frame is only written to an output file. Checking the accuracy of this information is not possible without viewing the frame's actual content.

As the blob tracker retrieves a better estimate of the object's actual pixels, it was chosen for the project's purpose. The code had to be studied and changed so that the pixels belonging to each label of the frame are recorded in the following format:

Xcoord   Ycoord   R   G   B

The tracker represents the information in RGB colour space. It could then be transferred into any other format.

The tracker code was changed so that the file structure, shown in figure 3.5, is logged during tracking. The library implemented by the author was designed to process this structure. Each file represents the information of one movie passed into the tracker.

```
frameNo
objectNo
object pixel info
+ // end of object info
objectNo
object pixel info
+
- // End of frame info
```

Figure 3.5 Structure of the Input File

4. Project Design and Methodologies

4.1. Requirements Specification

Based on the review of literature in chapter two, it was decided that colour information using histograms or GMMs is worth testing within the course of this project. The case studies of both trackers in chapter three concluded that the blob tracker is more suitable for experimentation. Editing the tracker to generate the interface file was successfully accomplished.

The requirements for the visual signature library are:

- The library should use the interface file structure shown in figure 3.5.
- One driver class should represent each experiment. It should produce an output file with the results of that experiment.
- Five colour spaces are to be used for testing. These are: RGB, HS, Normalized RG, CIE and Normalized CIE colour spaces.
- The two techniques used for data sampling and signature representation are the histograms and the GMMs.
- The library should enable re-usability so that the histograms, GMMs and other similar functionalities could be used for future relevant researches. This also requires proper documentation of the resulting code.
- The library implemented does not need to be a full running system on its own. Its usage is purely restricted to research testing and experiments.
- The library can benefit from the computer vision group library. The computer vision's library, written in C and C++, is available for non-commercial uses. It is allowed to directly reference the library or reuse its algorithms.

4.2. Design Decisions

Considering the requirements in section 4.1, decisions were made on the implementation techniques and the design strategy. Regarding the choice of the programming language, Java was chosen. Hence, the library is platform independent.

The usage of Java as an implementation language was based primarily on the developer's preference. It was also a good chance to start a Java library for the computer vision group. All the code currently available in the library was written and implemented in C or C++. Thus, the author took the challenge of writing the library in Java converting any relevant C or C++ functions from the current library if required. It was also the preference of HP to use a Java library for their researches.

Both Java and C++ have the advantage of using the OOA&D implementation strategy. Object Oriented libraries are primarily designed for re-usability. The standard UML notation that accompanies OOA&D strategy enables well-structured documentation. The standard JavaDoc style was chosen for the final library documentation.

4.3. Data Collection

The project required the collection of data using digital camera. MiniDVC Camera was used to capture the movies of pedestrians. Each recorded movie was captured using Adobe Premier software. Each movie was converted into an MPEG file of size 355*280 excluding the audio track.

A total of thirty-three different recordings are available. These movies cover a wide variety of samples and settings. Details of the settings for each sample are explained under the experiment's relevant section in chapter five.

4.4. Methodologies

The project has a very clear aim. Trials are to be conducted in the search for the visual signature. Because each experimentation cycle affects the way following experiments are conducted, action research and object oriented analysis and design methodologies are used. Action research is used in choosing and carrying out iterative experimentation cycles. Iterative development is used in developing the library too. As a result, consecutive cycles of analysis, design, implementation and evaluation could be used until the final version of the library is ready.

4.4.1. Action Research (AR)

Action Research technique has been intensively used in humanities research. It has been used as a teaching and educational research methodology. In their book "Becoming Critical: Knowing Through Action Research", Carr and Kemmis explained how the action research process happens. It "begins with one pattern of practices and understandings, and ends with another, in which some practices or elements of them are continuous through the improvement process while others are discontinuous (new elements have been added, old ones have been dropped, and transformations have occurred in still others)". (1986, p. 182)

Action Research tries to perform a collection of experiments and tests. It then refines the research techniques and methods according to the results of these tests. An essential feature of AR is the iterative research. It allows research to be conducted in cycles.

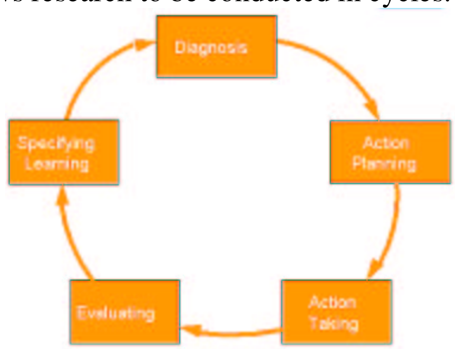

Figure 4.1 Action Research Life Cycle (Scholl 2001)

Figure 4.1 shows the general life cycle of action research. This life cycle is taken from a medical and health context. The research cycle begins with figuring out the problem requirements. The action is chosen based on knowledge in the area, then the experiment is carried out. Evaluating the results of the experiments should follow. This will result in a better understanding of the problem and of the selected techniques. The cycle goes on again.

4.4.2. Object Oriented Analysis and Design (OOA&D)

Object Oriented Analysis and Design methodology was designed for situations where the full understanding of the product is not available at the beginning of the development process. (Braude 1997, p.6) At the beginning of this project, the only clear understanding was the goal of the project and the details of the first experiment. Any later phases depended on the results of the first experiment. Iterative development enables the usage of the results of a previous cycle in the design of subsequent cycles. (Larman 1997, p.297)

Different books and articles listed the advantages of iterative development in general and OOA&D in particular. Some of these advantages, that directly affect this project's specifications, could be found in Oestereich's book entitled "Developing software with UML". These advantages are (1999, p.15):

- Because OOA&D is based on iterative and incremental development, it enables the addition of new requirements during the course of development. This is essential for this project, as subsequent experiments might result in new implementation needs.
- UML is a widely accepted notation. Thus, the usage of UML for documentation enables standardizing the documentation and thus facilitates reading and using the documentation. As the library implemented during this project is aimed at future re-usage, documentation is a necessity.
- Object oriented models tend to be more stable which increases the lifetime of the product. It is, at the same time, easy to amend.
- "Object oriented software development is more fun" (p.15)

Iterative development strategy was chosen for the implementation phase of the project. Iterative development is a development approach that is carried out in cycles. Moreover, each cycle ensures a product can be delivered depending on the current version of requirements. Figure 4.2 clarifies the life cycle of the project using iterative development.

Figure 4.2 Iterative Software Development Life Cycle (Kruchten 2001, p.2)

It should be clearly explained that there is no clear ending to the project that uses iterative development. Each cycle refines the product towards a better usage of the system. The development phase ends when the developer and/or customer realizes the product is satisfactory.

Although the iterative development technique has very clear advantages, one should take the following issues into consideration during developing:

- As each phase has its own requirements and design decisions, one should allow time for analysis and design during each implementation cycle. In other competitive techniques like the waterfall model, it would be enough to brainstorm during the analysis phase; the resulting implementation would then be straightforward.
- The results of each development cycle should be accurately recorded and studied, as they affect the following development cycle.

5. Experiments & Evaluation

5.1. Phase I – Colour Histograms

5.1.1. Analysis and Design

It was decided, based on the literature review, that the first experiment should study the visual signature based on colour histograms. Five colour spaces were selected. These are: RGB, HS, Normalized RG, CIE and Normalized CIE.

Colour space conversion equations are listed in table 5.1. As was explained in section 3.3, the blob tracker provided RGB colour information, conversion to the other four colour spaces should be done before the histogram is created.

Colour Space		Conversion Equations
HS colour space	H	$H = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right)$ (Cheng 2000,p.2263)
	S	$S = 1 - \frac{3\min(R,G,B)}{(R+G+B)}$ (Cheng 2000,p.2263)
Normalized RG colour space	r	$r = \frac{R}{R+G+B}$ (Finlayson 2002, p.261)
	g	$g = \frac{G}{R+G+B}$ (Finlayson 2002, p.261)
CIE colour space	X, Y, Z	$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.6067 & 0.1736 & 0.2001 \\ 0.2988 & 0.5868 & 0.1143 \\ 0.0000 & 0.0661 & 1.1149 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$ (Pattern Recognition Group)
Normalized CIE colour space	x	$x = \frac{X}{X+Y+Z}$ (Pattern Recognition Group)
	y	$y = \frac{Y}{X+Y+Z}$ (Pattern Recognition Group)
	z	$z = \frac{Z}{X+Y+Z}$ (Pattern Recognition Group)

Table 5.1 Colour conversion table

The design stage for this phase can be found in appendix E.1.1 along with the class diagram followed in implementing this part of the library.

5.1.2. Implementation & Experiment

The current version of the library driver, VisualSignature.java, has the input arguments:

VisualSignature inputfile outputfile <width> <height> <beginFrame>

where inputfile is the file generated by the blob tracker. Output file is the file representing the basic statistics about people in the movie. Width and height of the movie could also be specified. The default width and height of the movie is 352x288. BeginFrame enables the program to start from a specific frame number rather than the beginning of the file.

Due to the fact that each frame is only identified by a number, which does not enable the understanding on how colours are really represented, the frame information was written into an image representing the foreground information. The image, named ‘outputimage<frameno>.jpg’, could then be referenced for analysis.

The histograms for the colour spaces were linearly quantized. Each axis was divided into 15 equal divisions. Table 5.2 shows the details of colour space information:

Colour Space	Dimension	minValue	maxValue
RGB	R	0	255
	G	0	255
	B	0	255
HS	H	0	1
	S	0	1
Normalized rg	R	0	1
	G	0	1
CIE	X	0	≈302
	Y	0	≈302
	Z	0	≈302
Normalized CIE	X	0	1
	Y	0	1
	Z	0	1

Table 5.2 Colour Space Value Range

Five different signatures represent each object. The signature is the sequence of bin counts separated by commas. A part of the signature is displayed in figure 5.1. Writing only the counts of the non-empty bins could decrease the signature length, yet this is delayed for the minute until the signature is tested.

0.01183432,0.109467454,0.00887574,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.112426035,0.42307693,0.30473372,0.00591716,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.00887574,0.0147929, …

Figure 5.1 Visual Signature Example

After the signatures are stored into files, the comparing driver programs could be called to generate an output file with the matching probabilities. In comparing, the movie frames are quantized by considering one frame amongst each five consecutive frames. The resulting matching probabilities can then be analysed for comparison and decision-making. The matching of sampled subsequent frames utilizes the adaptation mentioned in the literature. As objects are compared and matched, the most recent object information is considered for subsequent matching.

The sample selected for this experiment, consisted of ten different students wearing casual clothes. Students were not previously informed about the experiment or its aim, thus the clothes they were wearing represented their normal day-to-day style. The students were asked to walk in four different directions. Each student had to walk twice in each direction to avoid any background subtraction issues. The lighting was consistent and all movies were recorded in shadowed outdoor settings.

The average movie length was two minutes. The average of the number of frames per direction was 100 frames. After sampling, 15-20 frames representing the full body were compared for each direction. These only included full body appearance of the pedestrian. The sampled frame information was then passed to the driver programs and analysed. The results of comparing similar and different people were plotted using the spreadsheet software, Excel. Examples of the actual results tables for each experiment are provided in appendix F.

5.1.3. Results & Evaluation

Because the sample selected is not a large one, it would not be representative to produce quantitative results such as averages and standard deviations. It is enough to analyse the techniques by comparing them, i.e. clarifying which methods seem to perform better than others.

The visual signatures of all the colour spaces for the ten pedestrians were generated. These signatures represented two of the four directions recorded. The two profiles from both sides of the pedestrian were considered as the pedestrians moved from left to right and from right to left in front of the camera.

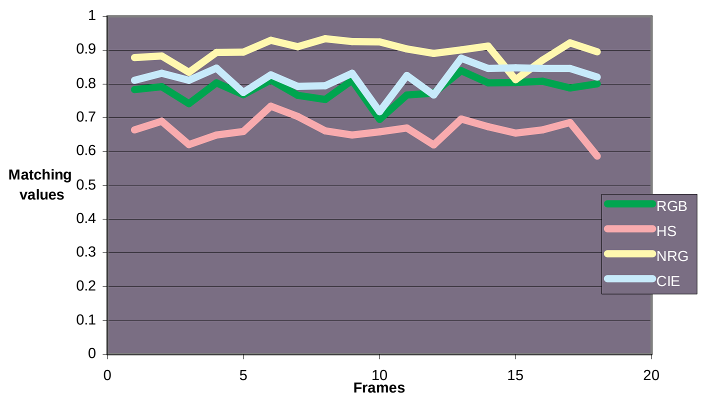

Figure 5.2 Comparing Frames of the Same Person (person 9)

Figure 5.2 shows the matching values for comparing frames of the same person. The diagram shows that as the frames are compared, normalized RG colour space gives the highest probability matching (varies between 80% and 94%). CIE gives the second highest matching (71% to 87%). RGB gives slightly lower matching values (70% to 84%). Finally, HS colour space gives the least matching (59% to 73%). This figure is typical to most other pedestrians. The order of colour spaces is consistent for all the ten different people. An exception to this is that RGB and CIE may alternately switch between the second and third positions. It was noticed that normalized CIE does not give any extra information and it usually matches exactly the HS colour space. Therefore, it was discarded right from the beginning.

The colour space that gives the best matching is expected to produce better visual signatures. Yet, the same colour space should produce significantly lower values for different people. This is to ensure it is good both at matching the same person frames and at differentiation amongst people. Averaging the results for the ten pedestrians results is shown in figure 5.3. The figure shows how the order of the colour spaces matches that produced for the single person in figure 5.2.

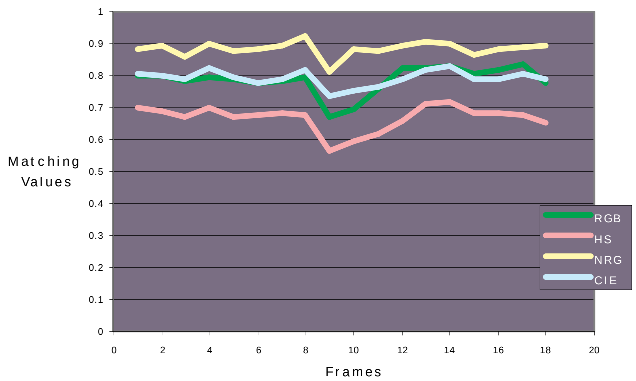

Figure 5.3 Average Results for 'the Same Person' Comparison

Then, the analysis proceeded to compare the colour representation of different people, which is expected to produce lower matching values than those for similar people. A clear separable area should be detected in the diagram to enable creation of a threshold and to distinguish between similar and different people.

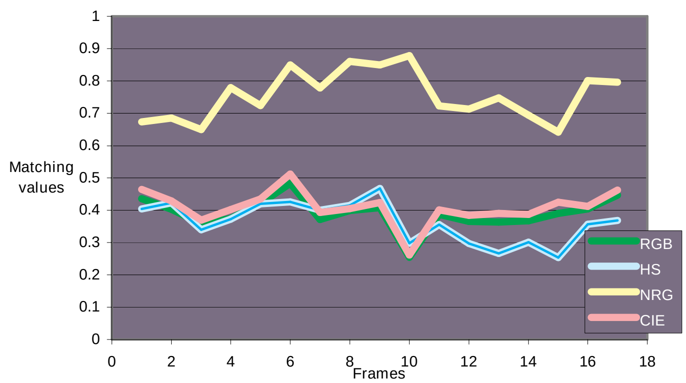

Figure 5.4 Different people Comparison (sample 4 and sample 6)

Figure 5.4 represents comparing the sequences of two different people identified as person 4 and person 6 in the sample. The diagram shows that the three colour spaces, RGB, HS and CIE, produce lower matches (below 50%). Normalized RG, on the other hand, produces much higher matches (values between 64% and 88%). This proves that normalized RG, although it gives higher matching values for the same colour, it is not good at differentiating amongst colours. It is clear that normalized RG colour space simply results in high general matches regardless of the colours.

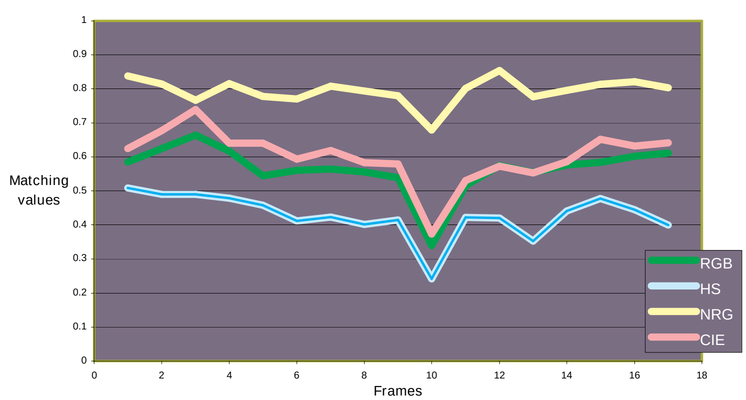

Figure 5.5 Different People Comparison (sample 4 and sample 9)

Figure 5.5 shows the results of comparing two people wearing blue jeans and grey shirts. Though humans are capable of differentiating between these two people using other characteristics (height, etc.), the computer algorithm begins to show ambiguities in colour differentiation.

It was clear, right from the beginning of the project, that colour information is not unique. It is very familiar to run into two people wearing the same or indistinguishable clothes colours. Colour analysis algorithms would not be able to differentiate amongst those two people easily. This is why colour information could never be sufficient for the visual signature. Additional geometric information would help to increase the opportunities of correct identification.

Both CIE and RGB are possible candidates (see table 5.3). Since the blob tracker retrieves RGB colour information, using RGB would avoid the colour space conversion overhead. On average, RGB tends to give values between 25% and 48% for different people. The average of RGB colour space for the same person, shown in figure 5.3, ranges between 66% and 84%.

Colour space	Average Minimum Matching value (%)	Average Maximum matching value (%)	Average value (%)
RGB	25	48	42
CIE	28	48	44
HS	27	42	37
Normalized RG	63	71	68

Table 5.3 Different people matching averages (summary of results)

The next experiment tries to improve the performance of RGB colour histograms by introducing some geometric information that is persistent across different viewpoints.

5.2. Phase II – Three Regions Colour Histograms

5.2.1. Analysis and Design

After the results of the first experiment were studied, and using the AR (Action Research) technique, this experiment introduces a method to improve the RGB colour histograms. The background reading clarified that the major problem facing the wide application of histograms is their lack of geometric information (see section 2.3).

The introduction of detailed geometric information, though, affects the viewpoint-independence advantage. Viewpoint-independence had a major influence on choosing the colour as the visual signature basis. The search is then confined to any geometric information that would not affect the independencies required for the visual signature.

The simplest technique, introduced by Bressan, for adding geometric information is to divide the studied object into regions that could be later compared for colour matching. (2003) The search for such geometric regions that can be viewpoint-independent was based on the simple human understanding of the colours worn by pedestrians.

Taking into consideration the human perception of pedestrians, one can easily divide the pedestrian horizontally into three regions. The first region represents the head and the neck. The second region represents the torso. The part of the body below the waist falls into the third region. The author thus proposed the three regions colour scheme. This suggestion is based purely on observations of our understanding of walking pedestrians. It is very common to describe a pedestrian by saying: ‘the person who wears a blue shirt and black trousers’.

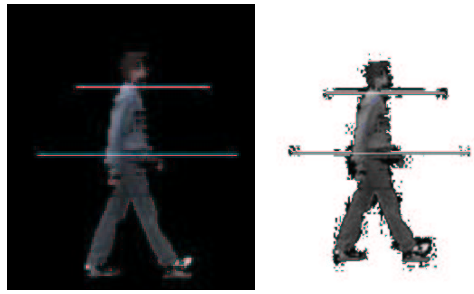

Figure 5.6 The Three-Regions Scheme Division

The automation of dividing the body into the three segments is non-trivial. However, when using the histogram technique, this division does not need to be highly accurate, as any minor deficiency would not affect the total histogram result significantly. This is because the histogram used is scale-independent. The histogram bin counts are divided by the total number of pixels in the region. Thus, a small number of pixels of peculiar or different colour have a minor effect.

The regions are divided automatically using the percentages of the person's height. It is straightforward to calculate the height of the person from the segmented pixels. This is accomplished by subtracting the highest and lowest y-values. Although this is based on the assumption that the ground plane is perpendicular to the camera's image plane, it is still a reasonable estimate for expected CCTV camera angles. The height is then divided using specific ratios into the three regions.

To find these ratios, the ten pedestrians of the sample were studied. Manual detection and marking of the beginning of each region was accomplished. The manual detection was done by the author for ten different frames per person. Then, the percentages of these regions in each frame were recorded. The average percentage of the regions' ratios, based on the 100 frames, resulted to be as follows:

Region1	Region2	Region3
15%	30%	55%

Table 5.4 Automation of dividing the pedestrian into three regions

Typically, the variation in the first region percentage is very limited, while Region3 is expected to vary between around 45% and 60%. Further sampling might affect these results, but the results worked fairly well for the experiments reported.

Before going into implementing this phase, it was worth testing whether the three regions do occupy slightly separable areas of the colour space. This is because the three regions scheme introduces three histograms per pedestrian instead of one, which increases the length of the signature dramatically. If the regions occupy separable areas then the number of nonempty bins per histogram would be less. This ensures the result would not triple the size of the signature.

The colour space of the whole body pixels was represented for a collection of random pedestrian frames using GNUPLOT. Later, the pixels were divided into the three regions by the scheme shown in table 5.4. The colour space of each area was represented too.

Figure 5.7 shows the collective colour space for the whole body of one frame. Figures 5.8, 5.9 and 5.10 show the colour space corresponding to each of the three regions. Each region's representation clearly occupies fewer bins than the whole body. As a result the number of occupied bin counts in the each region's signature would decrease.

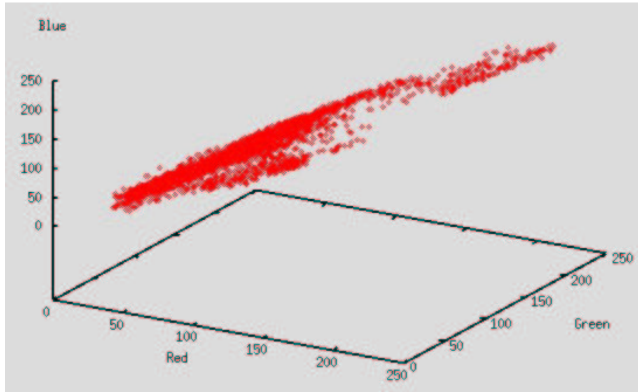

Figure 5.7 Colour space of the Whole Body

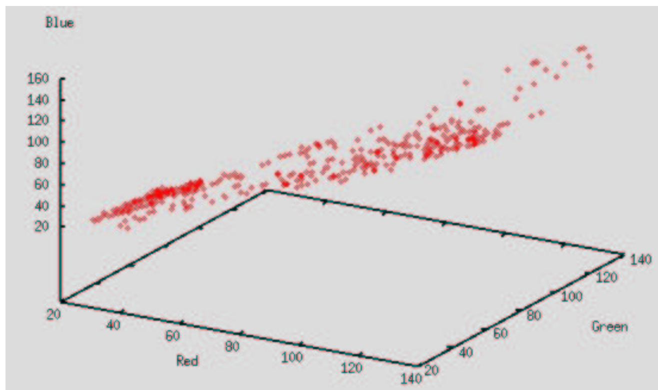

Figure 5.8 Region 1 Colour Space

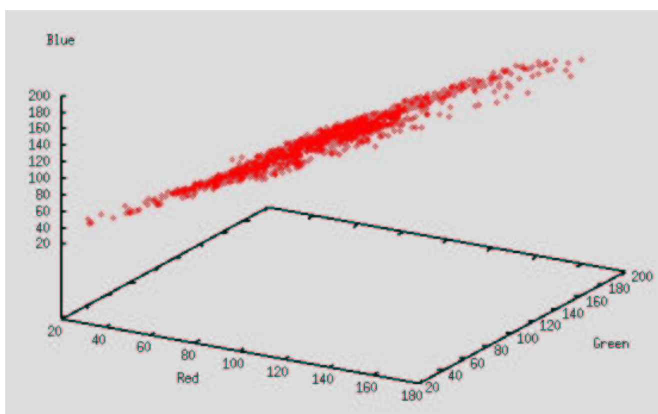

Figure 5.9 Region 2 Colour Space

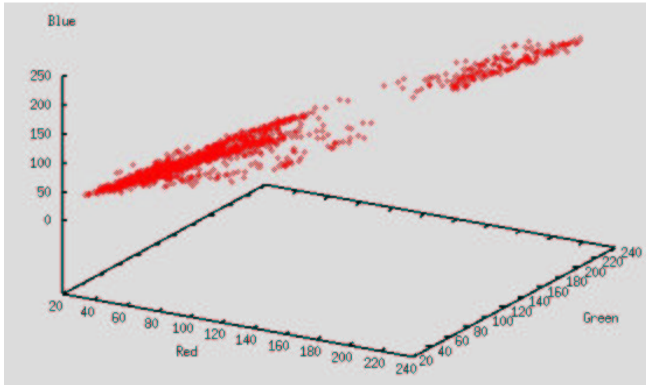

Figure 5.10 Region 3 Colour Space

In comparing the visual signatures, each region's signature, from one frame, was compared against the same region's signature in the other frame. The resulting matches were recorded.

The 'Region' class was added to the library and the updated class diagram can be found in appendix E.1.2 along with details of how the functionality of the code was changed to suit the proposed scheme.

5.2.2. Results and Evaluation

Comparing the three regions for the same person walking in two directions resulted in the values shown in figure 5.11. It is very clear that regions two and three tend to give values ranging between 60% and 90%. Yet, region 1 (representing the head) fluctuates and drops to nearly zero at some frames.

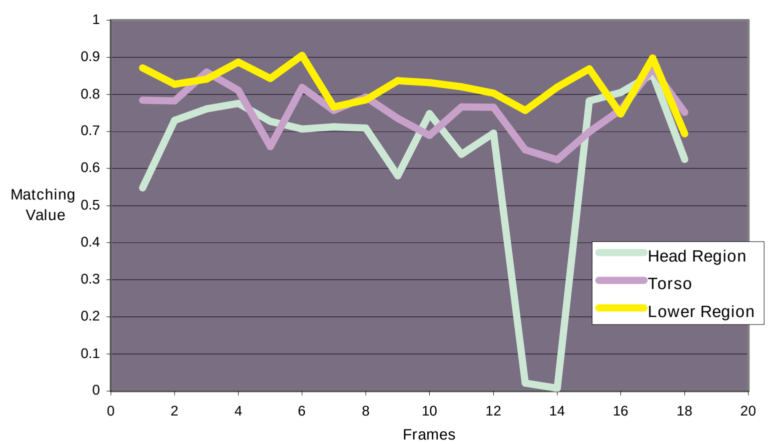

Figure 5.11 The Three-Regions Matching for the Same Person (person 8)

Looking back at the recorded movie, it was noticed that the pedestrian turned his head around at the specified frame. This results in very low matching as the difference between the skin colour and the hair colour is usually very large. It was concluded from this and other similar diagrams that the head region affects the resulting comparison. Its extraction into a separate region enables the other regions' comparison to be more accurate. The head region could either be eliminated completely during comparison or a lower consideration is given to the results of its match.

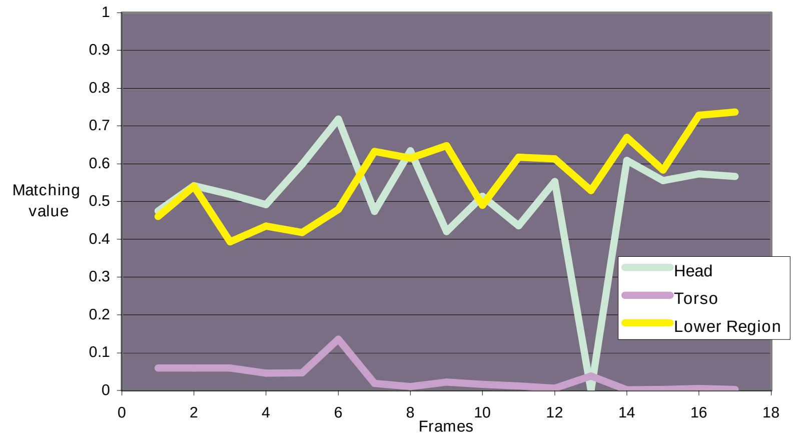

Figure 5.12 The Three-Regions Matching for Different People (person 3 and person 4)

Figure 5.12 then shows another case where the three regions scheme enhances the performance of the visual signature. In comparing two different people wearing blue jeans but different shirts, the three regions scheme enables distinguishing amongst the two persons by detecting very low matching values for the torso region. Moreover, the head region tends to give high matches between different people. This is because differences in skin colour between people tend to be small.

This suggests that low matches for one of the three regions could mean different people. It was noticed that the torso region tends to give more distinguishing values. This is because people tend to wear different coloured and textured shirts. In addition, weighting could then be utilized by assigning a higher weight to the torso region.

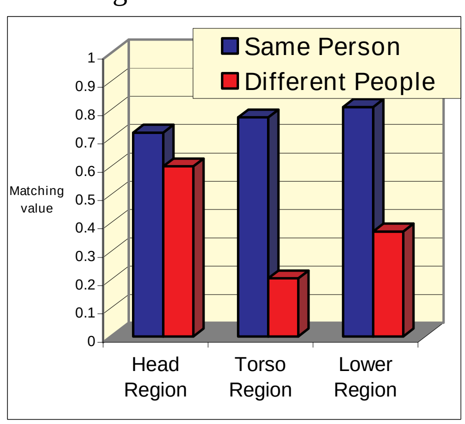

Figure 5.13 The Three-Regions Matching Averages

To verify these findings across the whole sample, figure 5.13 represents the average matching values for the same person and different people. The figure clearly generalizes the finding that the head region tends to give the lowest match when comparing the same person. It tends to give about 60% matching average for different people. The torso region is the most distinguishing region as it gives below 20% average match for different people across the whole sample.

The three regions scheme was then compared for a pedestrian walking in the four different directions. The pedestrians were requested to walk in front of the camera from left to right, from right to left, towards the camera and away from it. Figure 5.14 shows the matching for the whole sequence. It is clear that the matching drops and increases at some points. The torso region's match drops dramatically as the person turns between a profile and facing the camera. A totally different colour of the shirt appeared. The lower region tends to give the most stable results. This is because trousers generally have a consistent colour from all sides.

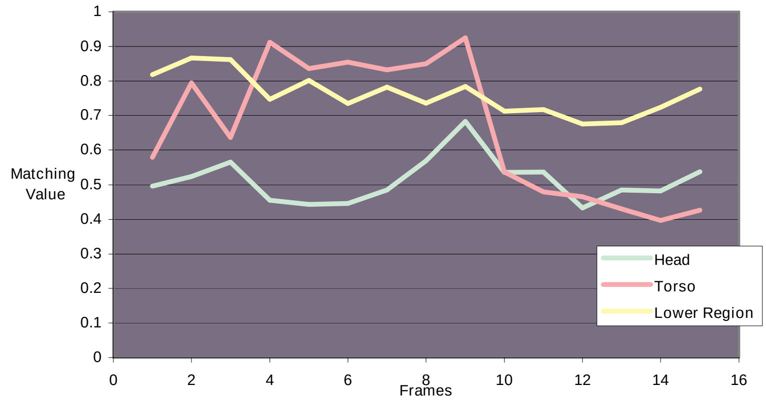

Figure 5.14 Whole Sequence Matching for the Same Person

In conclusion, the three-regions colour scheme has four major advantages. First, it is closer to the human perception. Second, a high match of the three regions increases the probability that the two frames belong to the same person. Separating the matching results of the regions, rather than averaging them out, ensures that the mismatch of one region could be clearly noticed and considered in the resulting decision.

The third advantage is that this technique allows the weighting of the regions for the final matching decision. Different regions could be given different weights according to their matching probability. Based on the analysis above, the head region appears to give high matching for different people. This is because people are expected to have a limited range of skin colours. The Torso region was the most distinguishing region. This is because people wear shirts and blouses of varied and different colours. The highest weight should be given to the second region and a lower weight should be assigned to the first region. Determination of the exact weights needs further analysis of a wider sample.

The fourth advantage is the ability to perform non-linear quantisation for some region's histograms. The head region, for example, might need a finer quantisation in areas of the colour space that represent skin and hair colours. This would enable the first region to perform better in differentiating amongst people.

5.3. Phase III – Three Regions GMMs

5.3.1. Analysis and Design

Although GMMs have become more of a standard technique in many areas of research, their value is highly dependent on their initialisation choices. The first choice in applying GMMs is the number of Gaussians per mixture model. Orwell et. al. used four Gaussians to represent the RGB colour space of pedestrians. (2001) It was then decided to use the same number of Gaussians for this project's experiments.

The EM algorithm (see section 2.3), most commonly used for GMMs, is very sensitive to its initialisation techniques. Most researches initialise the weights of the Gaussians to: $v_j = \frac{1}{n}$ where n is the number of Gaussians used in the GMM. Thus for our implementation, the weight's initialisation was always $v_j = \frac{1}{4}$.

The mean's initialisation represents the most difficult choice in implementing the EM algorithm. It was decided to test the random initialisation technique first, as this is the one most commonly mentioned in other research. The problem with this choice is that the GMMs of the two frames will be compared for similarity. Since the GMM is highly sensitive to its initialisation, it would then be obvious that two GMMs initialised to randomly different means might result to be different, even though they represent the same data.

The initialisation technique used for the covariance matrices was based on the research conducted by McKenna. (1999) The research proposed initialising the covariance matrix of each Gaussian to σI where I is the identity matrix and σ is "the Euclidean distance from the component's mean to its nearest neighbouring component's mean". (p.226) Other initialisation options use the identity matrix only, yet McKenna et. al. discuss that their technique produces better results.

5.3.2. Implementation

The implementation stage of the GMM and EM algorithm was the most difficult implementation phase. This is because of the algorithm's complexity. In addition, it was not until the whole code was implemented that proper testing could be carried out. It was then difficult to debug the code.

In addition, all the literature read in the topic did not mention some particular implementation concerns. It was necessary to review an already written code on EM and GMM to find these hints. The code, provided by Dr. Derek Magee, enabled the refinement of the code to function properly. The list of issues that need to be taken into consideration in implementing EM algorithm is:

- In several cases, the covariance matrix could result to be a singular matrix. The matrix should then be adjusted. In converting the singular matrix to a non-singular one, all elements, that do not lie on the diagonal of the matrix (i = j), are multiplied by a float number slightly lower than 1. This suggestion for adjusting the covariance matrix was implemented as follows:

```
for (int i = 0; i < dimension; i++)
     for (int j = 0; j < dimension; j++)
        if (i != j)
             covariance[i][j] *= 0.99;
```

- The determinant of the covariance matrix drops to very low values sometimes. This would cause considerable unnecessary calculations. A threshold should be set to enable stopping the calculations and setting the parameters to zero when the determinant drops to a fairly low value.
- The conditions to stop the iterations of the GMMs also vary according to implementation. One of these is to calculate the value $\sum_{i=1}^{m} p(x_i \mid j)$. The iterations stop when the value tends to stay constant for each Gaussian. Yet, this requires considerable calculations per iteration. Many researchers tend to run their iterations for long enough before stabilization. It was found that the "algorithm should converge before about 100 iterations". (Schwardt 2003)
- The Euclidean distance was used to calculate σ, which is the closest distance between the mean of the guassian and each of the other three Gaussians' means.

The most challenging problem facing the usage of GMMs for the visual signature is searching for a standard technique to compare two GMMs after creation. A GMM was generated for each frame. The parameters of these GMMs were stored. Later, a comparison should be done between these parameters to ensure similarity and matching. It was previously mentioned in section 2.3 that there is no standard technique to compare GMM parameters. (Orwell 1999, p.15)

The use of GMMs in a computer vision context has been confined to creating a model that represents the training data. Then, collected data are compared with the model, and a probability that this data belongs to the model can be retrieved. For the purpose of the visual signature, each frame's information is modelled into a GMM. Then, the parameters of the GMMs need to be compared to ensure if they were derived from the same set of pixel colours.

A thorough search was conducted to find a standard technique similar to the histogram intersection. Yet, it was not a successful search. A standard distance measure, the Euclidean distance, was used instead to represent the difference between the parameters.

The four Gaussians were ordered by weight so that the Gaussian with the highest weight would be compared against the highest weight Gaussian in the other GMM. The Euclidean distances between the corresponding weights and means were summed to represent the difference value. In this case, high values represent different GMMs rather than similar ones.

5.3.3. Results and Evaluation

The initialisation mechanism of the GMM had to be changed, because the random initialisation did not provide the required results. As it was expected in section 5.3.1, the same set of data points produced widely varied GMM parameters under different random initialisations. The following choices were then implemented:

- Multiple fixed initialisations: as the random initialisation does not provide the same start for both GMMs, fixed initialisation should do. Yet, some fixed initialisations might not suite the underlying data. It was thus decided to select more than one fixed initialisation. Later, the GMMs with the same fixed initialisation are compared. The difference between the two GMMs was chosen to be the lowest difference amongst the pairs of different initialisations. Yet, the technique complicated the code and made it much slower as several GMMs had to be created per data set. It also did not improve the matching performance.

- Data-dependent initialisation was then tried. Any initialisation that depends on the data is expected to solve the problem. The same and similar input data sets would then produce similar initial means. Thus, the vector-quantization technique was used to find the initial means. The class 'VectorQuantization.java' was implemented and added as a utility class. K-means algorithm was used to calculate the initial means. The class was implemented to ensure re-usability for any kind of data and clusters. K-means algorithm iterates as follows:

```
1. Initialise x means where x represents the number of clusters
2. Iterate around the data points
        Decide the closest cluster
3. Calculate the new mean of the cluster
4. Return to step 2 until stability
```

The GMM signature matching was checked for the same and different people using the vector-quantisation initialisation. Figure 5.15 shows the matching results for the three regions scheme of the same person, while figure 5.16 shows the same results for different people. It should again be emphasized that the y-axis in this and subsequent figures represent the difference in the GMM parameters. The higher the y-value is, the more different the inputs are.

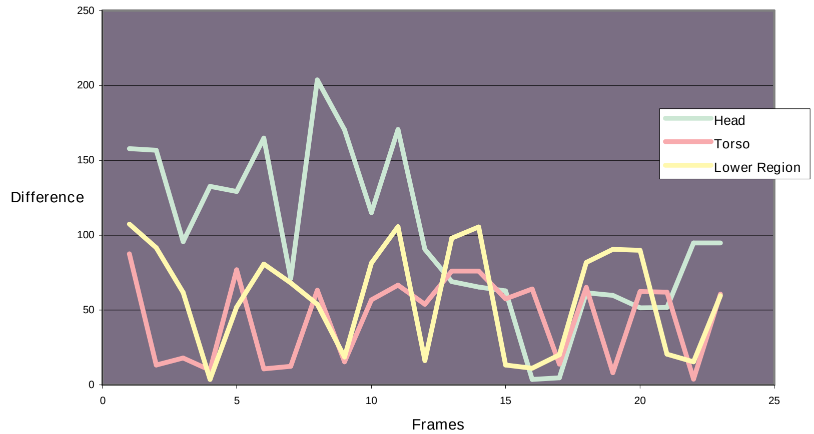

Figure 5.15 K-means Initialisation GMM for the Same Person (person 8)

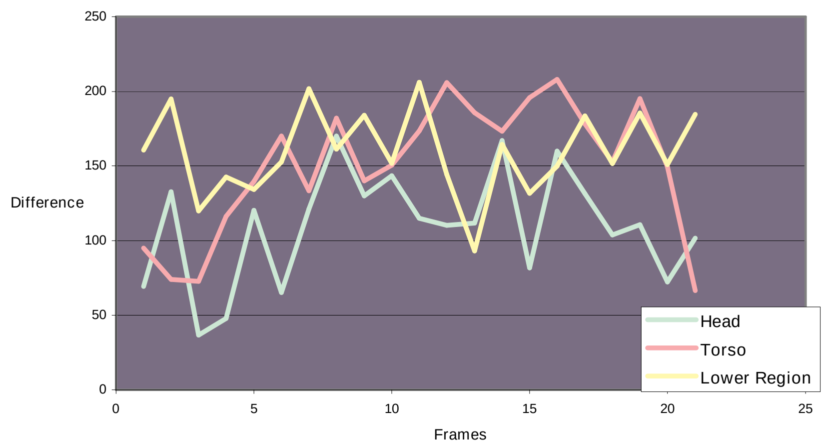

Figure 5.16 K-means Initialisation GMM for Different People (person 6 and person 8)

In general, the figures tend to prove the same results as the colour histogram technique. The head region performs worse than the other regions for the same person. Much higher differences were detected. On the other hand, it results in lower difference (higher similarity) for different people. Different people produce higher overall difference, ranging from 66 to 207 for the torso and the lower region. These regions have lower difference for the same person; results range from 4 to 107.

The averages of these results for the whole sample could be found in figure 5.17. The figure shows the mean and standard deviation results of the GMM across the entire sample. It is clear from the figure that although the same person gives lower difference than different people, there is still no clear line (threshold) that can be considered as a decision value. This is because the standard deviation is very high.

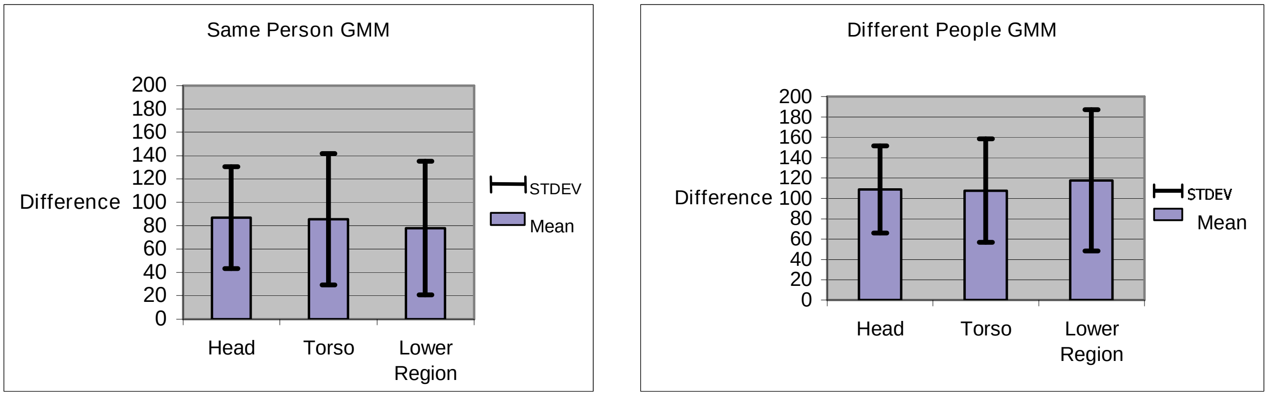

Figure 5.17 GMM Average Results for the Same and Different People

As a result, it was clear that the colour histogram gives better results than the implemented GMMs. Yet, this conclusion is far from generalisation due to the following reasons:

- The failure to find a standard method for comparing GMMs, while histogram comparison used the well-defined histogram intersection technique.
- The effect of the choice of number of Gaussians in the GMM was not researched due to the time limit. This could have affected the results considerably.

The major fault in the design and analysis phases of this cycle was in not considering the comparison technique early enough. Although the code of the stage took a considerable time to accomplish, it was not used efficiently because of the comparison technique dilemma.

At the end of the third implementation phase, the resulting packaged classes of the library could be viewed in figure 5.18.

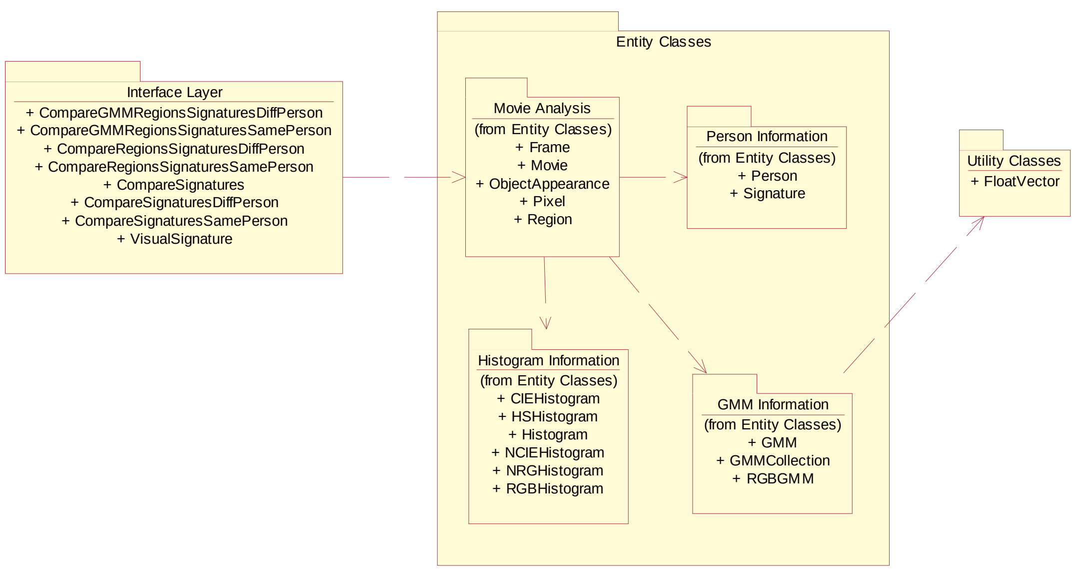

Figure 5.18 Packaged Layout of the Developed Library

5.4. Phase IV – Illumination Changes

The lighting conditions of the scene pose a major threat to the proposed visual signature. This is because colour information can change dramatically under different lighting conditions. This phase tests how much would the signature be affected by the lighting changes.

5.4.1. Analysis and Design

A new sample had to be collected for the purpose of this experiment. Five students were recorded walking in two different directions. The aim of the experiment was to check the effect of lighting conditions on the colour spaces. The three colour spaces under investigation were the RGB, CIE and HS colour spaces. The colour histogram signature was used because it gives the best matching values. The students were asked to move from shadowed to sunny locations. The outdoors lighting was chosen to test this experiment because it is expected to produce a clear lighting contrast.

Figure 5.19 Lighting Conditions Settings

The experiment consisted of two parts. The first part followed the pedestrians walking from shadow to light. The frames were sampled in the same way explained in section 5.1.2 and compared. The comparison detected the gradual change in lighting between each frame and a subsequent one. The second part of the experiment compared the maximum change value. Thus, the fully shadowed and fully lit frames were compared for matching the same person.

5.4.2. Results and Evaluation

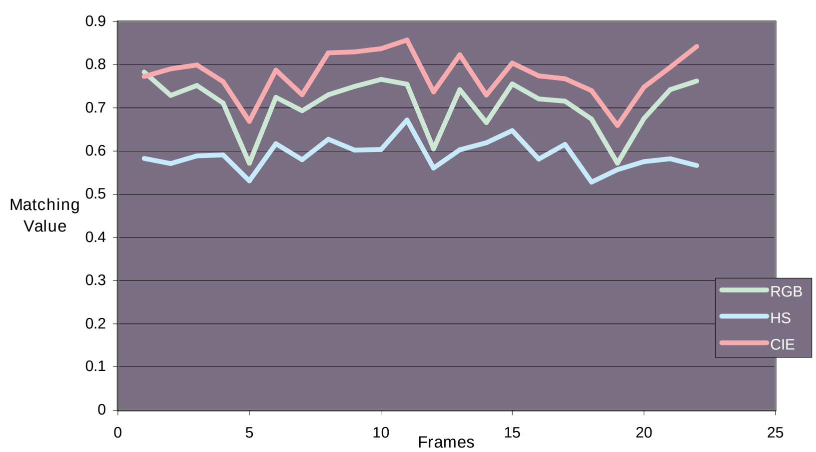

Figure 5.20 Slight Lighting Changes for the Same Person (person 3)

Figure 5.20 shows the sequence matching for a person moving under a change in lighting. It clearly shows that the CIE colour space is more immune to slight lighting changes. Yet all the three colour spaces were capable of coping with the gradual lighting changes. They gave results that are similar to those under consistent lighting.

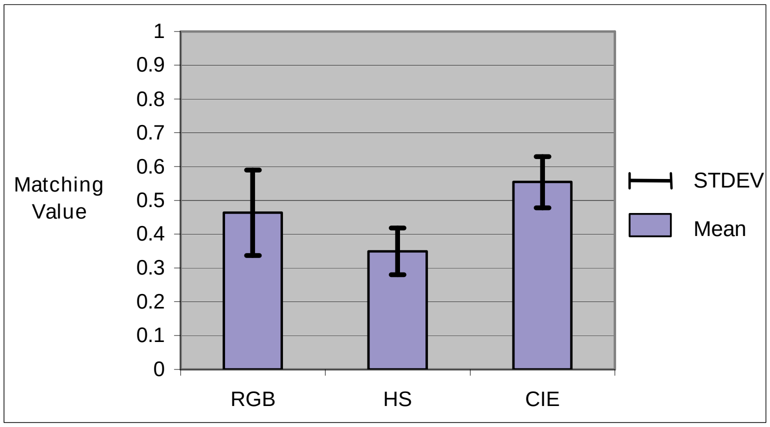

Figure 5.21 Average Matching Values of the Colour Space for Large Changes in Lighting.

The challenge facing the visual signature, then, is the wide change of lighting conditions. Figure 5.21 shows the resulting matching values where the shadowed and lit frames are compared for the same person.

As expected, it is clear from the figure that the lighting affects the colour considerably. All the colour space comparison was affected. Table 5.5 shows the average of these three colour spaces for a constant and widely varied lighting conditions. A clear, and expected, drop in the matching values could be detected.

Colour Space	Average matching under fixed lighting (%)	Average matching under varied lighting (%)
RGB	79	46
CIE	80	55
HS	67	35

Table 5.5 Average colour space matching values for consistent and Changing lighting conditions

This test tried to give an idea of the lighting changes results. It is not an extensive test of different lighting conditions. Only one lighting condition was tested. Indoor lighting was not tested at all.

Though the signature suffers from wide lighting changes, current researches propose methods to decrease the lighting effect. Some of these methods could be found in the future directions in section 7.1.

6. Project Management

6.1. Project Plan: Before and After

The initial plan, included in the interim report, was as follows:

#	Stage Name	June Week1	June Week2	June Week3	July Week1	July Week2	July Week3	July Week4	August Week1	August Week2	August Week3	August Week4
1	Further research	--------						----	----			
2	Software familiarity & data collection		--------									
3	Solution A&D		----	--------				----	--------			
4	Solution Implementation				--------	--------	----					
5	Solution Testing & data collection						----	--------				
6	Evaluation and Enhancements A&D								--------	--------		
7	Write up	----			--------					--------	--------	
8	Wrap up											--------

The plan had to be shifted due to the following reasons:

- The experimentation and implementation phase was divided into cycles. Each cycle included the analysis, design, implementation, experimentation and evaluation. Thus the evaluation was not delayed until the ending of the implementation as the initial plan proposed. This decision was made only at the end of the background reading and software familiarity stage.
- New data collection has to be conducted for the last phase of implementation.
- The IEEE conference paper listed in appendix D consumed a considerable amount of the time and effort. Yet, it was a very good way to summarize and re-order the project's findings.
- The software familiarity stage took two weeks instead of one because the software expected to be used (Dr. Baumberg's tracker) did not satisfy the needs of the project. Thus, a new software had to be studied.

The resulting working plan:

#	Stage Name	June Week1	June Week2	June Week3	July Week1	July Week2	July Week3	July Week4	August Week1	August Week2	August Week3	August Week4
1	Further research	--------	--------	----								
2	Software familiarity & data collection	----	----	--------						--		
3	Phase I			----	--------			----	--------			
4	Phase II					--------						
5	Phase III						--------	--------				
6	phase IV									----		
7	Presentation							----				
8	Conference paper writing								--------	----		
9	Write up		--------							--------	--------	
10	Wrap up											--------

It should be mentioned that dividing the month into four weeks helps in following up with the plan in the remaining two or three days of the months. This enables extra few days for emergencies and delays.

6.2. Project Log

A major concern in research-oriented projects is to keep track of the notes and observations for the following experiments. Notes help in clearing the researcher's mind and enable later review and evaluation of the undertaken decisions. It reminds the researcher what his previous thoughts and ideas were.

Thus, the author decided to use the online project log to keep track, not only of the schedule but also of the notes, observations, results and general ideas. The project log was updated at the end of each working day; the author devoted some time to write the progress alongside her thoughts and ideas in the log.

Not only was the project log used for the author's own ideas and notes but it also included notes and quotes from all the background material reviewed. After reading each article, the author used to add a section in the log regarding the advantages of the article from the project's point of view. Important quotes and references were also written there. The articles were fully referenced at the end of the log to facilitate later usage of these quotes. Cross-references were provided between the quotes and their reference details.

This decision was very helpful all through the project. The project log was used as the author's own personal memory of the project's development. During writing this report, the log was the main source of ideas. The author did not need to depend on her memory or any other sort of scattered notes.

The author used to search the log for ideas and notes. It enabled a continuous evaluation of the project's progress. The log was also used as a communication link with the supervisor and the external body. Moreover, the project log included some guidelines for future researchers in the field.

6.3. Key Challenges

The project was not only new in its content but also involved other challenges.

6.3.1. Literature Review Challenge

Starting the project by gathering the techniques available in the literature was a major challenge. This is because the author was not aware of some basics required for understanding the literature. The area of research was new and thus the literature was not an easy read.

Finding the relevant literature was difficult too. The visual signature, or any technique related to it, is not discussed in the normal tracking researches. Current tracking researches are more concerned about objects segmentation from the background. It thus required further and more detailed research into the general computer vision literature. A parallel search in related fields of research was also essential.

The articles reviewed were highly specialized. Thus, suggestions or techniques mentioned were usually brief and not clearly explained. References and previous researches needed to be traced and reviewed to enable clear understanding.

Another major problem faced in the computer vision literature generally is the lack of consistent results. Each researcher proposes and tests a technique, then tends to claim that this technique works efficiently. Conflicts and contradictions could be easily found in the literature. Each research had its own individual findings.

6.3.2. Tracker Choice Challenge

It was clear from the beginning that this project is not about how the object can be segmented from the background. So, there was a need to use a tracker that retrieves the visual information belonging to the tracked person for analysis. The search for the tracker was nontrivial.

After deciding on the tracker, this tracker had to be studied and an interface had to be built for the tracker to retrieve the information required for analysis. In addition, the tracker's performance had to be evaluated to avoid its faults and flows. This would avoid these faults from affecting the results of this project's experiments.

6.3.3. Sample Selection Challenge

The project only analysed data especially collected for it. The choice of the sample is a trade-off between sufficiency and ease of collection. Each collected data takes time for analysis and thus a large sample is not easy to obtain. At the same time, the sample should be sufficient for producing initial results. The samples are not used to retrieve quantitative evaluation of how many times the technique is expected to fail. It is only needed to differentiate between promising techniques and those that prove to be non workable.

In conclusion, collected samples should be carefully selected for the purpose of the experiment carried out. Different samples were needed for different experiments.

6.3.4. Experimentation Challenge

The techniques had to be implemented for testing. This results in the detailed understanding of the algorithms. The coding stage was not the main aim of the project. Yet, correct testing of the techniques directly depended on the algorithms and code used.

6.3.5. Evaluation Challenge

A major obstacle hindering the computer vision research is the lack of standard evaluation techniques. No clear rules have been set for evaluation. Lately, tracking data sets have been distributed by some bodies like IEEE. Such data sets are used for evaluating object trackers. Hence, these evaluation data sets are not applicable to this project's experiments. A major concern for all researches in the field is the lack of any standards for measuring the performance of the systems and algorithms. Agbinya and Rees stated, "no established standard is available for comparing the tracking performance of video object tracker". (1999, p.301)

6.3.6. Project Management Challenge

As this project does not have a clear ending to reach and stop at, it was left to the author to decide on how to manage the project's duration. It was difficult to make choices regarding which techniques to pursue, as the time limits of the project did not allow experimentation with a wide variety of techniques.

It was also the choice of the author to adjust the plan and the subsequent testing according to the findings of the experiments. The author needed to not only concentrate on the experiments but also keep track of the time. Implementation priorities were adjusted based on both the time and the experiments results.

6.4. Challenges Evaluation

6.4.1. Literature Review Evaluation

The project involved a collection of articles to be read. The author had to keep track of all the articles read and compare them. Classifying the articles and summarizing the key points of each article was essential. Because the topic was new to the author, some of the earlier reviewed articles were not understood fully. As more articles were read, the more recent articles, which were understood in more depth, had a wider effect on the path of the project.

The analytical understanding of the articles was the major judge towards deciding which techniques are to be pursued for implementation. Due to the fact that the visual signature search was not conducted before, a search in other topics was necessary. The database imaging context was found to be useful and the majority of the articles reviewed resulted to be in that field.

In addition, the search for articles depended on the already read and analysed articles. If one technique seemed to be promising, then the author started a search for other articles that support or criticize the technique. This would help in realizing the real advantages and disadvantages of the technique. Subsequent articles by the same author(s) sometimes enhanced on the technique or proposed other more beneficial techniques. Though some gaps were still found after the literature review, the resulting comparison was a good beginning for the project.

6.4.2. Tracker Choice Evaluation

The evaluation of the advantages and disadvantages of each tracker could be found in chapter three. The evaluation criteria followed was simply based on the requirements of the project rather than the degree of success in tracking.

6.4.3. Sample Selection Evaluation

The two samples selected for experiments were sufficient but definitely not comprehensive samples. Though the sample was only used to compare techniques rather than evaluate each technique individually, it was a compromise between the size of the sample and the analysis time of the sample.

Future enhancements of the project should include testing the chosen technique against a wider sample under different conditions. The sample’s choice also depends on the application’s context. Deploying the tracker in schools where students wear similar uniforms or in business environments, affect the sample’s choice. It was then left for the application-specific samples to be tested against the technique.

6.4.4. Experimentation Evaluation

The implemented library was designed for future re-usability. It was properly documented and followed all the design requirements of section 4.1. In addition, the choice of developing the library iteratively made the development process easier and clearer. Details of the library design and documentation could be found in appendix E.

The GMM implementation stage was the most difficult implementation context. This is because the algorithms of the GMM creation and the EM algorithm were not clearly explained in any of the reviewed literature. Different implementation techniques could be used to implement them. The result proved to be a satisfactory Java implementation of the algorithm and could be used for future research although its usage for the visual signature did not seem very promising.

6.4.5. Evaluation Choices

The evaluation of each design choice could be found in this report under the choice's relevant section. Evaluating each phase was essential to pursue the subsequent phases in the implementation. Each phase contained an evaluation section where the results are explained and criticized.

It was decided that a major part of the evaluation is to produce clearly understandable diagrams that compare the techniques with each other. Further evaluation results could be extracted from the basic statistical measures. Due to the evaluation challenge mentioned before, choosing the comparative way of evaluating the techniques rather than simple quantitative results made the results more trustable.

6.4.6. Project Management Evaluation

The project was successfully carried out in the amount of time given. Setting close and far-away deadlines was a good technique to keep track of time, and discussing the plan with the supervisor in each meeting helped in making optimal use of time.

Balancing the concentration between the actual experimentation and the project management was not easy to achieve. To ensure the project was on track all the time, meetings with the supervisor were not fixed for a weekly time but were adjusted according to the needs of the project progress. Before making a major decision or choosing a particular track, a meeting was conducted to exchange ideas and ensure the decision made is fully justified.

6.5. Focus Group Evaluation

Due to the lack of a standard evaluation explained in section 6.3.5, at the end of July two evaluation sessions were conducted. The sessions were proceeded by a thirty-five minutes presentation explaining the project's progress and findings. Due to the fact that the author is new in the field of computer vision, it was very important to know the opinion and comments of people in the field about the topic.

The first presentation was held in the school of computing for the members of the computer vision group. The presentation was followed by a discussion. The second presentation was held in HP labs, Bristol. The members at both sessions valued the findings of the project.

The computer vision group at the University of Leeds approved the findings of the project and ensured the lack of a comparison technique for the GMMs. A suggestion was given to try representing each GMM as a histogram before comparing it. The technique was presented by McKenna. (1999) Moreover, they emphasized the importance of a wider testing of the suggested technique. They also approved the need for a context-specific sample before evaluating the technique.

The presentation in HP was followed by two suggestions. The first is using the cross-entropy techniques to compare GMMs. The second concern was about using the compression techniques for the signature. These are within the future work for this project due to the time limits.

7. Conclusion

'Visual Signature for Large Scale Tracking' project tried to utilize the current literature in search for the visual signature. The project concluded that colour could be used as part of the resulting visual signature vector.

The findings of the project were supported by other researches too. The research entitled "Real-time closed-world tracking" concluded that "average color is a reliable discriminator between many objects" when the shadow represents a small percentage of the segmented pixels. The research mentioned that for a frame rate of 3KHz or higher, average colour changes between consecutive frames is so small, which enables comparison and matching amongst objects. (Intille 1997, p.669) Moreover, under controlled lighting conditions, Gevers and Smeulders concluded that RGB colour space is the most appropriate for multi-coloured object recognition. (1999, p.463)

Though the project's duration was short, the project was able to implement and try several solutions. A promising suggestion, the three-regions scheme, was proposed and tested. The suggestion was a better solution than the original colour histogram.

7.1. Future Work

As the project was designed to investigate the possibility of using colour information for the visual signature, it is a major part of the project's evaluation to state the current stage of this research and the probable future stages.

The project aimed at producing a comparison of the techniques rather than an evaluation for each technique on its own. Hence, it is now necessary to pursue the promising technique (the three-regions colour histogram) and produce quantitative results for the performance of the technique over a large sample. This would require implementing the technique as part of a real-time system, and producing the percentages of correct identification.

It was clear from the results of the last experiment phase that an extra solution should be provided for various lighting conditions. Although the lighting experiments did not cover a wide variety of lighting conditions, it still clarified the need for lighting robustness. Gevers and Smeulders summarized the requirements of lighting robustness in three points (1999, p.454):

- Robustness to a change in the direction of illumination.
- Robustness to a change in the intensity of illumination.
- Robustness to a change in the spectral power distribution (SPD) of the illumination.

One interesting research carried out by Paul et. al. was devoted to the change of the location of a pixel colour in the RGB colour space under various lighting conditions.

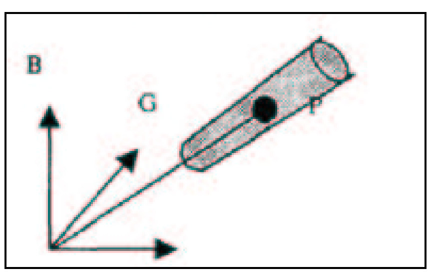

Figure 7.1 RGB Truncated Cone for Various Lighting Conditions (Paul 2001, p.138)

The research concluded that a point in the RGB colour space would always stay within a "small-truncated cone" under lighting changes. (2001, p.138) Two thresholds have to be identified for the angle of the cone and its length. A future suggestion is to apply this idea to the histogram so that the whole histogram shifts within the cone in figure 7.1 under varied lighting conditions.

Other researchers have discussed the possible shifts of other colour spaces under varied lighting conditions. HS colour space suffers a shift under lighting changes. The patches of HS not only shift under various lighting conditions but also have different statistical conditions under different intensities. Each has "different centre of gravity, different shape and different standard deviation". (Yong 2001, p. 1660) Yong's research suggests a B-spline curve function to model the shift.

As the fourth experiment results showed, the signature can perform properly under gradual lighting changes as the person moves within the same camera. Yet the lighting problem arises as the person moves between cameras. The lighting change can be studied between cameras and used later for identification. The transformation T between two cameras can be found and then used for unknown data. (Orwell 2001, p.406) This would enhance the usage of the signature for multi-camera tracking. An extended suggestion is to divide the lighting conditions of the scenes into categories according to the daylight time and find the appropriate transformations for each daylight time. (Tsin 2001) This suggestion enables the outdoor lighting conditions to be changed according to the system time. Xu and Ellis suggested detecting the fast illumination changes "like flood of sunlight, shadow or lights switched on" by monitoring the change of the intensity average of the whole frame. (2001, p.163) Their technique could be used to detect the situations where the signature would be expected to fail.

The current signature is created using linear quantisation of the colour space. Another idea that requires further investigation is utilising the three-regions scheme by changing the linear quantisation into non-linear division according to the needs of each region. The head region, for example, might perform better under non-linear quantisation. The skin colour regions of the colour space require fine-grained quantisation. Kettnaker and Zabih used the non-linear quantisation for human clothing areas. They designed the HSV colour space to distinguish between close colours like beige, off-white or denim while being coarse enough to handle the changes of lighting conditions. (1999, p.257)

Another enhancement of the RGB colour histogram signature is the usage of other comparison techniques. Fieguth and Terzopoulo's proposal explained in section 2.3 could be tested. The X-squared distance measure could also be used.

If H and M are the two histograms of n bins then

$$X^2 = 2\sum_{i=1}^{n} \frac{(H_i - M_i)^2}{(H_i + M_i)}$$

(Mason 2001, p.155)

It was clarified from the beginning of the project that the signature could not be only confined to colour information. The rest of the feature vector components need further investigation. The readings proposed a different set of vectors that might be utilized. (Orwell 2001; Intille 1997; Grimson 1998) The list included: size, velocity, direction of motion, current and past position, height and width (or their aspect ratio).

Though the current colour spaces are expected to perform quite well for the purpose of the signature, the search for a closer colour space to the human perception should continue. “Even though computers can represent over 16 million colors, people tend to use relatively few names for colors”. (Du & Crisman 1995, p.361) The human perception of colour is a topic under search in other disciplines like psychology. The cooperation with these disciplines is important to find the required optimal colour space.

Other colour spaces try to use ratios rather than exact pixel colour information. These colour spaces were not tested and are worth a trial. The one that the author thinks seems more promising is the m1m2m3 colour space proposed by Gevers and Smeulders. If x1 and x2 are two neighbouring pixels then the three components of this colour scheme can be derived as follows (1999, p.458):

$$m1 = \frac{R(x1)G(x2)}{R(x2)G(x1)} \qquad m2 = \frac{R(x1)B(x2)}{R(x2)B(x1)} \qquad m3 = \frac{G(x1)B(x2)}{G(x2)B(x1)}$$

Future work on the visual signature should also involve compression of the signature using different compression techniques. Compression of the histogram representation is thus another future work.

7.2. This Project…

The project aimed at finding the visual signature that would satisfy the needs of public surveillance systems. The project started by carrying out a thorough background review. A search for articles in the field or close-by fields of research was conducted. Articles were read, classified and analysed. Conflicts between articles had to be studied theoretically. Promising techniques were implemented.

Four different cycles of implementation, experimentation and evaluation were conducted. The first cycle tested the colour histograms as visual signature information. Different colour spaces were compared. RGB proved to be the best in both matching the same person’s colour and contrasting different people’s clothes. The second implementation cycle involved testing the new idea of the three regions colour scheme. The three regions colour histogram had several advantages over the traditional histogram. The third cycle tested the GMM as a data representation. Due to the lack of a standard comparison technique, the three regions GMMs produced worse results than the histograms. The last phase tried to study the effect of the lighting on histograms. It is very clear that the histogram is severely affected by abrupt lighting changes. The visual signature though could stand gradual lighting changes within the same camera. Each phase was evaluated to decide on the subsequent phase.

Thus, several techniques were implemented, compared and contrasted. The project shows the advantages of using the three regions RGB colour histograms for the signature required. The suggestion proved to be the best trade-off under controlled situations.

A library was also developed during the course of the project that could be used for future researches. Java implementation of histograms, GMMs and other algorithms is now available and documented. The implementation phase followed the iterative development strategy.

The project started with questioning whether the visual signature is at all achievable. It ended by concluding that the feature vector representing the signature should utilize colour information. The suggested form of colour representation is the three regions scheme colour histogram. The colour space choice is still a matter for discussion, though RGB colour space seems to be sufficient for controlled environments.

The project is a step towards the visual signature that is expected to change the current understanding of security systems. Future visual surveillance systems can produce a daily report on each person’s activities and where he or she has been. The ethical concerns of such systems are within another area of research that is left to the social and humanities field.

References

Abginya, J. and David, R. (1999). Multi-Object Tracking in Video. Real-Time Imaging 5(5), pp. 295-304.

Bishop, Christopher (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

Braude, Eric. (1997) Analyzing and Designing Object-Oriented Systems USA: IEEE Standards Office.

Bressan M.; Guillament D; Vitria J. (2003). Using an ICA representation of local color histograms for object recognition. Pattern Recognition 36(3) pp. 691-701.

Carr, W. and Kemmis, S. (1986) Becoming Critical: Knowing Through Action Research. UK: Deaking University

Chang P.; Krumm, J. (1999). Object recognition with color cooccurrence histograms. IEEE Computer Society Conference on. Computer Vision and Pattern Recognition, (2) pp.498-504.

Chen TQ; Lu, Y. (2002) Color image segmentation – An innovative approach. Pattern Recognition 35 (2), pp. 395-405.

Cheng HD; Jiang XH; Sun, Y; Wang J. (2001). Color image segmentation: Advances and prospects. Pattern Recognition 34 (12), pp. 2259-2281.

Cootes, T.F.; Edwards, G.J.; Taylor, C.J. (1998). Active Appearance Models. Proc. European conference on Computer Vision (2), pp. 484-498.

Du, Yue; Crisman, J. (1995). A color projection for fast Generic Target Tracking. IEEE/RSJ International Conference on Human Robot Interaction and Cooperative Robots (1), pp. 360-365.

Duda, Richard (1997). Pattern Recognition for HCI. San Joes State University. Available at: http://www.engr.sjsu.edu/~knapp/HCIRODPR/PR_home.htm [last accessed: August 10th 2003]

Elgammal A.; Duraiswami, R.; Davis, L.S. (2001). Efficient non-parametric adaptive color modeling using fast Gauss transform CVPR 2001 v. 2 pp. II -563-70.

Fieguth, P.; Terzopoulos, D. (June 1997). Color-based tracking of heads and other mobile objects at video frame rates. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 21 –27.

Finlayson, Graham; Hordley, Steven; Hubel, Paul (2002). Illuminant Estimation for Object Recognition Color research and application 27(4), 2002, pp. 260-270.

Funt B. V. and Ginlayson G. D. (1995). Color Constant Color Indexing. IEEE Transaction on Pattern Analysis and Machine Vision 17(5), pp. 522-529.

Gevers, Thoe and Smeulders, Arnold (1999). Color-based object recognition. Pattern Recognition (32), pp. 453-464.

Grimson, W.; Stauffer, C.; Romano, R.; Lee, L. (1998). Using adaptive tracking to classify and monitor activities in a site. CVRP pp. 22-29.

Grove, T.D.; Baker, K.D.; Tan, T.N. (1998). Colour based object tracking Proceedings. Fourteenth International Conference on Pattern Recognition, v. 2 pp. 1442 -1444.

Han, Ju; Ma, Kai-Kuang (2002). Fuzzy color histogram and its use in color image retrieval. IEEE Transactions on Image Processing 11(8), pp. 944-954.

Haritaoglu I.; Harwood, D.; Davis, L.S. (1998). W4: A Real Time System for Detecting and Tracking People. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 962-966.

Huang, J; Kumar, SR; Mitra, M; Zhu, WJ; Zabih, R. (1999) Spatial Color Indexing and Applications. International Journal of Computer Vision. 35 (3), pp. 602-607).

Intille, S.; Davis, J.; Bobick, A. (1997) Real-time closed-world tracking. Proceedings of IEEE CVPR, pp. 697–703.

Kettnaker, Vera and Zabih, Ramin (1999). Bayesian Multi-camera Surveillance. IEEE Computer Society Conference on Computer Vision and Patter Recognition (2), pp. 253-259.

Korhonen, M.; Heikkila, J.; Silvcn, O. (2000). Intensity independent color models and visual tracking. Proceedings. 15th International Conference on Pattern Recognition (3) pp. 600 –604.

Krumm, J.; Harris, S.; Meyers, B.; Brumitt, B.; Hale, M.; Shafer, S. (2000). Multi-camera multi-person tracking for EasyLiving. Proceedings. Third IEEE International Workshop on Visual Surveillance pp. 3 –10.

Larman, Craig (1998) Applying UML and patterns New Jersey: Prentice Hall.

Lee, Yong-Boem; You, Burn-Jae; Lee, Seong-Whan (2001). A real-time color-based object tracking robust to irregular illumination variations. IEEE International Conference on Robotics and Automation ICRA 2001 (2), pp. 1659-1664.

Li, Y.; Hilton, A.; Illingworth, J. (2001). Towards reliable real-time multiview tracking. Proceedings. 2001 IEEE Workshop on Multi-Object Tracking pp. 43 –50.

Low, Colin (2002) Visual Sensing for Large-Scale Tracking. Hewlett Packard Laboratories.

Lu, W. and Tan, Y. (2001). A color histogram based people tracking system The 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001. (2), pp. 137 -140.

Magee, Derek (2000). Machine Vision Techniques for the Evaluation of Animal Behaviour. University of Leeds, PhD Thesis.

Mason, M; Duric, Z. (2001). Using histograms to detect and track objects in color video. Applied Imagery Pattern Recognition Workshop, AIPR 2001. pp. 154 –159.

Matas, J.; Mark, R.; Kittler, J. (1995) On representation and matching of multi-coloured objects. Proceedings of the ICCV'95, Cambridge, MA. pp. 726-732.

McKenna, Stephen J; Raja, Yogesh; Gong Shaogang (Mar. 1999) Tracking colour objects using adaptive mixture models. Image and Vision Computing 17 (3-4), pp. 225-231.

Oestereich, Bernd (1999) Developing Software with UML England: Addison Wisley Longman Ltd.

Orwell, J.; Remagnino, P.; Jones, G.A. (1999). Multi-camera colour tracking. Second IEEE Workshop on Visual Surveillance pp. 14 –21.

Orwell J.; Remagnino P.M.; Jones G. A. (2001). Optimal Color Quantization for Real-Time Object Recognition. Real Time Imaging 7(5) pp. 401-414.

Pattern Recognition Group. Colour Sensitivity. Faculty of Applied Science, Delft University of Technology. Netherlands. available at: http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip-Color.html [last accessed: August 27th 2003]

Paul, G.; Beach, G.; Cohen, C. (2001). A realtime object tracking system using a color camera. Applied Imagery Pattern Recognition Workshop, AIPR 2001 pp. 137 -142.

Raja, Y.; McKenna, S.J.; Shaogang Gong (1998a). Tracking and segmenting people in varying lighting conditions using colour. Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition. pp. 228 –233.

Raja, Y.; McKeann, S.; Gong S. (1998b). Segmentation and tracking using colour mixture models. Asian Conference on Computer Vision, Hong Kong, pp. 607-614.

Roberts, Stephen; Husmeier, Dirk; Rezek, lead; Penny, William (1998). Byesian Approaches to Gaussian Mixture Modelling. IEEE Transactions on Pattern Analysis and Machine Vision 20(11), pp. 1133-1142.

Rothwell, Charles (1995). Object Recognition through Invariant Indexing. Oxford: Oxford Science Publication.

Schwardt, Ludwig (2003). Lecture 6: Gaussian Mixture Models. Universit of Stellenbosch. South Africa. Available at: http://dsp.ee.sun.ac.za/~schwardt/pr813/lectures/lecture06/pr813_lecture06.pdf [last accessed: August 27th 2003]

Scholl, Hans (2001). System Dynamics and Action Research. Univeristy at Albany. An online presentation available at: http://www.albany.edu/cpr/sdgroup/pad824/Action_Research_PAD824.ppt [last accessed: August 10th 2003]

Swain, M. J.; Ballard, D.H (Dec. 1990). Indexing via color histograms. Proceedings, Third International Conference on Computer Vision, 4-7 Dec. 1990. pp. 390-39.

Tsin, Yanghai; Colins, Robert; Ramesh, Visvanathan; Kanade, Takeo (2001). Bayesian color constancy for outdoor Object Recognition. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001 (1), pp. 1132-1139.

Xu, Ming and Ellis, Tim (2001). Illumination-Invariant Motion Detection Using Colour Mixture Models. British Machine Vision Conference, BMVC 2001, pp. 163-172.

Yoo, Jang-Hee; Nixon, Mark S.; Harris, Chris J. (2002) Extracting Human Gait Signatures by Body Segment Properties. Proceedings Proc. IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 35-39.

Bibliography

Abginya, J. and David, R. (1999). Multi-Object Tracking in Video. Real-Time Imaging 5(5), pp. 295-304.

Baumberg, Adam (1995). Learning Deformable Models for Tracking Human Motion, PhD thesis, University of Leeds.

Bishop, Christopher (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press.

Boer, P.; Kroese D.; Mannor, S.; Rubinstein, R. (2003). A tutorial on the Cross-Entropy Method. Available at: http://wwwhome.cs.utwente.nl/~ptdeboer/ce/tutorial [last accessed: August 28th 2003]

Boult, T. E.; Micheals, R.; Gao, X.; Lewis, P.; Power, C.; Yin, W.; Erkan, A. (1999). Frame-Rate Omnidirectional Surveillance and Tracking of Camouflaged and Occluded Targets. Second IEEE Workshop on Visual Surveillance (VS'99), pp. 48-55.

Braude, Eric. (1997) Analyzing and Designing Object-Oriented Systems USA: IEEE Standards Office.

Bressan M.; Guillament D; Vitria J. (2003). Using an ICA representation of local color histograms for object recognition. Pattern Recognition 36(3) pp. 691-701.

Carr, W. and Kemmis, S. (1986) Becoming Critical: Knowing Through Action Research UK: Deaking University

Chang P.; Krumm, J. (1999). Object recognition with color cooccurrence histograms. IEEE Computer Society Conference on. Computer Vision and Pattern Recognition (2), pp.498-504.

Chen TQ; Lu, Y. (2002) Color image segmentation – An innovative approach. Pattern Recognition 35 (2), pp. 395-405.

Cheng HD; Jiang XH; Sun, Y; Wang J. (2001). Color image segmentation: Advances and prospects. Pattern Recognition.34 (12) , pp. 2259-2281.

Cootes, T.F.; Edwards, G.J.; Taylor, C.J. (1998). Active Appearance Models. Proc. European conference on Computer Vision (2), pp. 484-498.

Das, Risemen and Drapper, M. (1997). Focus: Searching for multi-colored objects in a diverse image database. Proceedings of IEE Computer Vision and Pattern Recognition Conference, Puerto Rico, pp. 756-761.

Dempster, A.; Laird, N.; Rubin, D. (1977) Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of the Royal Statistical Society, Series B (39)1, p. 1-38.

Du, Yue; Crisman, J. (1995). A color projection for fast Generic Target Tracking. IEEE/RSJ International Conference on Human Robot Interaction and Cooperative Robots (1), pp. 360-365.

Duda, Richard (1973). Pattern Classification and Scene Analysis. New York: Wiley-Interscience.

Duda, Richard (1997). Pattern Recognition for HCI. San Joes State University. Available at: http://www.engr.sjsu.edu/~knapp/HCIRODPR/PR_home.htm [last accessed: August 10th 2003]

Elgammal A.; Duraiswami, R.; Davis, L.S. (2001). Efficient non-parametric adaptive color modeling using fast Gauss transform. CVPR 2001 (2), pp. II -563-70.

Faugeras, Olivier (1993). Three-Dimensional Computer Vision: a geometric viewpoint. Cambridge: MIT Print.

Fieguth, P.; Terzopoulos, D. (June 1997). Color-based tracking of heads and other mobile objects at video frame rates. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 21 –27.

Finlayson, Graham; Hordley, Steven; Hubel, Paul (2002). Illuminant Estimation for Object Recognition Color research and application 27(4), 2002, pp. 260-270.

Funt B. V. and Ginlayson G. D. (1995). Color Constant Color Indexing. IEEE Transaction on Pattern Analysis and Machine Vision 17(5), pp. 522-529.

Gevers, Thoe and Smeulders, Arnold (1999). Color-based object recognition. Pattern Recognition (32), pp. 453-464.

Grimson, W.; Stauffer, C.; Romano, R.; Lee, L. (1998). Using adaptive tracking to classify and monitor activities in a site. CVRP, pp. 22-29.

Grove, T.D.; Baker, K.D.; Tan, T.N. (1998). Colour based object tracking Proceedings. Fourteenth International Conference on Pattern Recognition, v. 2 pp. 1442 -1444.

Guillamet, D.; Vitria, J. (2000). A comparison of global versus local color histograms for object recognition. Proceedings. 15th International Conference on Pattern Recognition (2), pp. 422 -425.

Han, Ju; Ma, Kai-Kuang (2002). Fuzzy color histogram and its use in color image retrieval. IEEE Transactions on Image Processing 11(8), pp. 944-954.

Haritaoglu I.; Harwood, D.; Davis, L.S. (1998). W4: A Real Time System for Detecting and Tracking People. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 962-966.

Heap, Anthony James (1997). Learning Deformable Shape Models for Object Tracking. PhD Thesis, University of Leeds.

Huang, J; Kumar, SR; Mitra, M; Zhu, WJ; Zabih, R. (1999) Spatial Color Indexing and Applications. International Journal of Computer Vision. 35 (3), pp. 602-607).

Intille, S.; Davis, J.; Bobick, A. (1997) Real-time closed-world tracking. Proceedings of IEEE CVPR, pp. 697–703.

Kettnaker, Vera and Zabih, Ramin (1999). Bayesian Multi-camera Surveillance. IEEE Computer Society Conference on Computer Vision and Patter Recognition (2), pp. 253-259.

Korhonen, M.; Heikkila, J.; Silvcn, O. (2000). Intensity independent color models and visual tracking. Proceedings. 15th International Conference on Pattern Recognition (3), pp. 600 –604.

Krumm, J.; Harris, S.; Meyers, B.; Brumitt, B.; Hale, M.; Shafer, S. (2000). Multi-camera multi-person tracking for EasyLiving. Proceedings. Third IEEE International Workshop on Visual Surveillance pp. 3 –10.

Larman, Craig (1998) Applying UML and patterns New Jersey: Prentice Hall.

Lee, Yong-Boem; You, Burn-Jae; Lee, Seong-Whan (2001). A real-time color-based object tracking robust to irregular illumination variations. IEEE International Conference on Robotics and Automation ICRA 2001 (2), pp. 1659-1664.

Li, Y.; Hilton, A.; Illingworth, J. (2001). Towards reliable real-time multiview tracking. Proceedings. 2001 IEEE Workshop on Multi-Object Tracking pp. 43 –50.

Low, Colin (2002) Visual Sensing for Large-Scale Tracking. Hewlett Packard Laboratories.

Lu, W. and Tan, Y. (2001). A color histogram based people tracking system The 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001 (2), pp. 137 -140.

MacCormick, John (2002). Stochastic algorithms for visual tracking : probabilistic modelling and stochastic algorithms for visual localisation and tracking. London: Springer.

Magee, Derek (2000). Machine Vision Techniques for the Evaluation of Animal Behaviour. University of Leeds, PhD Thesis.

Magee, Derek (2003). Tracking Multiple Vehicles using Foreground, Background and Motion Models. Image and Vision Computing (to appear).

Mason, M; Duric, Z. (2001). Using histograms to detect and track objects in color video. Applied Imagery Pattern Recognition Workshop, AIPR 2001. pp. 154 –159.

Matas, J.; Mark, R.; Kittler, J. (1995) On representation and matching of multi-coloured objects. Proceedings of the ICCV'95, Cambridge, MA. pp. 726-732.

McKenna, Stephen J; Raja, Yogesh; Gong Shaogang (Mar. 1999) Tracking colour objects using adaptive mixture models . Image and Vision Computing 17 (3-4), pp. 225-231.

McLachlan G. and Krishnan T. (1996) The EM algorithm and Extensions. New York: John Wiley & Sons.

Oestereich, Bernd (1999) Developing Software with UML England: Addison Wisley Longman Ltd.

Orwell, J.; Remagnino, P.; Jones, G.A. (1999). Multi-camera colour tracking. Second IEEE Workshop on Visual Surveillance pp. 14 –21.

Orwell J.; Remagnino P.M.; Jones G. A. (2001). Optimal Color Quantization for Real-Time Object Recognition. Real Time Imaging 7(5) pp. 401-414.

Pattern Recognition Group. Colour Sensitivity. Faculty of Applied Science, Delft University of Technology. Netherlands. available at: http://www.ph.tn.tudelft.nl/Courses/FIP/frames/fip-Color.html [last accessed: August 27th 2003]

Paul, G.; Beach, G.; Cohen, C. (2001). A realtime object tracking system using a color camera. Applied Imagery Pattern Recognition Workshop, AIPR 2001 pp. 137 -142.

Raja, Y.; McKenna, S.J.; Shaogang Gong (1998a). Tracking and segmenting people in varying lighting conditions using colour. Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition. pp. 228 –233.

Raja, Y.; McKeann, S.; Gong S. (1998b). Segmentation and tracking using colour mixture models. Asian Conference on Computer Vision, Hong Kong, pp. 607-614.

Roberts, Stephen; Husmeier, Dirk; Rezek, lead; Penny, William (1998). Byesian Approaches to Gaussian Mixture Modelling. IEEE Transactions on Pattern Analysis and Machine Vision 20(11), pp. 1133-1142.

Rothwell, Charles (1995). Object Recognition through Invariant Indexing. Oxford: Oxford Science Publication.

Schwardt, Ludwig (2003). Lecture 6: Gaussian Mixture Models. Universit of Stellenbosch. South Africa. Available at: http://dsp.ee.sun.ac.za/~schwardt/pr813/lectures/lecture06/pr813_lecture06.pdf [last accessed: August 27th 2003]

Scholl, Hans (2001). System Dynamics and Action Research. Univeristy at Albany. An online presentation available at: http://www.albany.edu/cpr/sdgroup/pad824/Action_Research_PAD824.ppt [last accessed: August 10th 2003]

Sticker, M.; Swain, M. (1994). The capacity of color histogram indexing. Computer Vision and Pattern Recognition, pp. 704-708.

Swain, M. J.; Ballard, D.H (Dec. 1990). Indexing via color histograms. Proceedings, Third International Conference on Computer Vision, 4-7 Dec. 1990. pp. 390-39.

Tsin, Yanghai; Colins, Robert; Ramesh, Visvanathan; Kanade, Takeo (2001). Bayesian color constancy for outdoor Object Recognition. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001 (1), pp. 1132-1139.

Wren, C.R.; Azarbayejani, A; Darrell, T.; Pentland, A. (1997) Pfinder: real-time tracking of the human body, Adaptive background mixture models for real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), pp. 780-785.

Xu, Ming and Ellis, Tim (2001). Illumination-Invariant Motion Detection Using Colour Mixture Models. British Machine Vision Conference, BMVC 2001, pp. 163-172.

Yoo, Jang-Hee; Nixon, Mark S.; Harris, Chris J. (2002) Extracting Human Gait Signatures by Body Segment Properties. Proceedings Proc. IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 35-39.

Appendix A – Lessons Learnt

When the project was proposed, it was a completely new field of research to me. Though the computer vision research was introduced during the module 'Perceptual Systems', yet the basics were new and required a lot of reading to cope up with the level of research. These and other challenges faced during the course of the project were summarized in section 6.3. The major lesson that was learnt from pursuing this project is that one should try and challenge oneself in new fields. This will keep the person motivated all through the three months duration of the project.

The project also changed my understanding of implementation and the usage of programming. The requirements of libraries that are to be used for research are completely different than application-oriented libraries. Research libraries should facilitate the different choices and its re-usability issues extend beyond re-using the class. Re-using the algorithms, codes, or even ideas from such libraries should be taken into consideration.

The project required a wide collection of design and implementation decisions. The sample choice and the choices of charts to represent the findings were the most difficult design decisions. This is because I could not find any guidance from previous researches on how to make these decisions. The discussions with my supervisor were the leading help in taking these decisions. I had a difficult time trying to decide on ways to represent the findings of the experiments and as a result what charts should be more appropriate to represent the overall picture of the findings.

As the project's context was new, I had to revise a large number of articles to catch up with the latest news in the topic and evaluate the current status of research in the area. Yet, what I also learnt is that it is not only the topic's articles that are of relevance, but also a wide variety of articles from other areas. Similar areas might introduce ideas that are of value. In addition, other areas like the psychology of colours and statistical books were really important for the project. The project required combining different skills to satisfy the purpose. Mathematical and statistical skills, analytical skills, along with programming and computer vision understanding were all important for the achievements of the project.

One additional challenge that I had to face, and I learnt lots from, is trying to understand an already written, and barely documented, code for changing it or utilizing its algorithms. This required lots of patience and concentration to keep following the flow of the algorithms and procedures.

One unintentional error in the plan proved to be very useful. The plan was based on weekly basis. Each month actually contained two to three days over the four weeks. These extra days proved to be very useful in catching up with the plan. I thus advice future students to follow the weekly schedule or leave some days for emergencies and follow-ups.

The project log, which was explained in detail in section 6.2, was of real and clear value. It is a very good idea to keep track of the project's progress. In addition, it combines all the notes, schedules and readings in one place so the researcher does not waste time searching for papers or files. One extra suggestion that I did not do but I think is worth pursuing is to enable the log to have links to all the files relevant to the project. This will make the file management task easier.

A relevant problem was related to space management. Although my overall quota was increased, to cope with the fact that the project includes lots of movies and text files related to signatures, the quota was never enough. I had to put everything on CDs and keep my quota for the current working phase only. Space management and keeping track of where all the files were stored posed a problem. It would be good to keep an electronic archive of all the files, especially for research-oriented projects like this one.

On the personal scale, the project has taught me a lot of essential skills. My abilities to express my ideas out and talk them through with other researchers and experts have undoubtedly been enhanced. Through my meetings with my supervisor, HP and the computer vision group researchers, I developed an ability to explain my academic thoughts and answer questions about my ideas. I learnt when to defend my ideas and how to value other people's ideas and use them for the benefit of the research.

Moreover, the project gave me the chance to write my first academic paper under the supervision of my supervisor. Writing academic papers requires lots of skills like concise writing, summarising and decisions justification.

Research-oriented projects, where the goal is clear but the paths to satisfy the goal is up to the researcher, require a considerable amount of independence and self-discipline. I had to be really motivated and interested about the topic and the possible findings. My resulting advice then to future students is: do not take a project unless you are really interested and motivated to do it efficiently.

Though everything seems to be good and I learnt a lot of things, I doubt that my project management was efficient. I eventually resulted to devote the majority of my time working on my project. Revising my schedule, which is available in my project log, I realized that I even used the weekends to work. This could be acceptable for this project as I was really challenged and wanted to utilize all the time possible. Yet, I should take extra care in future work to make a better balance between my work and my social life. This was a balance that I clearly missed during the course of this project, due to the feelings of joy I had working on it.

Appendix B – Objectives Form

School of Computing, University of Leeds

MSc PROJECT OBJECTIVES AND DELIVERABLES

This form must be completed by the student, with the agreement of the supervisor of each project, and submitted to the MSc project co-ordinator (Mrs A. Roberts) via CSO by 27th March 2003. A copy should be given to the supervisor and a copy retained by the student. Amendments to the agreed objectives and deliverables may be made by agreement between the student and the supervisor during the project. Any such revision should be noted on this form. At the end of the project, a copy of this form must be included in the Project Report as an Appendix.

Student: _____Dima Jamal Al Damen______________

Programme of Study: ____MSc in Distributed Multimedia Systems__

Supervisor: _____Dr. Andy Bulpitt____________________

Title of project: __Visual Signature for Large-Scale Tracking

____

External Organisation*: ______Hewlett Packard Laboratories______

* (if applicable)

AGREED MARKING SCHEME

Understand the Problem	Produce a Solution *	Evaluation	Write-Up	Appendix A	TOTAL %
20	40	20	15	5	100

* This category includes Professionalism

OVERALL OBJECTIVES:

Based on the paper entitled "Visual Sensing for Large-Scale Tracking", HP laboratories are trying to track many thousands of people with a large scale network of cameras. The system architecture presented in the paper is based on the assumption that a "visual signature" can be generated for each person in the scene. In his paper, Colin Lowe defines the visual signature as "a finite bitstream of some kind.. a unique signature for each unique object, and each unique object results in the same unique signature being generated by each camera" (p. 2).

This project tries to study the possibilities of producing the "visual signature", the techniques and difficulties that might hinder such finding. The project will start by reviewing the literature in the field. It is known that such a signature has not yet been produced; yet some researchers proposed techniques that they think might help in this finding. The project will then try to propose a solution. The solution will be tested and evaluated as a step towards further search for the "visual signature".

The project aims at learning the process of reviewing current status, proposing a solution, testing the successes and failures of the proposed solution and evaluating this solution. It is clearly understood that the visual signature might not be feasible under known technologies, thus the project aims at providing another step further in this area of research.

MINIMUM REQUIRMENTS:

3. Clear understanding of the current state-of-art in real-time tracking
4. Understanding of the current systems available in the School of Computing to help in producing such a system.
5. Proposing one solution which might help in producing the "visual signature"

SOFTWARE AND HARDWARE RESOURCES REQUIRED:

1. Digital Camera and capturing tools
2. Continuous access to UNIX machines.

DELIVERABLE(s):

1. Project Report
2. A working demonstration of the success/failure of the proposed mechanism. (Due to the known and unknown difficulties, the possibility of producing a partially reliable visual signature cannot be placed as minimum requirements. Yet, it is the goal towards which the project will be going).

Signature of student: Dima Al Damen Date: 10/4/2003

Signature of supervisor: Date:

Amendments to agreed objectives and deliverables:

Date Amendment

Appendix C – Interim Report

School of Computing, University of Leeds

MSc PROJECT INTERIM REPORT

All MSc students must submit an interim report on their project to the MSc project co-ordinator (Mrs A. Roberts) via CSO ***by Thursday 8th May 2003.*** Note that it may require two or three iterations to agree a suitable report with your supervisor, so you should let him/her have an initial draft well in advance of the deadline. The report should be a maximum of 10 pages long and be attached to this header sheet. It should include:

- the objectives, deliverables and agreed marking scheme
- resources required
- progress report and project schedule
- proposed research methods
- a draft chapter on the literature review and/or an evaluation of tools/techniques
- the WWW document link for the project log to date

The report will be commented upon both by the supervisor and the assessor in order to provide you with feedback on your approach and progress so far.

The submission of this Interim Report is a pre-requisite for proceeding to the main phase of the project.

Student:	Dima J. Al Damen
Programme of Study:	MSc in Distributed Multimedia Systems
Title of project:	Visual Signature for Large-Scale Tracking
Supervisor:	Dr. Andy Bulpitt
External Company (if appropriate):	Hewlett Packard Laboratories

AGREED MARKING SCHEME

Understand the problem	Produce a solution *	Evaluation	Write-up	Appendix A	TOTAL %
20	40	20	15	5	100

*** This includes professionalism**

Signature of student: Dima Al Damen **Date:** 8/5/2003

Supervisor's and Assessor's comments overleaf è

Supervisor's comments on the Interim Report

Progress to date has been excellent. A thorough review of relevant literature has been performed.
Since writing the report a meeting with HP has defined the initial stages of the project to look at Active Appearance Models - further reading in the area is now required.
You should how you will evaluate the work when the implementation is complete.

Assessor's comments on the Interim Report

Very good background research. Literature review should probably have fewer direct quotes from sources.

Remember that you will need to evaluate

- Your solution (by performing suitable quantitative experiments)
- The process by which you designed and developed the solution.

Appendix D – IEEE Submission

Appendix E – Library Design and Documentation

E.1 Library Design

The design of the library took place in 4 consecutive phases.

E.1.1 Design of phase I – Colour Histograms

Several design stages are required to generate the visual signature using colour histogram. These design stages are:

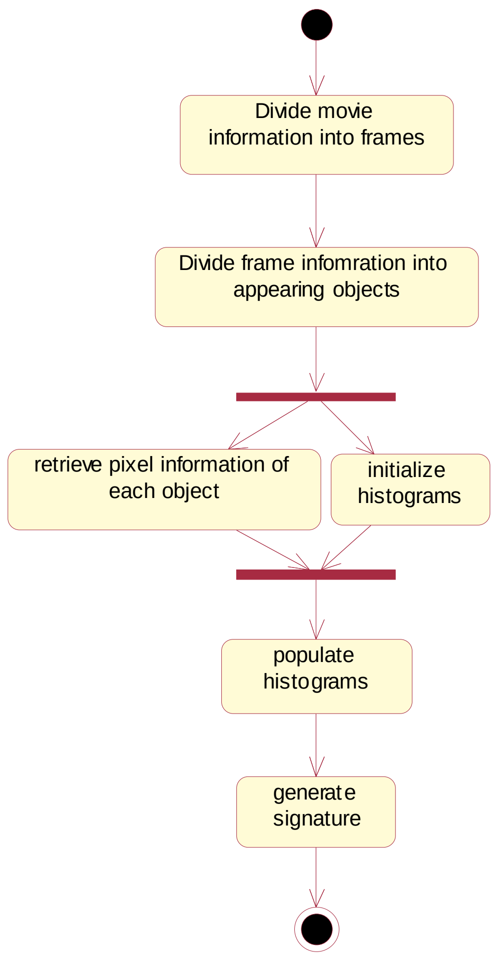

Figure E. 1 Activity Diagram – Movie Analysis

Based on the activities declared in figure E.1, the class diagram of figure E.2 is created. The class diagram can be read as follows: Each movie consists of a set of frames. The frame could contain one or more objects. Each object represents one person using the signature information. Object information is represented by a set of pixels. These pixels are grouped using a histogram. Histogram class is an abstract class containing the basic functionality of histograms. RGB, HS, NRG (Normalized RG), CIE and Normalized CIE histograms inherit the functionality of the class. This enables the addition of other colour spaces for later researches. Histogram class has the intersection method that compares the current histogram (model histogram) with another histogram of the same type.

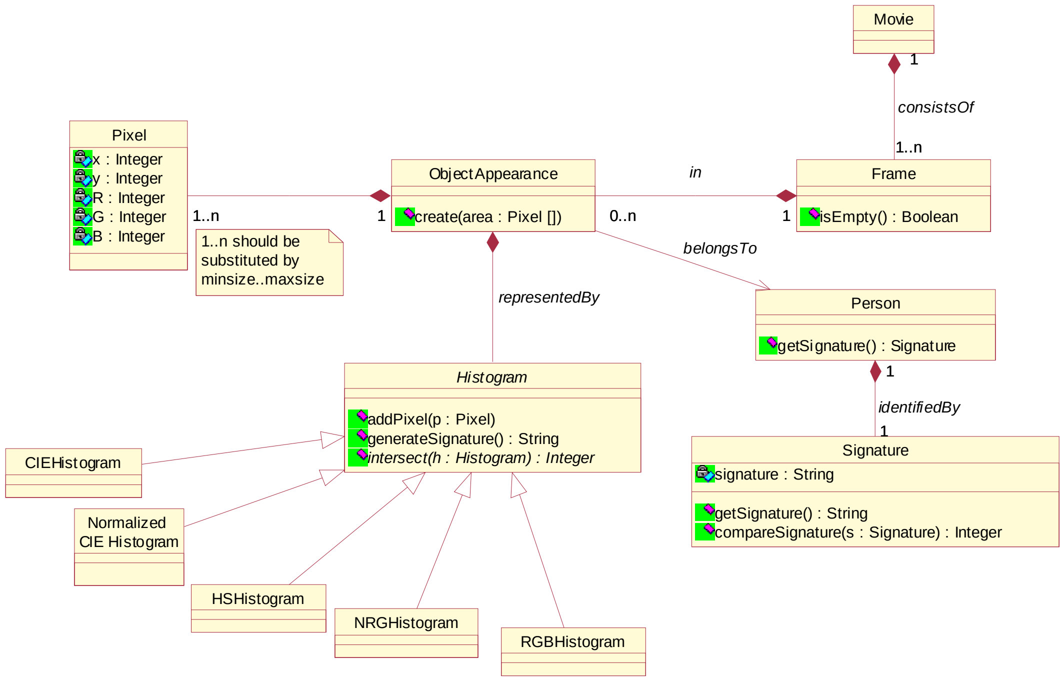

Figure E. 2 Class Diagram - Phase I

The main driver program called 'VisualSignature.java' inputs the initial information and processes the movie files passed to the program as arguments. It generates the resulting signatures of the objects in the movie. It was decided to write the signatures into files so that they can be analysed later in different ways. Each movie will have a directory containing all the signatures from different frames belonging to objects of the movie. The signature of each histogram colour is placed in a separate file.

Two driver programs were implemented for this experiment. The first program is called 'CompareSignaturesSamePerson.java'. It aims at comparing different frames belonging to the same movie for the purpose of generating the matching values. The second program, called 'CompareSignaturesDiffPerson.java', compares frames from two different directories.

E.1.2 Design of phase II – Three regions scheme colour histogram

The three regions scheme introduces a new and important class into the library, the 'Region' class. Each object will be divided into exactly three regions. The pixels of the object are then classified into the three regions before a histogram is created for each region.

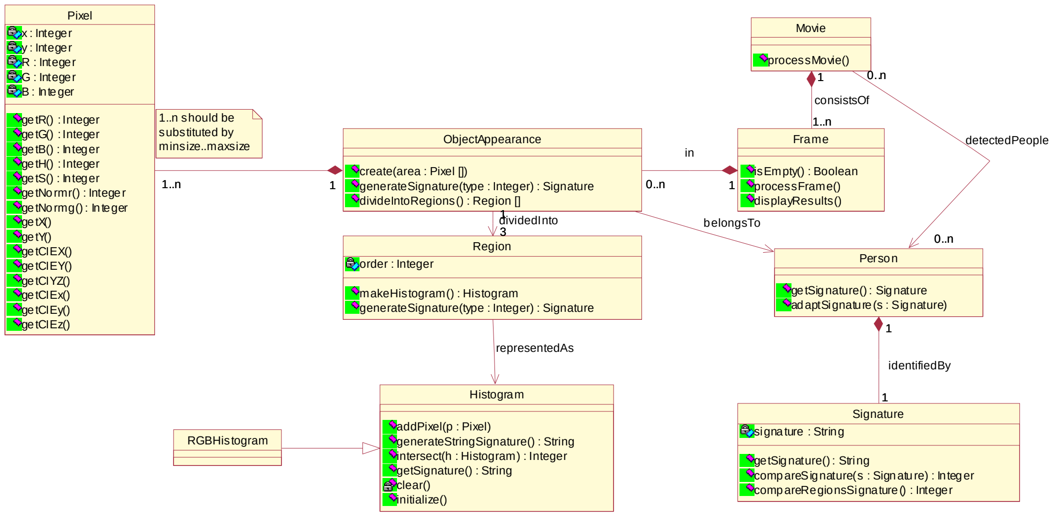

Figure E. 3 Class Diagram – Phase II

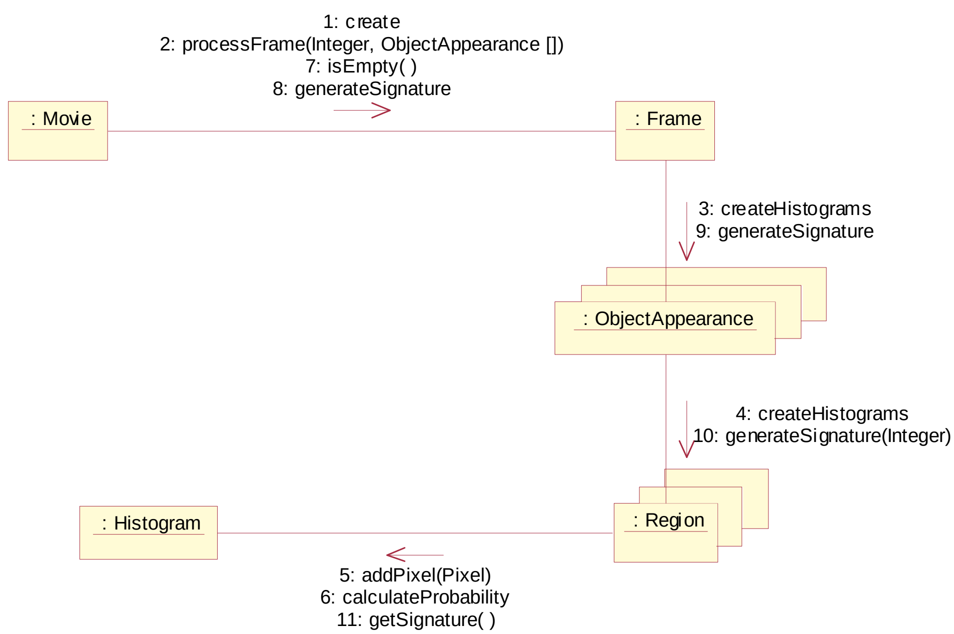

Figure E. 4 Collaboration Diagram – Creating Regions' Histograms

The signature would then be a collection of the three regions' signatures. It was decided to separate the signatures of the regions by semicolon (;) in the resulting file. So the resulting signature file would be:

region1Signature; region2Signature; region3Signature

A new method is added to compare the signature file of the above format. The method would extract the histogram information of each region in order then compare the corresponding region's histograms using the histogram intersection technique. Two new driver programs entitled 'CompareRegionsSignatureSamePerson' and 'CompareRegionsSignatureDiffPerson' are used for comparing the new format of the signature. The corresponding regions are compared for matching and the resulting match number is outputted to a file. The file has the format:

frame1No frame2No region1Match region2Match region3Match

The new class diagram was implemented into the library. Only RGB signatures were compared. This is because RGB colour histogram proved to give the best results for both similar and different pedestrians.

E.1.3 Design of phase III – The three regions GMM

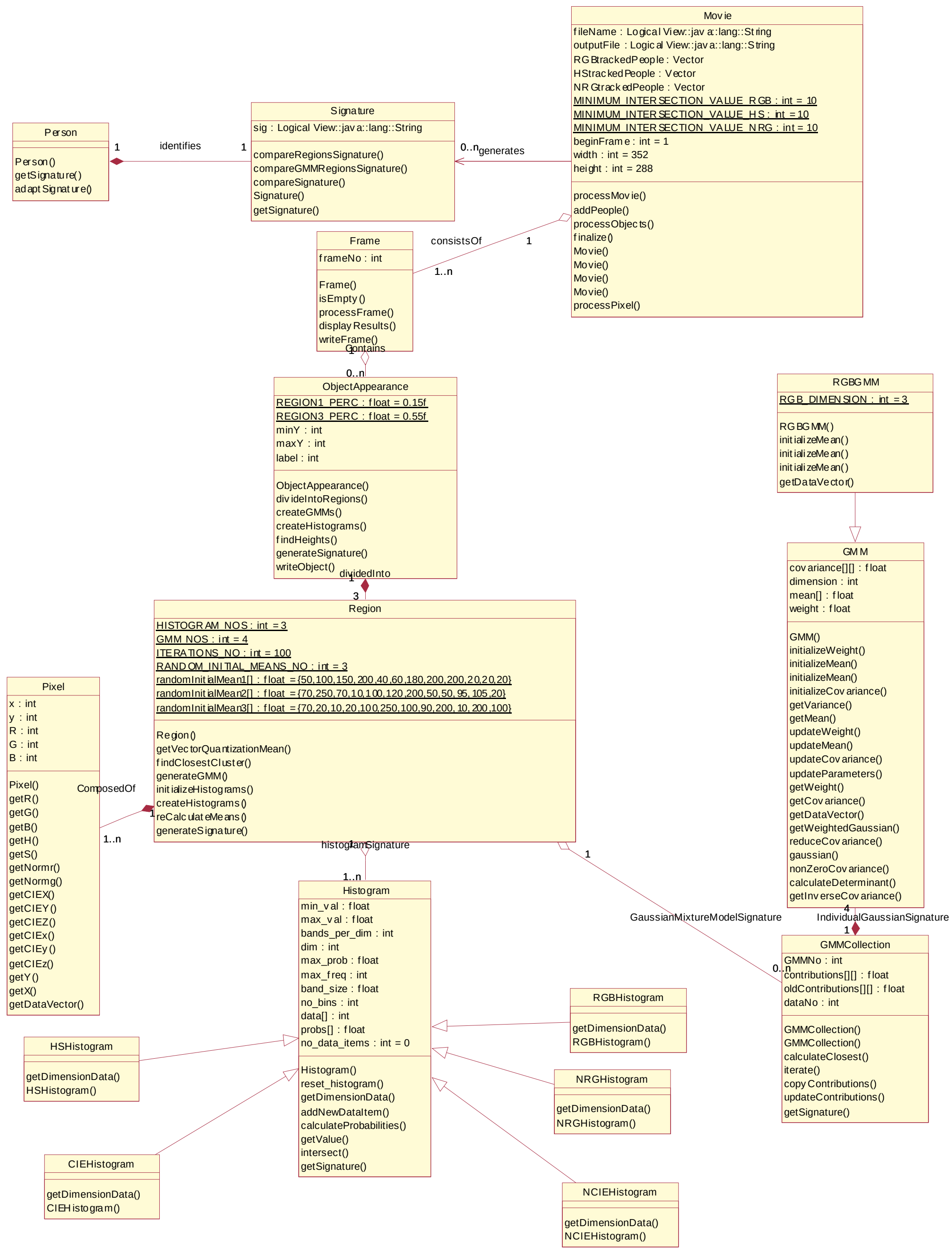

Figure E. 5 Class Diagram – Phase III

The GMM, and GMMCollection classes were added as figure E.5 shows. GMM class represents the information of individual Gaussians. GMMCollection represents the whole mixture model. The full methods of these classes can be reviewed in the library documentation in appendix E.2.

GMM and GMMCollection classes were designed for general re-usability. It is expected to use any of those two classes in other kinds of researches. Thus, the author gave extra care in developing these classes. The classes should represent the basic functionality of GMMs whether they are used for colour spaces or for any other kind of data point representation.

The iterations for creating and stabilizing the GMMCollection can be viewed in figures E.6 and E.7 below:

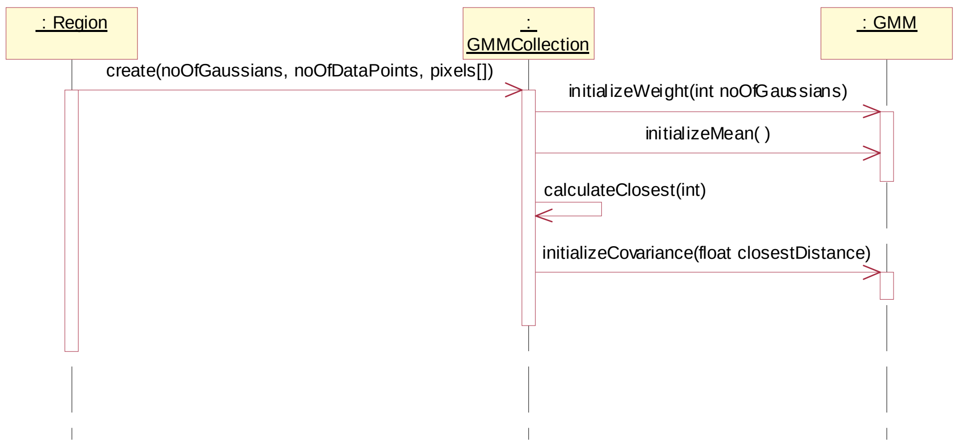

Figure E. 6 Sequence Diagram - Initialising GMM

Figure E.7 shows how each iteration works in the process to stabilize GMMs. ‘Contributions’ simply means the probability that a pixel belongs to the Gaussian. This is denoted by $p(x|j)$ in the equations in section 2.4.

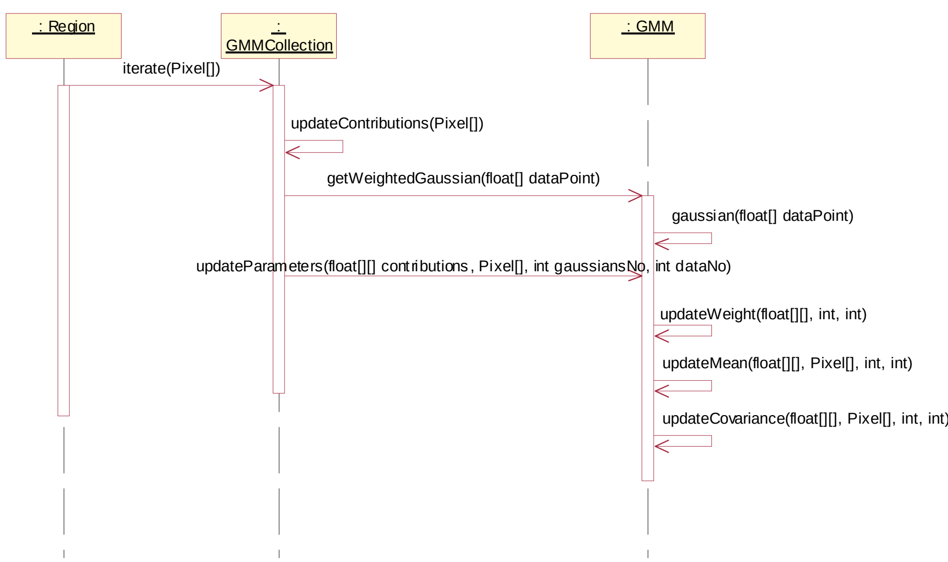

Figure E. 7 Sequence Diagram - Iterating through the EM Algorithm

The GMMs require a change in the structure of the signatures introduced. This is because the signature now consists only of the parameters of the Gaussian mixture model. The signature would have the following structure.

```
weight1, mean1R, mean1G, mean1B, covariance1elements+
weight2, mean2R, mean2G, mean2B, covaraince2elements+
weight3, mean3R, mean3G, mean3B, covariance3elements+
weight4, mean4R, mena4G, mean4B, covaraince4elements;
weight1, mean1R, …..
```

The plus sign '+' was used to separate the parameters of the four Gaussians. Any sign could substitute this sign. The dash '-' was used in the beginning but it was later changed as the covariance matrix parameters could be negative which results in ambiguity. The semicolon ';' is still used to separate the signatures of each region. An example of the GMM signature is:

```
0.09780906,52.641712,46.219025,46.264217,58.351364,14.967203,7.28569
9,14.967201,18.826889,8.46818,7.285699,8.46818,5.7950583,+0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,+0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,+0.3781268,44.14058,41.530937,41.307114,25.391659,13.011697,
10.064128,13.011697,15.196275,9.943146,10.064128,9.943146,9.651836,+;
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,+0.056652024,88.234886,9
3.25307,103.207825,349.04523,89.04939,-
8.6659355,89.04939,35.67506,1.3166641,-
8.6659355,1.3166641,5.7539864,+ ….
```

Two new driver programs were added 'CompareGMMRegionsSamePerson' and 'CompareGMMRegionsDiffPerson' for comparison.

E.2 Library Documentation

The following documentation was automatically generated by the JavaDoc utility of the JDK. The classes were ordered alphabetically:

1. Class CIEHistogram

public class CIEHistogram
extends Histogram
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003
Description: NRG histogram inherits the functionality of the abstract class Histogram

---

Constructor Summary
CIEHistogram() The number of bins is 25*25 (i.e.

Method Summary	
float	getDimensionData(Pixel p, int dim) returns the 2D info from the pixel first dimension is the normalized R information second dimension is the normalized G information

Methods inherited from class Histogram
addNewDataItem, calculateProbabilities, getSignature, getValue, intersect

2. Class FloatVector

public class FloatVector
extends java.lang.Object
Dima J Al Damen University of Leeds First Written: July 19th 2003 Last updated: July 25th 2003
Description: A utility class to manage the float vector The class contains functions for subtracting, calculating the length of the vector and generating a covariance matrix out of the multiplication of the vector by its transpose.

Constructor Summary
FloatVector()

Method Summary	
static float[][]	calculateCovariance(float[] f) returns a matrix of size [f.length][f.length] that's the result of multiplying the vector (array) by its transpose Input: f - input vector Output: float [] [] - resulting covariance matrix
static float	calculateEucledian(float[] v) returns a float value of the length of the vector the method works for any matrix length Input: v - the input vector float - the resulting length value
static float[]	subtractVector(float[] v1, float[] v2) returns a float array that's the subtraction of two vectors the method takes two equal value vectors Input: v1 - first vector v2 - second vector Output: float [] - the resulting vector

3. Class Frame

public class Frame
extends java.lang.Object

Constructor Summary
Frame(int frameNo, ObjectAppearance[] objs) number of the frame, sequential number in the containing movie.

Method Summary	
void	displayResults() Displays frame information onto the screen
boolean	isEmpty()

	Input: none Output: boolean: whether the frame has any objects or not
void	processFrame() Input: none Output: none loops aroung the objects in the frame creating a histogram for each
java.awt.image.BufferedImage	writeFrame(int width, int height) Writes the frame information into an image file

4. Class GMM

Direct Known Subclasses:
RGBGMM

public abstract class GMM
extends java.lang.Object
Dima J Al Damen University of Leeds First Written: July 17th 2003 Last updated: July 25th 2003 Description: Gaussian class that represents the information of one Gaussian in a GMM collection Credit: Some of the code implemented here is based on the EM code written by Dr Derek Magee Dr Magee allowed this code to be utilized for the purpose of the project The EMFitD.cc class in the library owned by the computer vision group - University of Leeds was used.

Field Summary	
protected float[][]	covariance the covariance matrix of the Gaussian
protected int	dimension the dimension of the data represented by the Gaussian
protected float[]	mean the mean of the Gaussian
protected float	weight the weight of this Gaussian in the GMM

Constructor Summary
GMM(int dim) Constructor of the gussain class - contains required initialisations Input: dim - dimension of the data represented by the Gaussian

Method Summary	
float[][]	getCovariance() returns the covariance matrix of the Gaussian
float[]	getMean()

	returns the mean of the Gaussian
float	getWeight() returns the weight of the Gaussian
float	getWeightedGaussian(float[] f) Input: f - the variance of the data point from the mean of the Gaussian Output: float - the value of the weighted Gaussian Description: returns the value of the G(input point) multiplied by the weight of that Gaussian in the GMM Equation: wj*G(value[i]: Meanj: Covariancej)
void	initializeCovariance(float value) Input: value - the value to initialize the covariance matrix - this is based on the article by McKenna et al (1999), Image and Vision Computing 17 (1999), pp 225-231 entitled "Tracking colour objects using adaptive mixture models" The article states: "the covariance matrices were initialised to xI, where x was the Euclidean distance from the component's mean to its nearest neighbouring component's mean" - I of course represents the Identity matrix of the size [dimension][dimension]
abstract void	initializeMean() abstract class that depends on the represented colour space
abstract void	initializeMean(int i, float[] f) abstract class depends on the represented colour space
void	initializeWeight(int number) Input: number - number of Gaussians considered in this GMM Description: this class initializes the weight of the Gaussian to 1/number of Gaussians
void	updateParameters(float[][] contributions, Pixel[] data, int gaussianNo, int total) Input: contributions - a matrix that contains the contributions of this GMM data - collection of data items to be represented by the Gaussian guassianNo - the order of the Gaussian in the GMM total - the total number of data items Description - updates the parameters of the guassian

5. Class GMMCollection

public class **GMMCollection**
extends java.lang.Object
Dima J Al Damen University of Leeds First Written: July 19th 2003 Last updated: July 25th 2003
Description: GMMCollection class for the collection of Gaussians to represent this mixture model

Constructor Summary
GMMCollection(int number, int dataNo) Input: number - the number of Gaussians in the GMM dataNo - the number of data items to be considered for the GMM Description: constructor to initialize the GMM; the weight, mean and covariance matrix are initialized The mean is initialized to random variables The weight is initialized to 1/GMMNo The covariance matrix is initialized to xI where x is the "nearest neighbouring component's mean" This is based on the article by McKenna et al (1999)
GMMCollection(int number, int dataNo, float[] initMean)

Input: number - the number of Gaussians in the GMM dataNo - the number of data items to be considered for the GMM initMean - the set of initial means for the Gaussian Description: this constructor calls the default constructor and initializes the Gaussian menas to the ones provided in a long array; the length of the array should be GMMNo*dimension

Method Summary	
java.lang.String	getSignature() Output: String - returns the signature information of the GMMCollection Format: The signature carries all the parameters of the Gaussians in the format of: weight, mean1, mean2, mean3 until mean(dim), cov11, cov12 until cov(dim)(dim) The Gaussians are separated by the symbol "+" Description: returns the signature of the GMM in the above format
void	iterate(Pixel[] data) Input: data - the set of data items to be considered for the GMM Description: iterates the GMMCollection to reach conversion
void	updateContributions(Pixel[] data) Input: data - the set of data items to be considered for the GMM Description: calculates the contribution of each sample (i) to each Gaussian (j) P(i,j) = w(j) * G(X, M, S) ------------------------------- sum[j=0 to J-1](w(j) * G(X, M, S))

6. Class Histogram

Direct Known Subclasses:

CIEHistogram, HSHistogram, NCIEHistogram, NRGHistogram, RGBHistogram

---

public abstract class **Histogram**
extends java.lang.Object
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003 Description: Histogram class calculates histogram information for the collection of pixels given The class The class also writes the frame information into an image Credit: Some of the code implemented here is a direct match from the C++ code implemented by Dr Derek Magee Dr Magee allowed this code to be utilized for the purpose of the project The Histogram.cpp class in the library owned by the computer vision group - University of Leeds was used.

Constructor Summary
Histogram(float minimum_data_value, float maximum_data_value, int bands_per_dimension, int dimensionality)   Input: minimum_data_value : the minimum value of the range of each dimension maximum_data_value : the maximum value of the range of each dimension bands_per_dimension : the number of bands in each dimension (have to be equal for all dimensions) dimensionality: the dimensions' number

Method Summary

void	addNewDataItem(Pixel p) Input: p: pixel containing the information Adds the pixel information onto the histogram frequency
void	calculateProbabilities() makes the histogram scale independent by dividing by the total number of bins
abstract float	getDimensionData(Pixel p, int dim) specific for each histogram type
java.lang.String	getSignature() returns a string representing the probabilities in the histogram this function is expected to be called only after calculateProbabilities() function has been called
float	getValue(int i) returns the value in the probability
float	intersect(Histogram h) returns an intersection of two histograms not implemented correctly yet (not being used actually)

7. Class HSHistogram

public class HSHistogram
extends Histogram
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003
Description: HSHistogram class inherits Histogram class and is specific to 2D HS colour spaces

Constructor Summary
HSHistogram() Initialized HS histogram Currently 25*25 bins are considered within the range of 0-1 for H and S 2Dimensions each divided into 25 equal spaces

Method Summary	
float	getDimensionData(Pixel p, int dim) Input: p : the pixel of which the information is to be extracted dim: the dimension considered.

Methods inherited from class Histogram
addNewDataItem, calculateProbabilities, getSignature, getValue, intersect

8. Class Movie

public class Movie
extends java.lang.Object

Constructor Summary

Movie(java.lang.String fileName, java.lang.String outputFileName, java.util.Vector RGBtrackedPeople, java.util.Vector HStrackedPeople, java.util.Vector NRGtrackedPeople) Input: fileName: name of input file supporting the above format outputFileName: name of the output file supporting the above format Description: Initialization
Movie(java.lang.String fileName, java.lang.String outputFileName, java.util.Vector RGBtrackedPeople, java.util.Vector HStrackedPeople, java.util.Vector NRGtrackedPeople, int beginFrame) Input: fileName: name of input file supporting the above format outputFileName: name of the output file supporting the above format RGBtrackedPeople: vector of current people being tracked using the RGB scheme HStrackedPeople: vector of current people being tracked using the HS scheme NRGtrackedPeople: vector of current people being tracked using the NRG scheme beginFrame: the beginning number of frame to be considered in analysis; this is added so I can run specific bits of the movie for the analysis Description: initialization
Movie(java.lang.String fileName, java.lang.String outputFileName, java.util.Vector RGBtrackedPeople, java.util.Vector HStrackedPeople, java.util.Vector NRGtrackedPeople, int width, int height) Input: fileName: name of input file supporting the above format outputFileName: name of the output file supporting the above format RGBtrackedPeople: vector of current people being tracked us1 ing the RGB scheme HStrackedPeople: vector of current people being tracked using the HS scheme NRGtrackedPeople: vector of current people being tracked using the NRG scheme width: the width of the frames to be written into images height: the height of the frames to be written into images Description: initialization
Movie(java.lang.String fileName, java.lang.String outputFileName, java.util.Vector RGBtrackedPeople, java.util.Vector HStrackedPeople, java.util.Vector NRGtrackedPeople, int width, int height, int beginFrame) Input: fileName: name of input file supporting the above format outputFileName: name of the output file supporting the above format RGBtrackedPeople: vector of current people being tracked using the RGB scheme HStrackedPeople: vector of current people being tracked using the HS scheme NRGtrackedPeople: vector of current people being tracked using the NRG scheme width: the width of the frames to be written into images height: the height of the frames to be written into images beginFrame: the beginning number of frame to be considered in analysis; this is added so I can run specific bits of the movie for the analysis Description: initialization

Method Summary

protected void	finalize() to ensure the output file is closed correctly
void	processMovie() Input: none Output: none throws an exception for errors in reading/writing into files

9. Class NCIEHistogram

public class NCIEHistogram
extends Histogram
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003
Description: NRG histogram inherits the functionality of the abstract class Histogram

Constructor Summary

NCIEHistogram() The number of bins is 25*25 (25 bands for each dimension).

Method Summary

float	getDimensionData(Pixel p, int dim) returns the 2D info from the pixel first dimension is the normalized R information second dimension is the normalized G information

Methods inherited from class Histogram
addNewDataItem, calculateProbabilities, getSignature, getValue, intersect

10. Class NRGHistogram

public class NRGHistogram
extends Histogram
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003
Description: NRG histogram inherits the functionality of the abstract class Histogram

Constructor Summary

NRGHistogram() The number of bins is 25*25 (25 bands for each dimension).

Method Summary

float	getDimensionData(Pixel p, int dim) returns the 2D info from the pixel first dimension is the normalized R information second dimension is the normalized G information

Methods inherited from class Histogram
addNewDataItem, calculateProbabilities, getSignature, getValue, intersect

11. Class ObjectAppearance

public class ObjectAppearance
extends java.lang.Object

Constructor Summary

ObjectAppearance(int label, Pixel[] p) Input: label: the order of the object in the frame, represented by sequence numbers: 1, 2, etc p: the collection of pixels in the frame representing that object Description: Dividing the pixels

into three regions.

Method Summary

void	createGMMs() Input: none Output: none Description: creates the GMM information for the object pixels
void	createHistograms() Input: none Ouptut: none Description: Creates a histogram information for the object pixels.
Signature	generateSignature(int type) Input: type: there are currently three types of signatures (the number will increase as more colour representations are added) 0 - RGB Signature 1 - HS Signature 2 - Normalized RG Signature Output: Signature: the returned signature represents the sum of the regions signatures separated by a semicolon.
void	writeObject(java.awt.image.BufferedImage img) Input: img: A buffered image object to which the pixel information is to be written Output: none Description: Loops around the pixel information adding the pixel information to the image.

12. Class Person

public class **Person**
extends java.lang.Object
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003
Description: Person class keeps signature information of the people tracked throughout a movie/session

Constructor Summary

Person(Signature sig) constructor, the person is constructed by a unique signature

Method Summary

void	adaptSignature(Signature s) This method is to be implemented yet! It adapts the signature of the person
Signature	getSignature() returns the signature

13. Class Pixel

public class **Pixel**
extends java.lang.Object

Constructor Summary

Pixel(int x, int y, int R, int G, int B)

Input: x: the x co-ordinate of the pixel y: the y co-ordinate of the pixel R: the R of the RGB colour information of the pixel G: the G of the RGB colour information of the pixel B: the B of the RGB colour information of the pixel

Method Summary

int	getB() returns the blue component of the RGB colour space
float	getCIEx() returns the x component of the normalized CIE colour space
float	getCIEX() returns the X component of the CIE colour space
float	getCIEy() returns the y component of the normalized CIE colour space
float	getCIEY() returns the Y component of the CIE colour space
float	getCIEz() returns the z component of the normalized CIE colour space
float	getCIEZ() returns the Z component of the CIE colour space
float[]	getDataVector() Output: float [] - an array of the RGB float values
int	getG() returns the green component of the RGB colour space
float	getH() returns the hue component of the HSI colour space
float	getNormg() returns the normalized green component of the normalized RG colour space
float	getNormr() returns the normalized red component of the normalized RG colour space
int	getR() returns the red component of the RGB colour space
float	getS() returns the saturation component of the HSI colour space
int	getX() returns the x coordinate of the pixel

int	getY() returns the y coordinate of the pixel

14. Class Region

public class Region
extends java.lang.Object

Field Summary	
static int	GMM_NOS the number of the Gaussians in the GMM
static int	HISTOGRAM_NOS the number of histograms being created for each region currently histogram no 0 represents the RGB colour space histogram no 1 represents the HS colour space histogram no 2 represents the normalized RG colour space

Constructor Summary
Region(java.util.Vector v) Input: v - the vector that contains the pixels for that region Description: the constructor converts the vector into an array of pixels

Method Summary	
void	createHistograms() Input: none Output: none Description: loops through the region's pixels adding each pixel as a new data item to the histogram
void	generateGMM() iterates the EM algorithm to generate the clusters to which the thing converges
java.lang.String	generateSignature(int type) Input: type: NOT USED HERE // used in the case of fixed three initialisations Output: String: the frequences of the bins of the histogram

15. Class RGBGMM

public class RGBGMM
extends GMM
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 24th 2003 Description: RGBGMM class represents the GMM that contains RGB colour space information the class inherits the GMM class

Field Summary

Fields inherited from class GMM

covariance, dimension, mean, weight

Constructor Summary
RGBGMM()   initialises the Gaussian with the dimension of the RGB colour space, i.e.

Method Summary	
void	initializeMean() initializes the GMM mean to random values
void	initializeMean(float[] initialMean) Input: initialMean - array of inital means for the Gaussian Description: initializes the mean to the one provided by the parameter
void	initializeMean(int gNo, float[] initialMean) Input: gNo - the order of this Gaussian in the GMM initMean - an array containing the initial values of the means for all the Gaussians in the GMM Format: initMean array has the format {x1,x2,x3, x1, x2, x3, etc } Description: Initializes the Gaussian to the mean from the array containing all the means in the above format

Methods inherited from class GMM
getCovariance, getMean, getWeight, getWeightedGaussian, initializeCovariance, initializeWeight, updateParameters

16. Class RGBHistogram

public class RGBHistogram
extends Histogram
Dima J Al Damen University of Leeds First Written: June 29th 2003 Last updated: July 2nd 2003 Description: Region class contains the collection of pixels belonging to one region of the object remember that the object is divided into three regions.

Constructor Summary
RGBHistogram()   Creates a histogram of 15*15*15 bins; each dimension ranges from 0 to 255 with 15 equal divisions.

Method Summary	
float	getDimensionData(Pixel p, int dim) Input: p : pixel whose information is to be recorded dim: 3 dimensions are represented in the RGB histogram; these represent: 0 is R, 1 is G, 2 is B

Methods inherited from class Histogram

addNewDataItem, calculateProbabilities, getSignature, getValue, intersect

17. Class Signature

public class Signature
extends java.lang.Object

Constructor Summary
Signature(java.lang.String s)

Method Summary	
float[]	compareGMMRegionsSignature(Signature s) Input: s: another signature of the same type Output: float [] : array indicating the degree to which region's match occurs; it returns a number between 0 and 1 The closer this number to 1, means the matching probability is higher Description: This method merely compares the difference between the parameters of all the Gaussians in the GMM The MAJOR problem with this technique is that the intialization is expected to give different order of the Gaussians; in addition, it gives the same value to the mean, weight and covariance while actually we should merely compare the mean
float[]	compareRegionsSignature(Signature s) Input: s: another signature of the same type Output: float []: array indicating the degree to which each region's match occurs; it returns a number between 0 and 1 The closer this number to 1, means the matching probability is higher Description: This method uses a technique called histogram intersection for comparing two histograms See Swain and Ballard 1990 for mathematical explanation (Log file)
float	compareSignature(Signature s) Input: s: another signature of the same type Output: float: indicating the degree to which the matching occurs; it returns a number between 0 and 1 The closer this number to 1, means the matching probability is higher Description: This method uses a technique called histogram intersection for comparing two histograms See Swain and Ballard 1990 for mathematical explanation (Log file)
java.lang.String	getSignature()

18. Class VectorQuantisation

public class VectorQuantisation
extends java.lang.Object

Constructor Summary
VectorQuantisation(int clustersNo)

Input: clustersNo - the number of clusters in the VQ algorithm Description: constructor

Method Summary	
float[]	getClusterMeans() returns the means
void	run(Pixel[] pixels) Input: pixels - the data set to be considered for the algorithm Description: the K-means algorithm to be run

19. Class VisualSignature

public class VisualSignature
extends java.lang.Object

Constructor Summary
VisualSignature()

Method Summary	
static void	main(java.lang.String[] args)

Appendix F – Sample Experiments Results

F.1 Experiment 1 Result Sample

Table F.1. 1 Colour Histograms - Same Person's Results Example (person 9)

No	Frame1	Frame2	RGB	HS	NRG	Norm CIE	CIE
1	70	75	0.783788	0.663902	0.878266	0.663902	0.811216
2	75	80	0.791473	0.689885	0.883327	0.689885	0.831823
3	80	85	0.741687	0.620473	0.834045	0.620473	0.811042
4	85	90	0.80342	0.649487	0.892926	0.649487	0.846769
5	90	100	0.768227	0.659778	0.894106	0.682431	0.774972
6	100	105	0.811903	0.73452	0.929557	0.73452	0.827284
7	105	110	0.766434	0.703191	0.910018	0.703191	0.792601
8	110	115	0.754456	0.661261	0.934393	0.661261	0.79559
9	115	120	0.811187	0.648868	0.925337	0.648868	0.832065
10	120	125	0.695867	0.657744	0.924466	0.657744	0.717528
11	125	130	0.767432	0.669518	0.904024	0.669518	0.825166
12	130	200	0.772865	0.619906	0.890093	0.619906	0.767988
13	200	205	0.83967	0.695929	0.901469	0.695929	0.876352
14	205	210	0.802936	0.673493	0.911821	0.673493	0.845806
15	210	215	0.804216	0.654081	0.813316	0.654081	0.848082
16	215	220	0.80859	0.664145	0.871061	0.664145	0.845672
17	220	225	0.788133	0.685318	0.921429	0.685318	0.846037
18	225	230	0.800606	0.586818	0.89525	0.586818	0.820219
		Average:	0.784049	0.66324	0.895272	0.664498	0.817567
		Standard Deviation:	0.032631	0.033573	0.031873	0.033859	0.037322

Table F.1. 2 Colour Histograms - Same Person's Results Average

No	RGBAvg	HSAvg	NRGAvg	Avg Norm CIE	Avg CIE
1	0.799311	0.700184	0.881816	0.784645	0.703932
2	0.798206	0.685333	0.891465	0.805961	0.708299
3	0.782643	0.672638	0.855998	0.781627	0.691521
4	0.79499	0.698098	0.902021	0.778833	0.709631
5	0.789932	0.670588	0.878326	0.786981	0.685816
6	0.777956	0.678394	0.881468	0.761346	0.687478
7	0.784558	0.681203	0.894882	0.76926	0.683652
8	0.797051	0.67685	0.925486	0.781864	0.682221
9	0.667944	0.564198	0.809095	0.685715	0.585347
10	0.696971	0.593796	0.882416	0.710096	0.63438
11	0.757668	0.62022	0.875617	0.793109	0.66161
12	0.823362	0.658244	0.891608	0.81368	0.672508
13	0.824076	0.713199	0.90314	0.814027	0.718391
14	0.829498	0.715346	0.90032	0.822701	0.726786
15	0.805853	0.682266	0.862552	0.822264	0.70351
16	0.817394	0.683121	0.880116	0.810472	0.683772
17	0.837565	0.677765	0.887362	0.82741	0.699673
18	0.777965	0.654207	0.896384	0.77356	0.661063
Average:	0.78683	0.668092	0.883337	0.784642	0.683311
STDEV:	0.043466	0.039327	0.024318	0.037491	0.033171

Table F.1. 3 Colour Histograms - Different People's Results Example (person 4 and person 9)

No	Person1	Person2	RGB	HS	NRG	Norm CIE	CIE
1	75	75	0.58512	0.508992	0.838274	0.508992	0.624382
2	80	80	0.624595	0.490384	0.814545	0.490384	0.676512
3	85	85	0.663668	0.490541	0.766677	0.490541	0.739022
4	90	90	0.616805	0.478607	0.815308	0.478607	0.64067
5	95	95	0.544228	0.45742	0.777204	0.45742	0.640328
6	100	100	0.560392	0.411698	0.770532	0.411698	0.59399
7	105	105	0.564821	0.422812	0.80722	0.422812	0.618179
8	110	110	0.555556	0.400769	0.793712	0.400769	0.583088
9	115	115	0.538677	0.414346	0.779228	0.414346	0.579783
10	120	120	0.338942	0.241588	0.679251	0.241588	0.372503
11	200	200	0.51502	0.422126	0.801176	0.422126	0.529834
12	205	205	0.574429	0.420774	0.854404	0.420774	0.572768
13	210	210	0.553709	0.352763	0.776597	0.352763	0.552627
14	215	215	0.576549	0.441164	0.796198	0.441164	0.58592
15	220	220	0.58431	0.477481	0.813624	0.477481	0.651852
16	225	225	0.601872	0.444766	0.820282	0.444766	0.631839
17	230	230	0.611519	0.398998	0.803129	0.398998	0.641398
		Average	0.565307	0.427955	0.79455	0.427955	0.602041
		STDEV:	0.06867	0.06287	0.03803	0.06287	0.07719

Table F.1. 4 Colour Histograms - Different People's Results Average

No	RGBAvg	HSAvg	NRGAvg	Avg Norm CIE	Avg CIE
1	0.4491	0.419097	0.680466	0.404114	0.654165
2	0.452101	0.414916	0.650488	0.402338	0.623145
3	0.434753	0.360252	0.634453	0.338537	0.612416
4	0.477907	0.390034	0.703156	0.375272	0.684464
5	0.427372	0.413492	0.670552	0.40617	0.652777
6	0.42406	0.378293	0.685227	0.372725	0.67101
7	0.429048	0.394462	0.714674	0.389737	0.699249
8	0.418002	0.37748	0.706576	0.373598	0.692053
9	0.419364	0.401942	0.699705	0.399875	0.686452
10	0.254	0.272785	0.638041	0.277984	0.631173
11	0.415352	0.355354	0.679284	0.344225	0.658968
12	0.416992	0.369012	0.687439	0.360385	0.659611
13	0.435067	0.337172	0.67711	0.334573	0.660529
14	0.456363	0.371737	0.671408	0.360165	0.65061
15	0.440681	0.370118	0.677547	0.352225	0.654867
16	0.443986	0.385379	0.709025	0.375482	0.690482
17	0.442954	0.381496	0.68766	0.378579	0.668416
Average	0.425712	0.37606	0.680754	0.367411	0.661787
STDEV:	0.04725	0.03451	0.02331	0.03233	0.02448

F.2 Experiment 2 Result Sample

Table F.2. 1 The Three-Regions Colour Histograms - Same Person's Results Example (person 7)

No	Frame1	Frame2	Head Region	Torso Region	Lower Region
1	45	50	0.54767	0.784107	0.871164
2	50	55	0.729848	0.782697	0.827189

3	55	60	0.760678	0.860827	0.84116
4	60	65	0.775633	0.810722	0.887548
5	65	70	0.727614	0.65961	0.842316
6	70	75	0.706554	0.818281	0.905418
7	75	80	0.712932	0.75708	0.766576
8	80	85	0.709211	0.79325	0.785272
9	85	90	0.580705	0.734999	0.836633
10	90	95	0.747473	0.689195	0.831081
11	200	205	0.638074	0.766467	0.821036
12	205	210	0.694688	0.764673	0.803126
13	210	215	0.021021	0.650236	0.756543
14	215	220	0.007273	0.623615	0.819743
15	220	225	0.782745	0.699397	0.868677
16	225	230	0.804403	0.758088	0.746686
17	230	235	0.854633	0.869768	0.897945
18	235	240	0.624333	0.751597	0.693595
		Average	0.634749	0.754145	0.822317
		STDEV	0.238443	0.06858	0.056333

Table F.2. 2 The Three-Regions Colour Histograms - Same Person's Results Average

No	Head Region	Torso Region	Lower Region
1	0.699334	0.811808	0.844195
2	0.762384	0.796783	0.79809
3	0.687689	0.749809	0.79823
4	0.755069	0.770807	0.818747
5	0.634749	0.754145	0.822317
6	0.74347	0.73994	0.818067
7	0.757304	0.790161	0.771769
8	0.699334	0.811808	0.844195
9	0.762384	0.796783	0.79809
10	0.687689	0.749809	0.79823
11	0.755069	0.770807	0.818747
12	0.634749	0.754145	0.822317
13	0.74347	0.73994	0.818067
14	0.757304	0.790161	0.771769
Average	0.72	0.77335	0.810202
STDEV	0.04592	0.02589	0.0222

Table F.2. 3 The Three-Regions Colour Histograms - Different People's Results Example (person 6 and person 9)

No	Person1	Person2	Head Region	Torso Region	Lower Region
1	95	70	0.604685	0.747828	0.221794
2	100	75	0.609487	0.546519	0.174187
3	105	80	0.619874	0.603031	0.136827
4	110	85	0.645146	0.711026	0.218961
5	115	90	0.578764	0.568907	0.147685
6	120	95	0.596603	0.676696	0.143001
7	125	100	0.53779	0.513897	0.173522
8	130	105	0.491598	0.6419	0.150161
9	135	110	0.548552	0.720868	0.165719
10	140	115	0.428853	0.475916	0.190483
11	205	205	0.782479	0.727025	0.209431

12	210	210	0.71871	0.700492	0.148206
13	215	215	0.706391	0.618993	0.172395
14	220	220	0.7424	0.655587	0.19176
15	225	225	0.720396	0.615825	0.181425
16	230	230	0.670343	0.653073	0.160806
17	235	235	0.758293	0.576172	0.170998
		Average:	0.632962	0.632574	0.173962
		STDEV	0.09861	0.0784	0.02589

Table F.2. 4 The Three-Regions Colour Histograms - Different People's Results Average

No	Head Region	Torso Region	Lower Region
1	0.60816	0.11988	0.521004
2	0.51065	0.031881	0.564217
3	0.535291	0.162014	0.19813
4	0.632962	0.632574	0.173962
5	0.584585	0.075494	0.137682
6	0.618845	0.062761	0.532756
7	0.667681	0.435521	0.35573
8	0.645154	0.117639	0.482134
9	0.60816	0.11988	0.521004
10	0.51065	0.031881	0.564217
11	0.535291	0.162014	0.19813
12	0.632962	0.632574	0.173962
13	0.584585	0.075494	0.137682
14	0.618845	0.062761	0.532756
15	0.667681	0.435521	0.35573
16	0.645154	0.117639	0.482134
Average	0.600416	0.204721	0.370702
STDEV	0.050664	0.199893	0.166481

F.3 Experiment 3 Result Sample

Table F.3. 1 GMMs - Three different initialisations - Same Person's Results Example (person 8)

No	Frame1	Frame2	first initialization	second intialization	third initialization
1	158	159	90.40689	81.53542	54.20696
2	159	160	111.0757	49.78805	68.19376
3	160	161	63.32861	47.87495	56.12399
4	161	162	117.0086	61.45155	28.79139
5	162	163	89.90151	66.69572	61.57787
6	163	164	166.2209	41.33645	62.48256
7	164	165	123.5842	64.24509	68.38767
8	165	166	129.6157	73.34414	39.568
9	166	167	120.407	77.30141	78.04556
10	167	168	118.0618	83.58366	76.93469
11	168	169	128.8709	26.21415	91.78927
12	169	170	23.4623	73.57582	44.34175
13	170	171	51.16411	54.14343	48.42452
14	171	172	95.64689	38.73488	81.49001

15	172	173	95.48828	56.0713	34.16813
16	173	174	85.15855	64.51768	16.57458
17	174	175	102.3285	39.51549	48.12844
18	175	176	40.17312	41.7879	49.79568
19	176	177	69.50885	24.68348	14.90332
		Average	95.86381	56.12635	53.89095
		STDEV	34.98996	17.83185	21.30843

Table F.3. 2 GMMs - Three different initialisations – Different People's Results Example (person 4 and person 6)

No	Person1	Person2	first initialization	second initialization	third initialization
1	75	95	69.55824	153.5919	76.76162
2	76	96	112.2055	108.4251	66.01453
3	77	97	95.42398	147.9477	82.47157
4	78	98	129.0146	109.2804	70.78029
5	79	99	69.50878	147.4008	85.71298
6	80	100	129.6206	31.19018	103.4966
7	81	101	97.23277	150.6929	86.5266
8	82	102	73.38687	103.6564	124.6911
9	83	103	137.7201	40.32187	137.8526
10	84	104	113.4707	116.2402	132.5874
11	85	105	120.2935	141.4206	123.3809
12	86	106	100.2917	86.43783	102.5344
13	87	107	93.36336	44.96427	81.41017
		Average	103.1608	106.2746	98.01697
		STDEV	23.07923	43.90793	24.50493

Table F.3. 3 GMMs – VQ Initialisation - Same Person's Results Example (person 6)

No	Frame1	Frame2	Head	Torso	Lower Region
1	95	96	61.34956	89.80418	62.01638
2	96	97	66.9791	27.1709	120.8834
3	97	98	85.66083	115.4728	106.3894
4	98	99	86.7553	21.95724	117.9857
5	99	100	101.4245	139.1021	132.66
6	100	101	23.33291	141.91	40.77999
7	101	102	94.65022	135.3635	56.1339
8	102	103	96.79768	193.1328	78.3304
9	103	104	62.27902	187.2902	250.3836
10	104	105	26.26728	27.59061	210.2275
11	105	106	137.3005	38.1493	60.92834
12	106	107	112.5124	58.83146	78.92648
13	107	108	90.0277	49.88718	39.73418
14	108	109	166.2796	72.7421	91.85803
15	109	110	157.1781	87.9208	70.76028
16	110	111	114.3384	24.57338	83.13697
17	111	112	98.15335	169.818	89.79153
18	112	113	44.50567	113.6423	14.69537
19	113	114	3.959083	74.47588	12.79784
20	114	115	103.2465	94.07244	15.96226
21	115	116	124.2837	47.28339	34.16155

22	116	117	154.6037	26.21257	29.92438
23	117	118	90.40318	17.26723	64.17437
24	118	119	33.46307	28.03339	59.17323
25	119	120	36.71792	155.0391	26.68092
		Average	86.89877	85.46971	77.93984
		STDEV	43.48379	56.33462	57.09529

Table F.3. 4 GMMs – VQ Initialisation - Different People's Results Example (person 6 and person 8)

No	Person1	Person2	Head Region	Torso Region	Lower Region
1	95	158	69.20941	94.88194	160.6167
2	96	159	132.7674	74.10882	194.8957
3	97	160	36.7776	72.88823	119.9566
4	98	161	47.79324	116.2515	142.5668
5	99	162	120.235	139.6116	134.1079
6	100	163	65.25122	169.9497	152.5992
7	101	164	120.9997	133.3706	201.7135
8	102	165	170.0051	182.0181	161.3654
9	103	166	130.0181	139.9126	184.0171
10	104	167	143.3504	150.3589	151.8367
11	105	168	114.7963	173.5564	206.0998
12	106	169	110.2112	205.7955	144.4672
13	107	170	111.7683	185.5548	93.15631
14	108	171	167.1075	173.1541	164.5613
15	109	172	81.70349	195.7163	131.6556
16	110	173	159.9202	207.9869	149.8174
17	111	174	131.5514	178.209	183.6118
18	112	175	103.7649	152.6364	151.7103
19	113	176	110.6328	195.0302	185.4731
20	114	177	72.23201	149.4797	150.9704
21	115	178	101.53	66.66434	184.5351
		Average	109.6012	150.3398	159.5111
		STDEV	36.96886	43.86685	28.24183

F.4 Experiment 4 Result Sample

Table F.4. 1 Gradual Lighting Changes - Same Person's Results Example (person 3)

No	Frame1	Frame2	RGB	HS	CIE
1	45	50	0.782765	0.583395	0.772601
2	50	55	0.728495	0.571498	0.790377
3	55	60	0.751887	0.588505	0.798681
4	60	65	0.711073	0.591054	0.760522
5	65	70	0.571607	0.531136	0.668706
6	70	75	0.724472	0.616839	0.787478
7	75	80	0.693387	0.579343	0.730608
8	80	85	0.730436	0.62789	0.827597
9	85	90	0.749486	0.601882	0.829668
10	90	95	0.766146	0.604068	0.837458
11	95	100	0.755219	0.67123	0.856501
12	180	185	0.605257	0.560694	0.736864
13	185	190	0.74212	0.602545	0.822721

14	190	195	0.665989	0.619438	0.729676
15	195	200	0.75581	0.646914	0.804026
16	200	205	0.721086	0.581562	0.773819
17	205	210	0.715071	0.615727	0.767383
18	210	215	0.674178	0.527951	0.739878
19	215	220	0.572058	0.557513	0.659196
20	220	225	0.675997	0.575285	0.748069
21	225	230	0.742567	0.58267	0.794464
22	230	235	0.7624	0.566092	0.842238
		Average	0.708977	0.591056	0.776297
		STDEV	0.060017	0.034253	0.052398

Table F.4. 2 Gradual Lighting Changes - Same Person's Results Example (person 3 and person 2)

No	Person1	Person2	RGB	HS	CIE
1	40	45	0.236563	0.283839	0.270636
2	45	50	0.301857	0.353094	0.34057
3	50	55	0.315217	0.341714	0.402341
4	55	60	0.323423	0.391026	0.36552
5	60	65	0.369791	0.316463	0.446154
6	65	70	0.32958	0.38764	0.367468
7	70	75	0.378339	0.346409	0.431884
8	75	80	0.371993	0.385748	0.437036
9	80	85	0.405773	0.416633	0.45625
10	85	90	0.395803	0.414049	0.436423
11	90	95	0.382294	0.503675	0.414771
12	95	100	0.3864	0.442787	0.455311
13	215	180	0.320818	0.400538	0.371486
14	220	185	0.35305	0.374288	0.39975
15	225	190	0.321107	0.336916	0.404876
16	230	195	0.376434	0.376669	0.411947
17	235	200	0.36574	0.45303	0.420842
18	240	205	0.359846	0.333901	0.421682
19	245	210	0.35312	0.386628	0.435861
20	250	215	0.294648	0.340721	0.376951
21	255	220	0.375262	0.365454	0.429094
22	260	225	0.315472	0.401656	0.364369
23	265	230	0.328333	0.301221	0.386113
24	270	235	0.32364	0.384275	0.384832
		Average	0.345188	0.376599	0.40134
		STDEV	0.039024	0.049585	0.042223

Table F.4. 3 Severe Lighting Changes - Same Person Results Example (person 3 and person 2)

No	Frame1	Frame2	RGB	HS	CIE
Person1	45	100	0.66016	0.452471	0.667052
Person2	185	235	0.369803	0.277276	0.468605
Person3	80	135	0.429308	0.326339	0.520293
Person4	40	100	0.50932	0.379977	0.586613

Person5	210	270	0.348867	0.309468	0.526646
		Average	0.463492	0.349106	0.553842
		STDEV	0.126356	0.068714	0.075863

Appendix G – List of Figures

Appendix H – List of Tables