Recognising Individuals by Appearance across Non-overlapping Stereo Cameras

Bram Stoeller

August 2012

Supervised by:

dr. ir. Leo Dorst

dr. Gwenn Englebienne

This thesis is submitted in partial fulfillment of the requirements
for the degree of Master of Science in Artificial Intelligence
with the specialization in Intelligent Systems and Computer Vision.

UNIVERSITY OF AMSTERDAM

Abstract

Faculty of Science

Informatics Institute

Master of Science

by Bram Stoeller

Optical people tracking systems need a robust representation of each individual in order to follow him or her over multiple non-overlapping cameras. Many researchers have done this for sequences of conventional video frames, i.e. digital images. But what if the three-dimensional location of each pixel is known in addition to its colour? What if the input is a three-dimensional point cloud instead of a flat image? This extra information about the location of each pixel introduces alternative representations based on both colour and three-dimensional shape.

The study described in this thesis investigates several appearance models to make inter-camera recognition of people possible. A novel circular hue-saturation histogram appears to be amongst the most accurate and efficient models and a classic k-means in a hybrid colour-geometry space yields the same results with only 11% of the data transfer.

Finally, excessive noise in the data prevents an adapted mixture of Gaussians model to be successful within this study. Nevertheless, the predictions for future research concerning this method are promising.

Acknowledgements

Contents

Notation

While this thesis is intended to be mainly scientific, it will also serve as an implementation documentation from which all algorithms should be reproducible. An overview of the used notations and some implementation notes are provided to support future programmers. All algorithms are implemented in MATLAB®, which uses mainly row vectors instead of column vectors and starts its indices at 1 instead of 0. The report will follow this convention. To avoid confusion, the notations used in this thesis are explained in table 1.

x	a scalar.	0.1
$\vec{x}$	a direction vector of size $1 \times m$, where m is the number of dimensions (around 3).	$[0.1\ 0.2\ 0.3]$
$\vec{x}^{\mathrm{D}}$	the diagonal matrix of size $m \times m$ based on the vector $\vec{x}$ where $\vec{x}^{\mathrm{D}}_{ii} = \vec{x}_i$.	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$
$\mathbf{x}$	a data vector of size $n \times 1$, where n is the number of elements (around 10^3).	$\begin{bmatrix} 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$
X	a calculation matrix (e.g. a rotation/translation matrix).	$\begin{bmatrix} 0.1 & 0.4 & 0.7 \\ 0.2 & 0.5 & 0.8 \\ 0.3 & 0.6 & 0.9 \end{bmatrix}$
$\mathbf{X}$	a data matrix of size $n \times m$, containing n direction vectors of length m. The $|\circ|$ operator is defined as: $|\mathbf{X}| = n$.	$\begin{bmatrix} 0.1 & 0.4 & 0.7 \\ \vdots & \vdots & \vdots \\ 0.3 & 0.6 & 0.9 \end{bmatrix}$
$\mathcal{X}$	an object. This can be a 3D matrix representing an image or point cloud of h rows, w columns and m dimensions or a more complex object.	
I_m	Identity matrix with m rows and m columns.	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

TABLE 1: Notation of variables

Chapter 1

Introduction

With the ever growing population of our planet the needs for automated security increase dramatically. The goal of this study is to find out if it is possible to follow people across multiple *stereo cameras* at different locations. I will not propagandise the Big Brother dystopia, as the English novelist George Orwell sketched in his book "1984"; on the contrary, the methods proposed in this thesis are designed to follow people without violating anyone's privacy. In contrast to controversial surveillance methods, the cameras in this research do not output a single image. Only a small amount of data describing colour features of a person will be used for inter-camera recognition.

Following or *tracking* people with a camera is a challenging computer task. Firstly it can be hard to distinguish foreground (the person to track) from background (the environment). Secondly by the effect of occlusion, a camera does not always have clear sight on a person. Occlusion occurs when the line of sight from the camera to a person is blocked, either by a person or by an object. To overcome these issues, *Eagle Vision Systems* (EVS)[1] has developed a system to detect and track people using stereo cameras.

A stereo camera typically has two lenses, each with its own *image sensor*. The lenses have the same direction and are shifted a number of decimetres in the plane perpendicular to the viewing axis. The camera records two images at once, a *main image* obtained by the first sensor and an *auxiliary image* obtained by the second sensor. Using these two images, the stereo camera can compute the distance of the camera to the pixels in the images. From this distance it is possible to create a three-dimensional representation (i.e. a *point cloud*) of the scene. The details of this process are out of the scope of this study, but to give an impression, the result is illustrated in figure 1.1. The main image and the auxiliary image

---

[1] *Eagle Vision Systems B.V.* is a medium sized machine vision company in Naarden, employing around 15 people. They have experience in vision systems for over 15 years, mainly in the food, beverage & packaging industry and in people logistics.

are shown in figure 1.1(a) and 1.1(b), and the result of combining these two images is a point cloud, shown in figure 1.1(c).
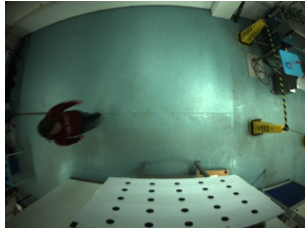
(a) Main image acquired by the stereo camera.
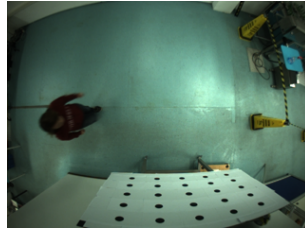
(b) Auxiliary image acquired by the stereo camera.
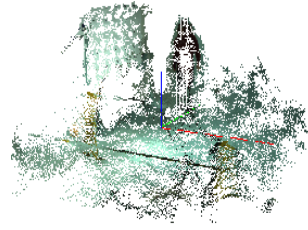
(c) The point cloud constructed from both images.

(d) Main image after removing distortion and applying colour correction.
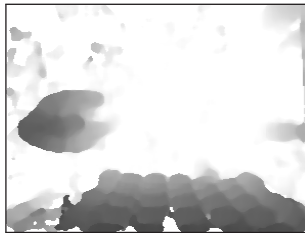
(e) The z-map. The ground plane is white, darker pixels are closer to the camera.
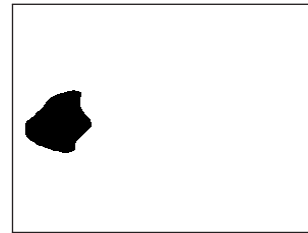
(f) The environment is filtered using a background z-map and one blob is detected as a person using a threshold.

FIGURE 1.1: Detection of a person in a stereo image within the Eagle Eye (frame number: EVS.5.2.94).

The cameras used in this study are attached to the ceiling and directed straight down. One advantage of this situation is that occlusion rarely occurs from this bird's-eye view. Furthermore, from this direction the distance of pixels to the camera corresponds to their height in the real world. The result of this is that there is a great difference between the distance to people walking through the view of the camera and the distance to the the background (the floor). This makes it relatively easy to detect people. This is illustrated in figure 1.1(e) and 1.1(f). From the main image computed by the stereo camera in figure 1.1(d), a *z-map* can be constructed that colours pixels lighter or darker, depending on their distance to the camera, as shown in figure 1.1(e). Then, the z-map is filtered in order to detect *blobs* that indicate people in the image. The result is shown in figure 1.1(f). The person visible in the image is clearly detected as a blob in the z-map.

A downside of this camera positioning is that the *field of view*, the part of the world that the camera can see, is relatively small. Wide angle lenses are used to increase the field of view, but even so the angle is limited because of the *distortion correction* that is necessary for *stereo vision*. Furthermore, figure 1.1(d) shows that the effective field of view is smaller than the visual field of view. An area of approximately 3 by 4 metres can be seen at the floor level, but a person standing in this area will not always be entirely visible. A full-length detection of a person is only possible in an area of approximately

2 by 3 metres. That is why EVS uses multiple cameras with overlapping *fields of view*. These cameras are connected to a *global tracking engine*, to track people in a larger area. Depending on the height of the ceiling, around 15 to 30 cameras are needed to cover an area of 100 square metres.

However, there is mostly no need to track a person constantly; frequently detecting and recognising a person (e.g. by placing one camera on each intersection in an office building) is sufficient for most applications. When the fields of view no longer overlap, tracking a person across the view of multiple cameras becomes a completely different task. The global tracking engine must create a representation of the tracked person in order to recognise him or her at some later time when he or she reappears. The only data available for such an *appearance model* is a coloured point cloud and the detected position of a person that I illustrated in figure 1.2.

(a) The point cloud of the entire scene.　(b) The extracted point cloud of detected a person.
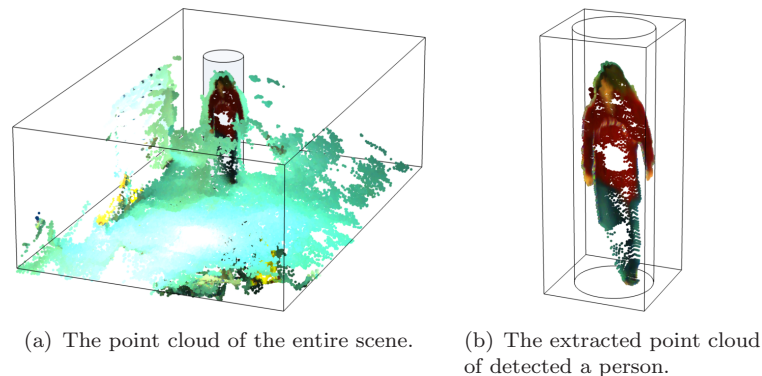
FIGURE 1.2: A cylinder containing a person is extracted from the scene. The colours have been manually adjusted for visualisation purposes.

The two main purposes of this study will be firstly to investigate if it is possible to use appearance models to track people across multiple non-overlapping stereo cameras and secondly to find out which appearance models is most suitable, by designing, testing and analysing a great number of methods. We will start by investigating appearance models that describe a person by the colours of his or her clothes, skin and hair. Figure 1.3 shows typical images recorded by the *Eagle Eye*, as EVS calls its people detection units with built-in stereo cameras. These images indicate that it might be difficult to gather enough colour information for a robust representation, because we might not see a lot of pixels from the person's clothes. However, we know the height of these pixels in the real world, which provides us with a lot more data. The expectation is that the information in this data can help us to build a robust representation of a person. For example we could group pixels from the lower, middle and upper part of the person's point cloud and use this as an appearance model like Zajdel et al. did for flat images [1]. This is indeed one of the appearance models implemented in this study, see section 4.3.

| EVS.5.2.94 | EVS.5.3.200 | EVS.5.3.266 | PSL.2.21.377 | PSL.2.33.605 | PSL.2.30.1573 |
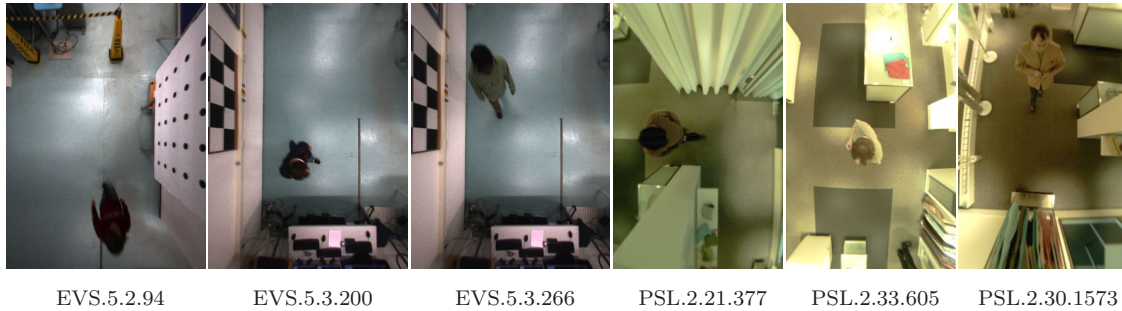
FIGURE 1.3: A selection of 6 recorded frames, each containing a person.

In chapter 2 of this report EVS's tracking system, the *Eagle Grid*, is explained and the findings of previous research will be discussed. Chapter 3 will provide some background information on *colour spaces*. Chapter 4 gives an extensively illustrated description of the investigated appearance models, their advantages and disadvantages and the way they were implemented for this study. How the appearance models are tested and compared will be explained in chapter 5, and chapter 6 shows the conducted experiments and their results. Finally, chapter 7 will give clarification about which appearance model to choose under which circumstances and some additional ideas for future research.

Chapter 2

Background

This chapter will provide some background information on the system used in this study. All models discussed in this thesis will be implemented in an existing *tracking system*. This system, the Eagle Grid, will be explained in section 2.1.

In Section 2.2 some related work will be presented that forms the basis of this thesis.

2.1   The Eagle Grid

The domain of this study is limited to *people tracking* in an indoor environment, with steady lighting, using ceiling-mounted stereo cameras. The people at *Eagle Vision Systems* (EVS) are experts in this field and they have developed a system specifically designed to do this. This study aims at improving that system and all experiments were conducted with it in mind. Their system, the Eagle Grid, is a centralised multi-camera tracking platform that serves as the foundation of numerous tracking algorithms with different applications like securing areas, counting people and several other statistical analyses. It consists of a number of stereo cameras, the *Eagle Eyes* and a *tracking engine*, the *Eagle Tracking Engine*. Currently the Eagle Tracking Engine assumes that the fields of view of the Eagle Eyes overlap. The goal of this study is to develop a method that can be used to extend the framework to work with non-overlapping Eagle Eyes.

2.1.1   Eagle Eyes

I will now discuss the Eagle Eyes in more detail. They consist of two cameras and a processing unit. The cameras are synchronised mono-CCD colour cameras. They capture two images at once, which are combined by the processing unit in order to construct a

point cloud of the scene. In this construction the real-world location of each pixel in the frame is estimated. Subsequently, a z-map of the scene is created. This z-map can be represented as a monochrome image representing the z position of each pixel, i.e., the height of the pixels with respect to the ground plane (for an example, see figure 1.1(e)). Because pixels that belong to people are very high in comparison to pixels that belong to the floor, people will be seen as blobs of high pixels in the z-map. The processing detects people on the scene by searching for such blobs. Finally, after a captured frame is processed, some information about each blob detected in that frame is sent to the tracking engine. All in all, the Eagle Eyes are responsible for image capturing, stereo matching, point cloud construction, people detection and data transmission.
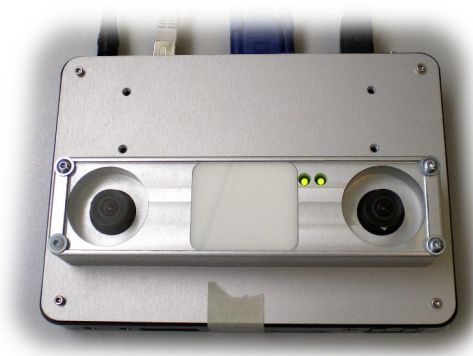
FIGURE 2.1: Prototype of the latest Eagle Eye, June 2012.

While the Eagle Eye looks like a stereo camera (figure 2.1), it is more appropriate to think of it as a *human detection sensor*. Indeed, the output of the Eagle Eye is not an ordinary video stream but a sequence of *feature vectors* accompanied by a timestamp, each representing a person detected on the scene. Currently the feature vector consists only of the location and a *bounding box* of the person, but this vector should be extended with information about a person's appearance. In order to do this, one of the appearance models proposed in this thesis should be applied to each detected person and the resulting *representation* should be appended to the current feature vector.

2.1.2   Eagle Tracking Engine

Turning to the Eagle Tracking Engine, it receives representations of all detected people within one frame from the Eagle Eyes. The tracking engine stores these representations in a buffer. They are processed in batches with a small delay to overcome the effect of network delays.

The tracking engine keeps a number of *tracks*. Each track represents a person in the real world. When a person is detected by the Eagle Eye, it is either assigned to the existing

track that matches it most closely, or a new track is created if no matching track exists. The tracks contain the routes that the person has travelled. In this study these tracks will be extended with a representation of the person. In this new method, all incoming representations will be compared against the representation of the known tracks to find the best match. Existing tracks are extended and their representations are updated. When a person appears to be new, a new track is created. Its representation will be based on the representation of that person.

2.1.3 Actuators
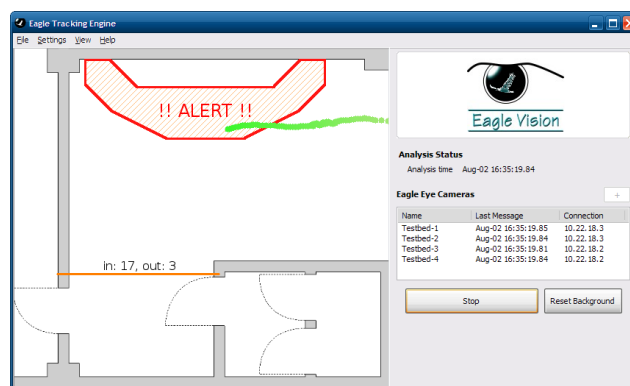
FIGURE 2.2: An example of an actuator. The GUI shows a person (green trace) has just entered a restricted area (red polygon). Furthermore, 4 cameras are connected and 20 people have crossed the orange line.

The Eagle Tracking Engine follows people, but that is of no value if nothing happens with the information. Each application developed by EVS needs an *actuator* to actually do something, but the types of actuator can vary, depending on the purpose of the systems. For instance: a counter that registers how many people cross a predefined line or and stores this in a database[1], an alarm that goes off whenever someone enters an area[2], a graphical user interface in which the track of every person is displayed for debugging purposes (figure 2.2 or a security system in which an Eagle Grid is used to detect the event of jumping over a security gate[3].

---

[1] At shopping mall *Kanaleneiland*, in Utrecht all incoming and outgoing customers are counted to gain insight in the logistics.

[2] This is currently used at Heineken to secure a machine with dangerous moving parts. The machine stops when someone comes too close.

[3] EVS is currently developing a jump-over-detector at the main office of an international bank in Amsterdam

2.2   Related work

In 2004, Zajdel et al. [1] described a multi-camera tracking system in which a colour based appearance model was used. They used monocular cameras that were mounted in such a way that they had a clear view of the person from head to toe. They manually selected a frame from the video sequence and extracted the person as a sub-image. The RGB colour pixels were converted to a colour channel *normalised space* [2] to suppress the effects introduced by the colour of illuminating light. Subsequently they calculated the mean colour of the lower, middle and upper body and used this as a representation.

In 2009, Englebienne et al. [3] adopted this method and applied it to a multi-camera tracking system that used stereo cameras instead of monocular ones. They assumed steady lighting and used the RGB values directly instead of converting them to a normalised space. They computed three mean RGB colours of each detected person, aimed at representing the colours of the lower, middle and upper part of the body. This is the reference that is most closely related to my work, because they used the same sensors that are used in this thesis. I build on their results by investigating six different *colour/geometry spaces* where they only discussed the RGB colour space.

Over the years many colour spaces have been proposed. Some of them are very straightforward and have been used by many, others are complex, unintuitive and have only been used in research settings. Gevers [4] describes 14 colour spaces of which four have been used in this thesis, two have been combined into one and that combination has been extended to form a novel colour/geometry space.

Chapter 3

Colour/geometry spaces

Whereas the human eye and brain are trained to observe the world as a collection of objects, a computer unfortunately is not. Humans use all kinds of methods to detect objects and determine their colour without even noticing they do it, but *segmentation* and *colour determination* are particularly hard for a computer due to the influence of lighting and shadows. When a surface contains only one colour, it may look like a whole band of different colours in RGB space, due to differences in luminance [4]. To illustrate the impact of the effect of illumination, a simple but powerful demonstration is provided in figure 3.1. It shows that the human brain automatically compensates for the fact that square B is covered by a shadow and therefore sees it as a lighter square. However, a computer will only see the RGB values of both squares, and figure 3.1(b) shows that these are equivalent.
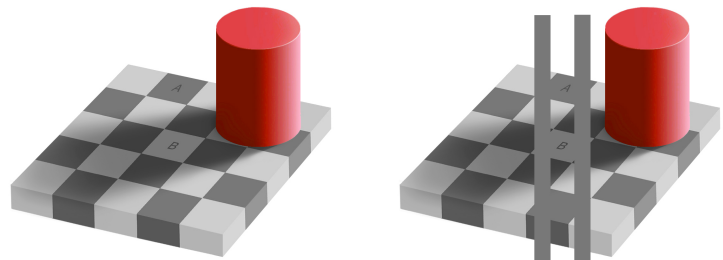
(a) The squares marked A and B seem to have a different colour because the human eye compensates for the effects of shadow.

(b) Two grey bars are drawn to show that both boxes are depicted with the same shade of grey.

FIGURE 3.1: Edward H. Adelsons "Checkerboard Illusion" [5].

This study aims at developing appearance models to represent people. An appearance model is considered to be a method to represent a person combined with a colour/geometry space in which it will be represented. This chapter will handle the colour/geometry spaces

one by one, explaining for each colour/geometry space how it was constructed and how it can be expected to behave.

3.1   RGB - Red, Green and Blue

FIGURE 3.2: RGB

Digital images are commonly recorded and stored as three channel image matrices. Each channel represents one of the three primary colours red, green and blue. For every pixel in the image, every channel has a value ranging from 0 to 1 depending on the amount of that primary colour that is in the pixel. Together, these channels represent the three-dimensional RGB space, see figure 3.2. In this colour space, a colour $\vec{c}$ is defined as an RGB row vector [R G B] (e.g. $\vec{c} = [1.0\ 0.6\ 0.0] = $ ⬤) and the difference between two colours $\vec{p}$ and $\vec{q}$ is defined as the Euclidean distance:
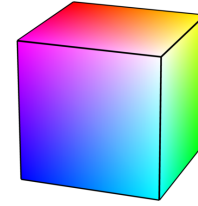
$$\Delta^{\mathsf{E}}(\vec{p}, \vec{q}) = \sqrt{(\vec{q} - \vec{p})(\vec{q} - \vec{p})^{\top}} \tag{3.1}$$

Figure 3.3 shows a picture of a person and the decomposition in the red, green and blue channels. The channels are represented on a greyscale where a lighter shade of grey stands for a greater amount of the corresponding colour. The distribution of the pixels projected on the blue-red plane is shown in figure 3.3(e). It shows that the lighting has a great influence on the RGB value: even though the jeans and sweater in the picture have only one colour, they have a great number of different RGB values in different lighting. That is why the RGB space causes problems in most *colour models* and the colour of an object is not conveniently represented in this colour space. It is included in this study because it is the original colour space in which the images are recorded. It is merely used as a base line, but I do not expect good results of the RGB space.

(a) RGB   (b) Red   (c) Green   (d) Blue   (e) blue-red distribution (4000 samples)
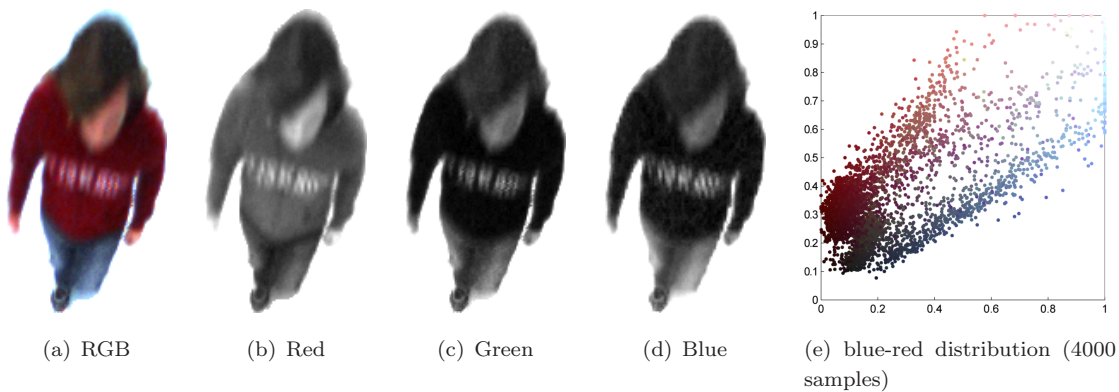
FIGURE 3.3: RGB colour space

3.2 rgb - Normalised Red, Green, Blue

To eliminate the effect of luminosity on the perceived colour, the intensity can be removed from the RGB values to create a *chromaticity space* in which only the colour tone is preserved without the black or white factor. This chromaticity space is denoted as rgb, in lower case. To convert a colour $\vec{c}$ from RGB to rgb, each channel is divided by the intensity of the colour, which is defined as the sum of the red, green and blue channels (equation 3.2 [4]).
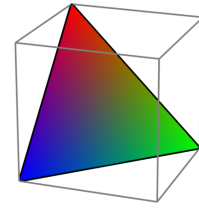
$$\vec{c} := \begin{cases} \frac{\vec{c}}{c_R + c_G + c_B} & \text{if } c_R + c_G + c_B > 0 \\ \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} & \text{if } c_R + c_G + c_B = 0 \end{cases} \tag{3.2}$$

Figure 3.5 shows the decomposition of the red, green and blue channels and the distribution when the pixels are projected on the blue-red plane. Because all colours are divided by the sum of the R, G and B values, the sum of the elements in the resulting rgb colour $\vec{c}$ is always 1. This means that each channel can be calculated if the other two are known (e.g. $c_b = 1 - c_r - c_g$), thus one of the channels can be removed to reduce the dimensionality. This is how this colour space is implemented in most applications and therefore this implementation is also adopted in this study: the blue channel is removed.

As is shown in figure 3.4, the normalised rgb space is in fact an *equilateral triangle* (i.e. triangle with equal sides), whereas projecting the entire space on the red-green plane results in an *isosceles right triangle* (i.e. two sides are equal and one corner is 90°). This means that the space is skewed and that the colours are not correctly distributed. Therefore, using a Euclidean distance in this space is incorrect. However, many researchers have done so, and the Euclidean distance gives a simple estimate of a correct distance measure. Therefore it will also be used here.

(a) rgb   (b) red   (c) green   (d) blue   (e) blue-red distribution (1000 samples)
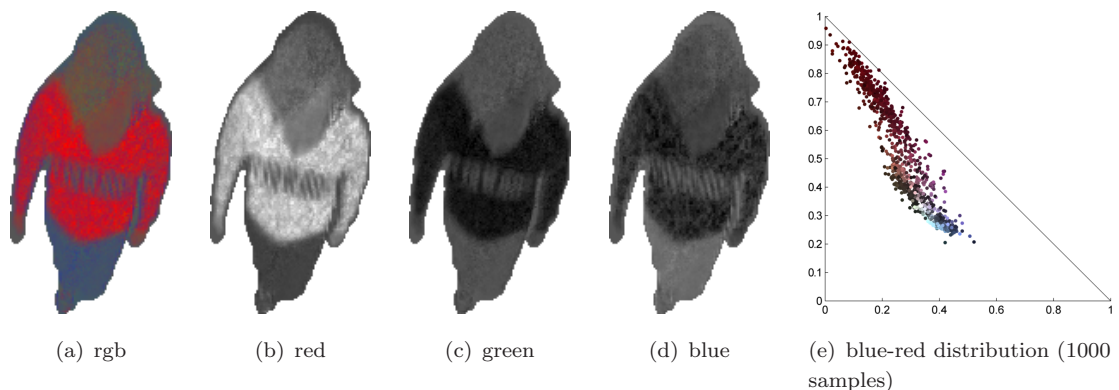
FIGURE 3.5: Normalised rgb colour space

3.3   HSV - Hue, Saturation and Value

FIGURE 3.6: HSV

The previous two sections showed that the RGB and rgb space both have their shortcomings. Also, both spaces are not very natural. A human would describe colours like 'bright orange' instead of 'red with a bit of green', and 'greyish blue' instead of 'some blue with a bit of red and green'. Because we are mimicking human behaviour it might be better to use a colour space that approximates our own observations and classifications.
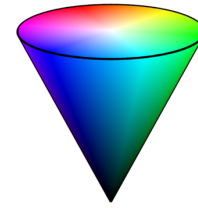
A widely used colour space that does this, is the hue-saturation-value space, or HSV. In this space the colour is defined by the *hue*, the amount of colour is represented by the *saturation* and the brightness of the colour is represented by the *value* component. The equations to calculate the hue, saturation and value are provided and discussed in [4]. See figure 3.7 for the decomposition of the hue, saturation and value channels and the distribution when the pixels are projected on the hue-saturation plane and on the hue-value plane.

(a) hue   (b) saturation   (c) value   (d) hue-saturation distribution (hue on the x-axis)   (e) hue-value distribution (hue on the x-axis)
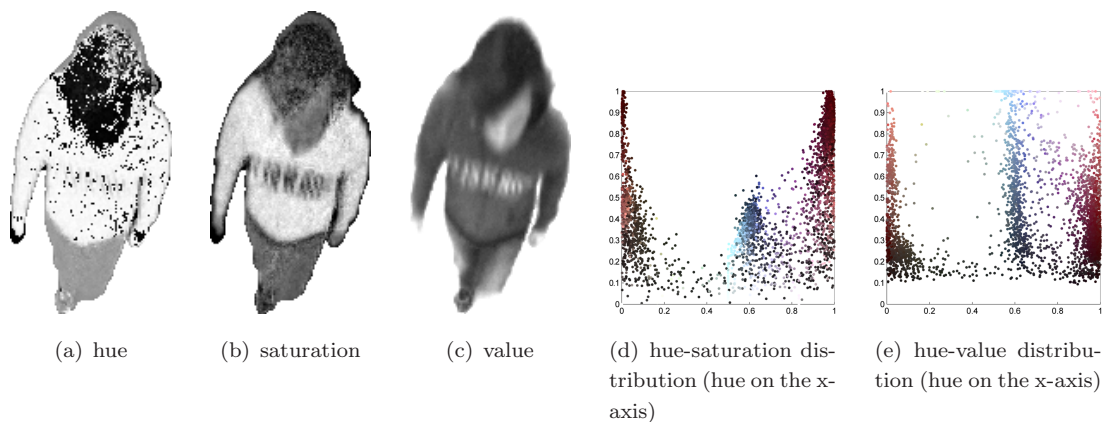
FIGURE 3.7: HSV colour space

As the hue dimension is circular, some issues arise around 0 and 1, i.e. the colour red. When the red is a bit more purple, the hue is around 1; when it is more orange, the hue is around 0. Another effect of this circularity can be seen in figure 3.7(d), there are much more pixels in the saturated area than in the less saturated area. The next section will elaborate a bit more on this circularity.

Like in the RGB space, an object with one colour only may look like a whole band of different colours in HSV space. However, this band ranges from dark colours to light colours. This means that they differ in the value dimension but not so much in the hue or

saturation dimension, meaning that the noise occurs mainly in one dimension. This can be seen in figure 3.7(e), where there are clear vertical clusters.

3.4 h - Hue

FIGURE 3.8: hue

The hue dimension of the HSV space is interesting because it is a continuous colour dimension describing only what the *pure colour* (i.e. without any white or black) of an object is. Because I am building a colour based appearance model it might be satisfying to use only these pure colours in my representation. Intuitively, this means that the colour of an object is described to be blue or red or any other colour of the rainbow without specifying how light, dark or greyish that colour is. Ranging the hue from 0 to 1, the consecutive primary and secondary colours are: red, yellow, green, cyan, blue, magenta and again red, see figure 3.8.

(a) Hue in colours

(b) Hue in greyscale

(c) Hue distribution (1000 samples, jitter: $\mathcal{N}(0, 0.1)$)
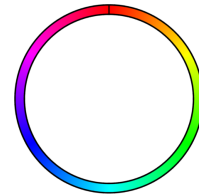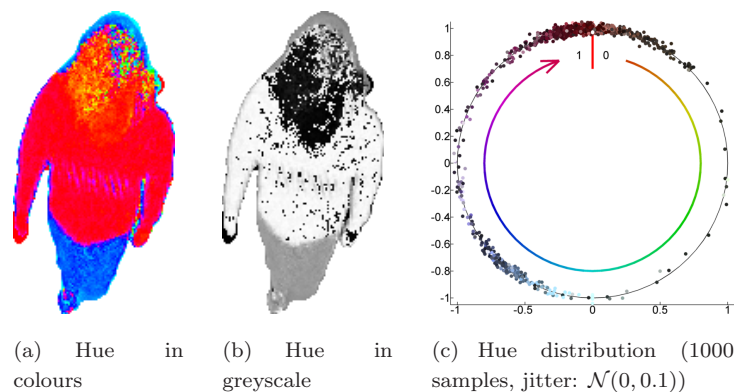
FIGURE 3.9: Hue colour space

As described in the previous section, and as can be seen in the hue value of the sweater in figure 3.9(b), some problems arise due to the circularity of the hue space. There is a turning point at 1, so problem occur especially when we are dealing with colours around red. I will elaborate on this a bit more.

For example, what happens when we want to calculate the mean colour of a dataset of n pixels in this circular hue space? The dataset consist of two colours ($n = 2$): $c_1 = 0.1 =$ ● and $c_2 = 0.8 =$ ●, respectively orange and purple. Calculating the mean as if the hue is

a linear space would result in:

$$\bar{c} = \frac{1}{n} \sum_{i=1}^{n} c_i \qquad (3.3)$$

$$0.45 = \text{turquoise } (\bullet).$$

It is easy to see that turquoise can not be the mean colour of orange and purple (●●). Although the hue is a one-dimensional space and a colour is defined as a scalar $0 \leq h < 1$, it can be converted to a *unit vector* $\vec{c}$ of two components a and b as follows:

$$\vec{c} = [a \ b] = [\sin(2\pi h), \cos(2\pi h)]. \qquad (3.4)$$

Now it is possible to calculate the mean of these vectors by simply dividing their sum by their number. Then, I use the arctan2 function to calculate the hue of the resulting vector. See appendix A for an explanation of the arctan2 function for vectors. For now, it can be seen as a way to go back from unit vectors to the scalar representation of the hue. Using this method, the mean colour of $\vec{c}_1$ and $\vec{c}_2$ can now be calculated as:

$$\bar{c} = \text{arctan2} \left(\frac{1}{n} \sum_{i=1}^{n} \vec{c}_i \right) \qquad (3.5)$$

$$\approx 0.95 = \text{pinkish red } (\bullet).$$

And indeed this looks more likely to be the mean of purple and orange than turquoise did (●●). This method is a good estimation when there are a lot of data points, but it is not completely correct. For instance, the mean of a dataset $C = \{0.1, 0.1, 0.4\}$ should be 0.2, but with the method of equation 3.4 this becomes ≈ 0.1815.

As a solution, both methods are combined in this study. Firstly the mean hue is estimated using equation 3.5. Then the opposite point on the hue circle is calculated as $p = \bar{c} + 0.5$ modulo 1. The colours on the hue circle are now shifted such that the 0 to 1 transition is exactly opposite to the estimated mean. A set of temporary colours C' is defined as $c'_i = c_i - p$ modulo 1. Now a temporary mean colour $\bar{c}'$ is calculated using equation 3.3. And finally the improved mean $\bar{c}^*$ is calculated by shifting $\bar{c}'$ back to the right position: $\bar{c}^* = \bar{c}' + p$ modulo 1.

Let me apply this method to the previous example:

$$
\begin{aligned}
C &= \{0.1, 0.1, 0.4\}, \\
\bar{c} &\approx 0.18, & \text{(eq. 3.5)} \\
p &= 0.68, \\
C' &= \{0.42, 0.42, 0.72\}, \\
\bar{c}' &= 0.52, & \text{(eq. 3.3)} \\
\bar{c}^* &= 0.2.
\end{aligned}
$$

Note that problems will still arise when the sum of the unit vectors results in [0 0]. However, chances are small that this will happen in practical applications with many data points, but if it does, the mean colour will be chosen to be $\bar{c} = 0$ (pure red) by convention.

3.5 hs - Hue and Saturation

FIGURE 3.10: hs

Describing only the tone of a colour (i.e. its hue value) introduces instability problems when the original colour approaches grey. This is caused by the fact that similar shades of grey may vary greatly in hue, as can be seen in the hair of the person in figure 3.9(a). The same holds for white and black, as they are merely special cases of grey. To overcome this effect I include the saturation in the colour space and construct a hue-saturation space. Compare figure 3.10 to figure 3.8 to see the difference between both colour spaces.
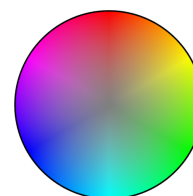
The visual colour effect of a change in the hue dimension varies greatly depending on the saturation of a colour. For example hs(0, 0.1) and hs(0.33, 0.1) are grey with a hint of red and green respectively (⬤), whereas hs(0, 1.0) and hs(0.33, 1.0) are bright red and bright green (⬤). Using the Euclidean distance to measure the difference between two colours in the Cartesian hue-saturation space would make no sense. In this example, the Euclidean distance between the colours is 0.33 for both pairs, while the difference in colour between the first two colours is much smaller than between the last two.

Therefore I will add the saturation to the hue colour space described in the previous section to define a new two-dimensional colour space. For this purpose I convert every colour to a vector $\vec{c}$ of two components a and b, based on the hue h and the saturation s, in a way similar to the decomposition given in equation 3.4:

$$
\vec{c} = [a \ b] = [s \cdot \sin(2\pi h), s \cdot \cos(2\pi h)], \tag{3.6}
$$

This decomposition leads to the colour space depicted in figure 3.10. In this new Cartesian space the distance between colours correspond to the difference between them. If the concept of colour difference can be modelled, then this is probably the best way to do so.

Figure 3.11 shows the decomposition of an image into the two components corresponding to the dimensions of the colour space. Figure 3.11(d) shows a plot of the data points in the newly defined space. Not only are the four clusters of the jeans, the sweater, the face and the hair clearly distinguishable, but their distribution also approximates a normal distribution. This indicates that the colour space gives a good representation of the colour differences. It is also very promising because in one of the methods I use, I will use a *mixture of Gaussians* to find clusters of colours. And a mixture of Gaussians assumes normally distributed clusters of data.

(a) hue & sat.   (b) $s \cdot \sin(2\pi h)$ (c) $s \cdot \cos(2\pi h)$ (d) hue-saturation distribution (2000 samples)
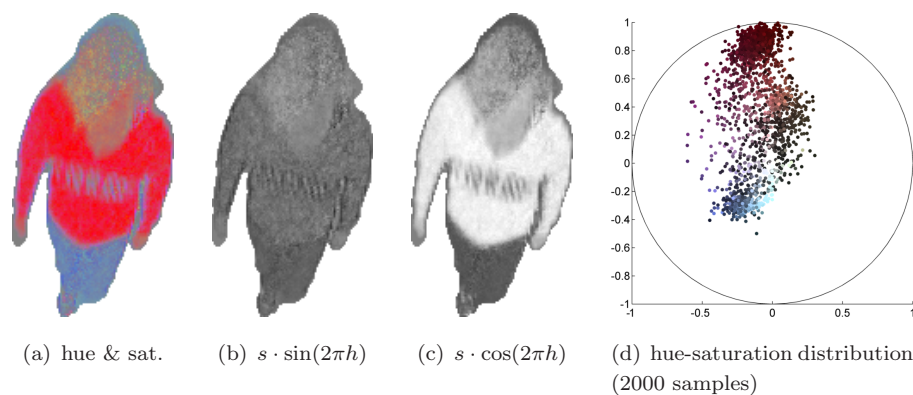
FIGURE 3.11: Hue-saturation colour space

3.6   hsz - Hue, Saturation and Height

FIGURE 3.12: hsz

So far, I completely ignored the fact that I am dealing with point clouds instead of images. To understand why this information could be useful we have to take a step ahead. The goal of this study is to create a representation of a person in the real world based on the distribution of colours over the pixels belonging to this person (e.g. using a *kernel method* or a *histogram*). For such methods, it might be useful to combine some of the *geometric information* the stereo cameras collect for each pixel with the *colour information* collected from the image. Therefore the last space is a hybrid Cartesian space containing both colour information and geometric information in one space. This is bit tricky because two different domains are used in one coordinate system, both with different measures and scales. In order to acquire a consistent unit of
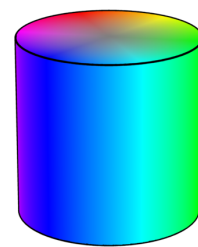
measure, I will normalise the values in each dimension to range from -1 to 1. In the next paragraph I will explain how I do this.

The two-dimensional colour component of this three-dimensional space is $\vec{c}$ as defined in the previous section, equation 3.6. It is a vector $\vec{c} = [a\ b]$ which is a combination of hue and saturation. The components of this vector are already between -1 and 1 by definition. The geometric component z' is a scalar that represents the *normalised height* of a data point. The original height z of a data point is its distance to the ground plane, which is inversely proportional to its distance to the camera. It is normalised by scaling with the minimum and maximum height[1], $z_\perp$ and $z^\top$ to form the *normalised height* $z' = 2\frac{z-z_\perp}{z^\top - z_\perp} - 1$, with range $[-1, 1]$, just like a and b. Now I define a data point $\vec{p}$ as $\vec{p} = [\vec{c}\ z'] = [a\ b\ z']$. This defines a cylindrical colour/geometry space in a Cartesian coordinate system where all dimensions range from -1 to 1. This new space, incorporating both colour and height, is shown in figure 3.12. Figure 3.13 shows that adding the height of the data points gives a lot of extra information that is very useful when detecting clusters in the data.

(a) Hue   (b) Hue & saturation   (c) Hue, saturation & height
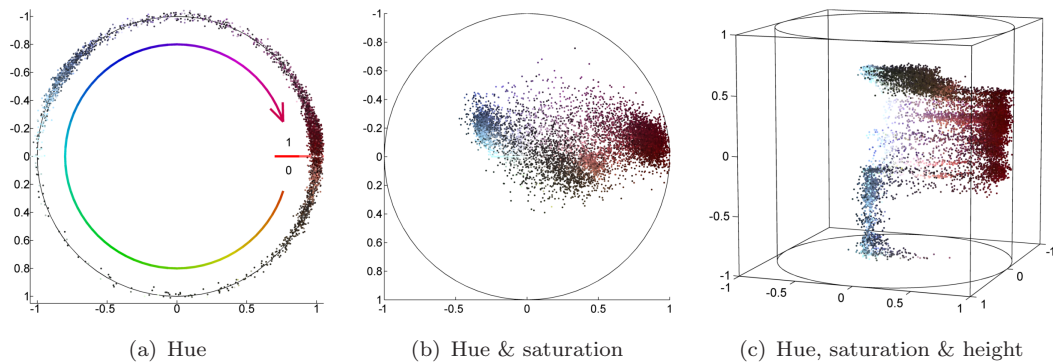
FIGURE 3.13: Hue, saturation and height colour/geometry space.

---

[1]The minimum and maximum height, $z_\perp$ and $z^\top$, are predefined parameters, typically 0.2 metres and 2.2 metres, to remove noise from the ground plane and to make sure most people fit in.

Chapter 4

Appearance Models

In this chapter, I will discuss the appearance models that I use to represent a person in the real world. It is essential for inter-camera tracking that these models are reliable. They are used by the Eagle Tracking Engine to assign people detected at the scene to a number of real-world individuals. As described in chapter 2, the input for these models is a point cloud of the scene, together with the position of each person detected in the scene. A point cloud corresponding to the detected person is obtained by extracting a cylinder from the point cloud of the scene at the location of the detection. This cylinder is chosen to be 60 centimetres in diameter and 2 metres high. The output of the appearance models is a representation of the detected person. This representation can then be matched to the representations of the different tracks to see whether the detected person belongs to one of these tracks. There are numerous methods to convert such a cylindrical point cloud to a representation. Six of them are investigated in the remainder of this chapter. Each model will be applied to all six colour spaces described in the previous chapter.

I start out with an appearance model based on histograms, which I will use as a base line. All the other models are based on calculating mean colours. Some of them use fixed partitions of the point cloud, while others try to automatically find cluster centres within the distribution of the data points.

The investigated models are Histograms ($\mathcal{M}^{\boxplus}$), Single mean ($\mathcal{M}^{\cdot}$), Disks ($\mathcal{M}^{\bigcirc}$), Rings ($\mathcal{M}^{\circledcirc}$), k-Means ($\mathcal{M}^{\cdot\cdot}$) and Mixture of Gaussians ($\mathcal{M}^{\odot}$).

Finally the Mixture of Gaussians model will be extended in two different ways. These extended models are the Semi correlated mixture of Gaussians ($\mathcal{M}^{\circledast}$) and the Semi correlated mixture of Gaussians with separation planes ($\mathcal{M}^{\ominus}$).

4.1   Histograms

A colour histogram is a discretized representation of the colour distribution in an image
or point cloud. The axes of a colour space are divided such that the entire space is cut up
into equally sized pieces, the so called bins. The pixels are distributed over the histogram
and their colour determines in which bin they belong. Finally the number of pixels in each
bin is counted and those numbers form the representation.

Histograms are a relatively easy method to capture complex distributions. That is why
they are widely used in recognition tasks. A downside of using histograms is that the
feature vector of the resulting representation is relatively long compared to other methods.
As stated before, the histograms will serve as a baseline for the other appearance models. I
expect the newly designed models to perform at least as good, with smaller feature vectors.

For a Cartesian space like RGB and normalised rgb, the histogram is easily defined by
cutting up each dimension in b intervals, resulting in b^m bins, where m is the number of
dimensions (see figure 4.1(a) and 4.1(b)).

(a) Conventional 3D RGB
histogram with b^3 bins

(b) Conventional 2D rgb his-
togram with b^2 bins
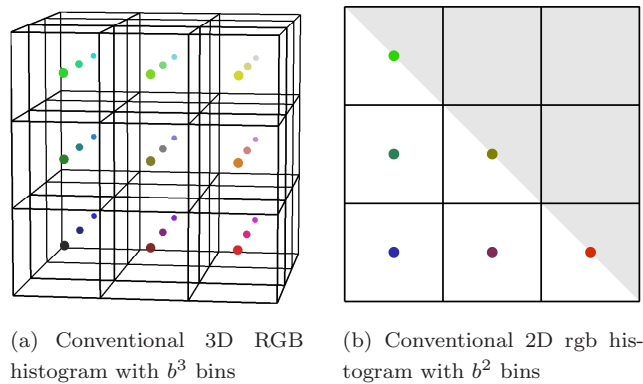
FIGURE 4.1: A conventional RGB and normalised rgb histogram. Note that almost half
of the bins of the normalised rgb histogram is unused.

Although this is not the right way to split up the normalised rgb space (recall section
3.2), in most literature it is done this way. I followed these researchers and created the
normalised rgb histograms like figure 4.1(b). I did however remove the completely empty
bins from the histogram, so the normalised rgb histograms in this paper contain $b(b+1)/2$
bins instead of b^2. I present an alternative way to uniformly divide the normalised rgb
space in figure 4.2.

(a) rgb plane in RGB space   (b) rgb plane extracted   (c) Improved 2D rgb histogram with b^2 bins
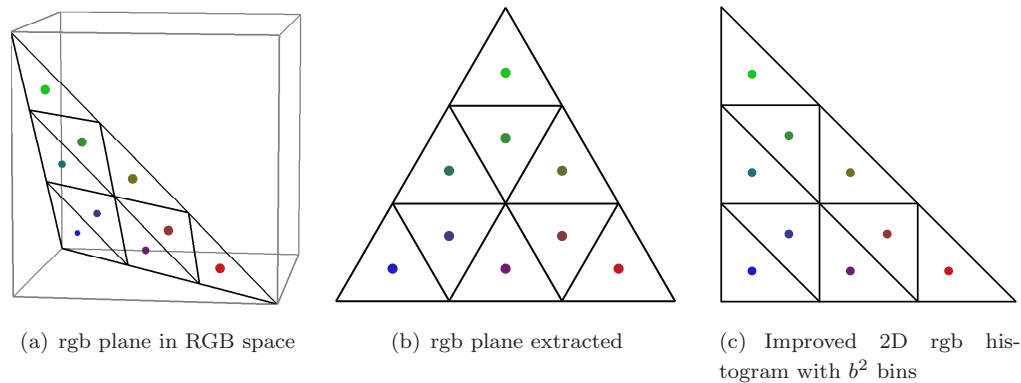
FIGURE 4.2: The construction of a correctly distributed rgb histogram, using the simple projection performed by neglecting 1 dimension (blue).

If the radial spaces would be divided in the same way as the Cartesian spaces are, this results in bins that differ greatly in size and an unequal distribution of the colours. Refer to figure 4.3(a) to see that the inner three bins contain almost the same tone of grey, while the outer three bins each contain a whole range of colours (i.e. the 'red' bin contains all colours from magenta, via red and orange to yellow). In other words, when the saturation is 0, the hue has no effect, so the range of hue there should be 0 and when the saturation is 1, the hue has great effect and should range from 0 to $2\pi \approx 6.28$, because of its circularity (recall figure 3.10). Thus, on average the hue should range from 0 to $\pi \approx 3.14 \approx 3$, therefore we might want to divide the hue dimension in $3 \times b$ bins instead of just b as in figure 4.3(b).

(a) Conventional histogram containing b^2 bins   (b) Alternative histogram containing $3 \times b^2$ bins   (c) Improved histogram containing $3 \times b^2$ equally sized bins
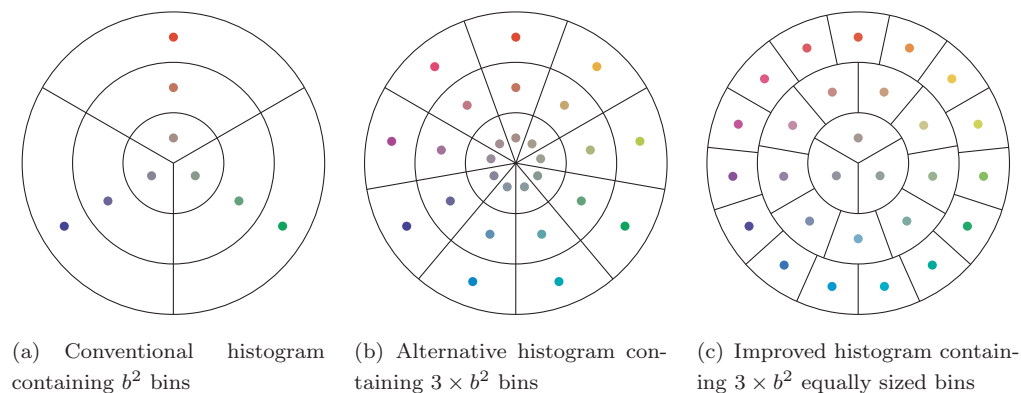
FIGURE 4.3: The first two images show how hue-saturation histograms are commonly constructed, the last images shows the histogram constructed for this study.

The outer ring clearly shows that much more colours are covered by the histogram now, but still the distribution is not optimal. Indeed, the bins in the centre contain even more similar tones of grey. That is why an alternative method was developed in this study to

make a proper hue-saturation histogram in which all bins are square-like and of equal size. The first ring (i.e. the inner circle) is divided in 3 bins, the next in 9 bins, then 15 bins and so on; every ring contains 6 bins more than the previous one. This results in $3b^2$ equally sized bins where b is the number of rings. I searched extensively for someone who has used this type of histograms before, but I could not find a single resource. It is based on the fact that the difference of squares is linear (i.e. $b^2 - (b-1)^2 = 2b - 1$). The proof that the bins are equally sized, is given in appendix B.

This radial histogram of figure 4.3(c) extends easily to the cylindrical and conical histograms in figure 4.4(a) and 4.4(b). Refer to figure 3.6 to see that the variance in colours decreases when the value decreases. Therefore the number of bins is lower in the darker region of the HSV space.

(a) Cylindrical hsz histogram   (b) Conical hsv histogram
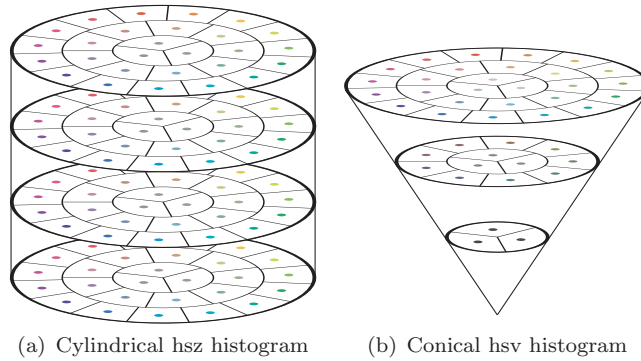
FIGURE 4.4: Two extensions of the radial hue-saturation histogram.

To compare two histograms $\mathcal{M}^{\boxplus}$ with k bins we need a distance measure. A histogram can be represented as a vector with one element ω_i for each bin, containing the number of pixels n_i in that bin divided by the total number of pixels $n = |\mathbf{X}|$ in a person's cylindrical point cloud $\mathbf{X}$. Many people use the Euclidean distance to compare two such vectors, however the Euclidean distance is proven to perform badly in high-dimensional spaces [6]. Therefore we simply use the total absolute difference Δ^{A}:

$$\mathcal{M}^{\boxplus} = \{\omega_1, \ldots, \omega_k\} \tag{4.1}$$

$$\omega_i = \frac{n_i}{n} \tag{4.2}$$

$$\Delta^{\boxplus}(\mathcal{M}_a^{\boxplus}, \mathcal{M}_b^{\boxplus}) = \Delta^{\mathsf{A}}([\omega_{a1} \ \ldots \ \omega_{ak}], [\omega_{b1} \ \ldots \ \omega_{bk}]) \tag{4.3}$$

$$\Delta^{\mathsf{A}}(\vec{p}, \vec{q}) = \sum_{i=1}^{k} \|p_i - q_i\| \tag{4.4}$$

4.2   Single mean

This is the first and most simple appearance model based on averaging colours, $\mathcal{M}^{\cdot}$. The mean colour $\vec{\mu}$ of all n pixels $\vec{x}_i$ in a person's point cloud $\mathbf{X}$ is calculated and the distance between two models is defined by the Euclidean distance.

$$\mathcal{M}^{\cdot} = \{\vec{\mu}\} \tag{4.5}$$

$$\vec{\mu} = \frac{1}{n}\sum_{i=1}^{n}\vec{x}_i \tag{4.6}$$

$$\Delta^{\cdot}(\mathcal{M}^{\cdot}_a, \mathcal{M}^{\cdot}_b) = \Delta^{\mathsf{E}}(\vec{\mu}_a, \vec{\mu}_b) \tag{4.7}$$

$$\Delta^{\mathsf{E}}(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{m}(q_i - p_i)^2} = \sqrt{(\vec{q}-\vec{p})(\vec{q}-\vec{p})^{\top}} \tag{4.8}$$

4.3   Disks

Instead of calculating the overall mean colour, we can also calculate the mean colours of different parts of the body. One heuristic is to divide the point clouds into three disks, or layers [1] (figure 4.5(a)). Where Zajdel et al. used three layers, we will use a variable number of disks, denoted by k. If the height of a person is $\hat{z} = \max_{i=1}^{n} z$, then the range of a layer $\mathbf{Y}_i$ is defined as $\left[\frac{2i-1}{2k+2}\hat{z}, \frac{2i+1}{2k+2}\hat{z}\right)$, see figure 4.5(c).

The portion of pixels in a single disk $\mathbf{Y}_i$ is denoted by $\omega_i = \frac{|\mathbf{Y}_i|}{|\mathbf{Y}|}$, where $|\mathbf{Y}|$ is the total number of pixels in all disks; this value is used as a weight when calculating distances. Note that the number of pixels in the disks is smaller than the total number of pixels in a person's point cloud, $|\mathbf{Y}| < |\mathbf{X}|$. That is because, following Zajdel et al., the upper and lower $\frac{1}{2k}$ part of the point cloud are not used for the model. Their reason was that the lower part of the point cloud contains only a person's shoes (and noise from the floor in our case), while the upper part contains only pixels from the face and the hair of the person. They were designing an appearance model that captures the colours of a person's clothes and therefore they excluded the upper and lower part.

$$\mathcal{M}^{\bigcirc} = \{\omega_1, \ldots, \omega_k, \vec{\mu}_1, \ldots, \vec{\mu}_k\} \tag{4.9}$$

$$\Delta^{\bigcirc}(\mathcal{M}_a^{\bigcirc}, \mathcal{M}_b^{\bigcirc}) = \sum_{i=1}^{k} \frac{\omega_{ai}\omega_{bi}}{\sum_{j=1}^{k} \omega_{aj}\omega_{bj}} \Delta^{\mathsf{E}}(\vec{\mu}_{ai}, \vec{\mu}_{bi}) \tag{4.10}$$

$$\omega_i = \frac{|\mathbf{Y}_i|}{\sum_{j=1}^{k} |\mathbf{Y}_j|} \tag{4.11}$$

(a) Image of the original idea by Zajdel et al. used in [1] and [3]

(b) The extracted point cloud of one person with separation planes.

(c) The representation. The colour of the disks is the mean colour μ_i of the pixels in that area, the height denotes the number of pixels $|\mathbf{Y}_i|$.
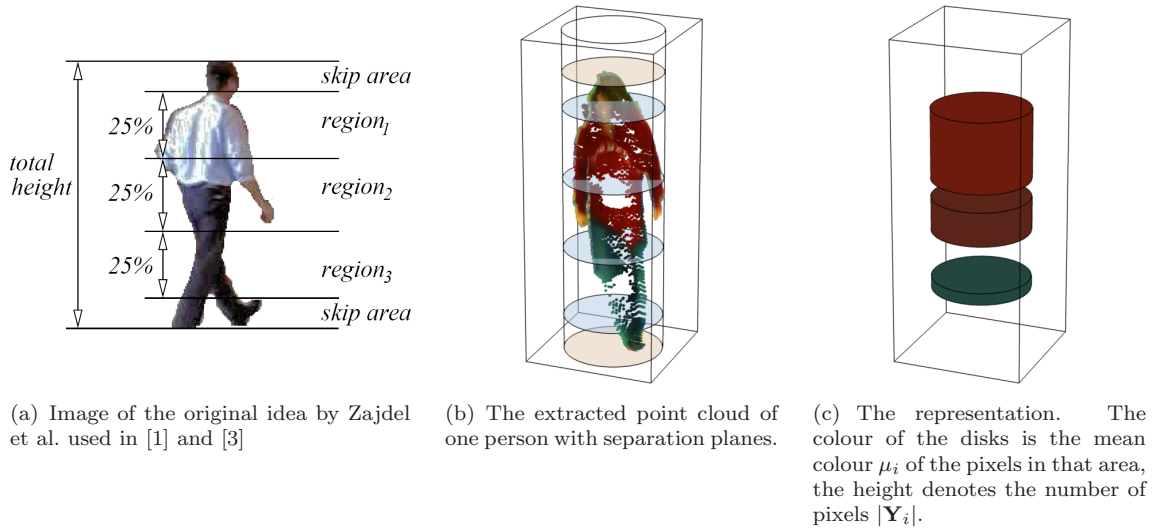
FIGURE 4.5: Interpretation of Zajdel et al. [1]. The colours of the point cloud have been manually adjusted for visualisation purposes.

Notice that using the hsz space in this model is similar to using the hs space, because the height of the pixels is embedded in both the appearance model and the hsz space. I did however include the hsz space in the experiments with this model, and indeed it shows similar results as the hs space.

4.4   Rings

I have extended this idea of the disks model to create a rings model, $\mathcal{M}^{\circledcirc}$. Each disk is divided into several rings of different radii. This implicitly brings more information about a person's shape into the model (see figure 4.6(b)). This extension makes it possible to capture different colours at the edge of the virtual cylinder than in the centre, which allows modelling concepts like bare arms or a shirt with an open jacket.

A difference with the previous appearance model is that the height of the rings model was fixed at $z^\top = 2$ metres and that the entire cylindrical point cloud was used. The parameters are similar to those of the disks model, k denotes the total number of rings:

$$\mathcal{M}^\circledcirc = \{\omega_1, \ldots, \omega_k, \vec{\mu}_1, \ldots, \vec{\mu}_k\} \tag{4.12}$$

$$\Delta^\circledcirc(\mathcal{M}_a^\circledcirc, \mathcal{M}_b^\circledcirc) = \sum_{i=1}^{k} \frac{\omega_{ai}\omega_{bi}}{\sum_{j=1}^{k} \omega_{aj}\omega_{bj}} \Delta^\mathsf{E}(\vec{\mu}_{ai}, \vec{\mu}_{bi}) \tag{4.13}$$

$$\omega_i = \frac{|\mathbf{X}_i|}{|\mathbf{X}|} \tag{4.14}$$
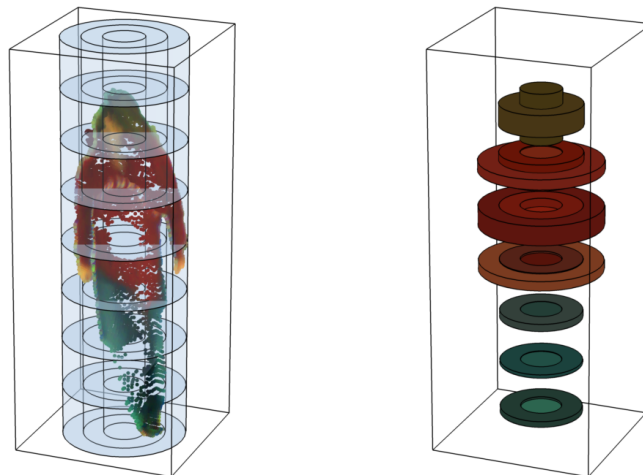
(a) Extension of the disks model that includes separation by radius.

(b) The height of the rings implicitly describe a person's shape.

FIGURE 4.6: Extension of Zajdel et al. [1]. The colours of the point cloud have been manually adjusted for visualisation purposes.

4.5 k-Means

Another way to calculate multiple mean colours is the classic k-means algorithm. Common human appearances contain around four separate colours; the shirt, the trousers, the hair and the skin, thus $k \approx 4$.

k-means is a well known classic method and therefore I will describe it only briefly. Please read Bishop [7] for more information. The k-means algorithm starts by randomly picking k data points and uses these as cluster centres. Each data point is assigned to the nearest

cluster centre, using the Euclidean distance Δ^{E} (equation 4.8). k-Means basically divides the data space into k partitions separated by hyper planes, the mid-planes between the cluster centres (see figure 4.7(a)). After this assignment, or *classification*, the means of the data points in each cluster are calculated and used as the new cluster centres. By repeating this process k-means will converge. See algorithm 1 for the pseudo code.

(a) k-Means   (b) Mixture of Gaussians
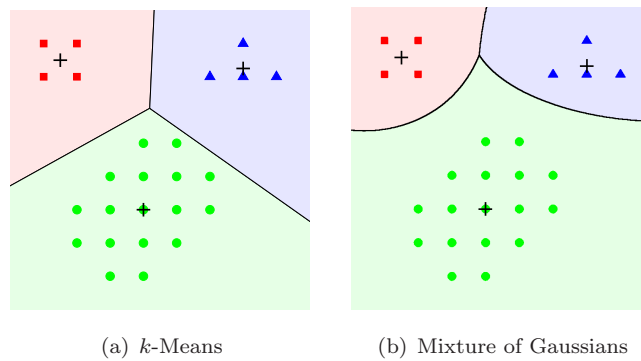
FIGURE 4.7: Separation of clusters using k-means and a mixture of Gaussians.

Algorithm 1 k-Means

Input: A pixel set $\mathbf{X}$ with n pixel vectors x_i of length m.
Output: A matrix M containing k means $\vec{\mu}_i$.

1:                     ▷ Initialisation
2:   $M \leftarrow \textsc{randomSamples}(\mathbf{X}, k)$
3: **while** $\mathcal{M}^{\cdot\cdot}$ is not converged **do**
4:                     ▷ Expectation
5:    **for** $c = 1 \rightarrow k$ **do**
6:     $\mathbf{C}_c \leftarrow \varnothing$
7:    **end for**
8:    **for** $i \in \{1, \ldots, n\}$ **do**
9:     $c \leftarrow \operatorname{argmin}_{j=1}^{k}(\vec{x}_i - \vec{\mu}_j)(\vec{x}_i - \vec{\mu}_j)^{\top}$
10:    $\mathbf{C}_c \leftarrow \mathbf{C}_c \cup \{x_i\}$
11:    **end for**
12:                    ▷ Maximisation
13:    **for** $c = 1 \rightarrow k$ **do**
14:     $\vec{\mu} \leftarrow \textsc{mean}(\mathbf{C}_c)$
15:    **end for**
16: **end while**

The resulting parameters of the k-means model $\mathcal{M}^{..}$ look similar to those of the disks and rings models:

$$\mathcal{M}^{..} = \{\omega_1, \ldots, \omega_k, \vec{\mu}_1, \ldots, \vec{\mu}_k\} \tag{4.15}$$

$$\Delta^{..}(\mathcal{M}_a^{..}, \mathcal{M}_b^{..}) = \sum_{i=1}^{k} \frac{\omega_{ai}\omega_{bi}}{\sum_{j=1}^{k} \omega_{aj}\omega_{bj}} \Delta^{\mathsf{E}}(\vec{\mu}_{ai}, \vec{\mu}_{bi}) \tag{4.16}$$

$$\omega_i = \frac{|\mathbf{X}_i|}{|\mathbf{X}|} \tag{4.17}$$

One issue with k-means is that the outcome of the algorithm can depend on the initialisation. The initial centres are chosen by random sampling from the dataset for simplicity reasons. Nevertheless there do exist better initialisation methods like k-means++ [8], see algorithm 2 in Appendix C.

4.5.1 Remarks on the distance measure

The distance measure $\Delta^{..}$ in equation 4.16 is a bit to simplistic; the means can be sorted in any order, resulting in an unreliable and meaningless distance function. Therefore the means of two models have to be aligned. One method to do this is by calculating the distance of all permutations of clusters and choose the smallest. This is $O(k!)$, however k is the number of kernels and will be small, typically around 4.

4.6 Mixture of Gaussians

Whereas k-means assumes the separation planes to lie exactly in between the cluster centres, figure 4.7 shows that this separation is not necessarily optimal. For instance, the separation boundary between the green circles and the blue triangles is much closer to the circles. Clusters of data points are not generally of equal size, nor do they have equal variance in all directions. Therefore, it might be better to use a mixture of Gaussians to model these clusters.

A mixture of Gaussians, $\mathcal{M}^{\odot}$, can be fitted on the data by applying an Expectation Maximisation (EM) algorithm [7]. It is similar to k-means, with the extension that each data point $\vec{x}_j$ is assigned to each cluster centre $\vec{\mu}_i$ with a certain probability $p_{ij} = \omega_i \cdot \mathcal{N}(\vec{\mu}_i, \Sigma_i, \vec{x}_j)$, see equation 4.18. instead of being assigned to one cluster centre only. Therefore each cluster has an extra parameter additional to the mean $\vec{\mu}_i$; the *covariance matrix* Σ_i. I would like to refer you to Bishop [7] for further details.

$$\mathcal{N}(\vec{\mu}, \Sigma, \vec{x}) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} \exp\left(\frac{1}{2}(\vec{q} - \vec{p})\Sigma^{-1}(\vec{q} - \vec{p})^\top\right) \tag{4.18}$$

The probability p_{ij} depends on the cluster's *prior* and the distance from the data point $\vec{x}_j$ to the cluster centre $\vec{\mu}_i$.

This distance is no longer the Euclidean distance, but the Mahalanobis [9] distance $\Delta_\Sigma^{\mathsf{M}}$ (equation 4.19). Notice that the Euclidean distance Δ^{E} is a special case of the Mahalanobis distance, i.e. $\Sigma = I$.

$$\Delta_\Sigma^{\mathsf{M}}(\vec{p}, \vec{q}) = \sqrt{(\vec{q} - \vec{p})\Sigma^{-1}(\vec{q} - \vec{p})^\top} \tag{4.19}$$

Calculating the distance between two *Gaussian distributions* is far from trivial. The Kullback-Leibler divergence [10] is a non-symmetric measure that defines how much a probability distribution $\mathcal{P}$ differs from another probability distribution $\mathcal{Q}$ (in that order). Its general formula is shown in equation 4.23 the formula for two m-dimensional Gaussian distributions is shown in equation 4.24.

$$\mathcal{M}^\odot = \{\omega_1, \ldots, \omega_k, \vec{\mu}_1, \ldots, \vec{\mu}_k, \Sigma_1, \ldots, \Sigma_k\} \tag{4.20}$$

$$\Sigma_i = \begin{bmatrix} \sigma_{i,1,1} & \cdots & \sigma_{i,1,m} \\ \vdots & \ddots & \vdots \\ \sigma_{i,m,1} & \cdots & \sigma_{i,m,m} \end{bmatrix} \tag{4.21}$$

$$\Delta^\odot\left(\mathcal{M}_a^\odot, \mathcal{M}_b^\odot\right) = \sum_{i=1}^{k} \frac{\omega_{ai}\omega_{bi}}{\sum_{j=1}^{k} \omega_{aj}\omega_{bj}} \Delta^{\mathsf{KL}}\left(\mathcal{M}_b^\odot \| \mathcal{M}_a^\odot\right) \tag{4.22}$$

$$\Delta^{\mathsf{KL}}\left(\mathcal{P} \| \mathcal{Q}\right) = \int_{-\infty}^{\infty} p(x) \ln \frac{p(x)}{q(x)} \, \mathrm{d}x \tag{4.23}$$

$$\Delta^{\mathsf{KL}}\left(\mathcal{N}_b \| \mathcal{N}_a\right) = \frac{1}{2}\left(\mathrm{tr}\left(\Sigma_a^{-1}\Sigma_b\right) + (\vec{\mu}_b - \vec{\mu}_a)\Sigma_a^{-1}(\vec{\mu}_b - \vec{\mu}_a)^\top - \ln\left(\frac{\det\Sigma_b}{\det\Sigma_a}\right) - m\right) \tag{4.24}$$

$$\mathrm{tr}(\Sigma) = \sum_{i=1}^{m} \sigma_{ii} \tag{4.25}$$

Off all investigated appearance models, the mixture of Gaussians is expected to be the best method to describe the colour distributions of the individuals, because it can adapt itself to the distribution of the colours in one cluster.

The previous chapter described all colour spaces in which the appearance models can be applied. One of them is the novel colour/geometry space hsz that describes the hue, saturation and height of each pixel in the point cloud. This one is particularly interesting because it includes information about the position of the pixel in the data space instead of in the model, like the Disks and Rings model. This allows for general methods like a mixture of Gaussians to apply to both colour and geometry at once. The next two extensions use explicit knowledge of this construction to improve the general mixture of Gaussians.

4.7 Semi correlated mixture of Gaussians

The *covariance matrix* Σ describes the covariance between the dimensions. In the hsz space these dimensions are: $a = \text{saturation} \times \sin(\text{hue})$, $b = \text{saturation} \times \cos(\text{hue})$, $z = \text{height}$. The first two dimensions span a Cartesian colour space, distances in this plane reflect distances in colour. The third dimension denotes the height of the pixel in the real world.

(a) A person's point cloud in the hsz space.

(b) The result of the clustering when three Gaussians are used.
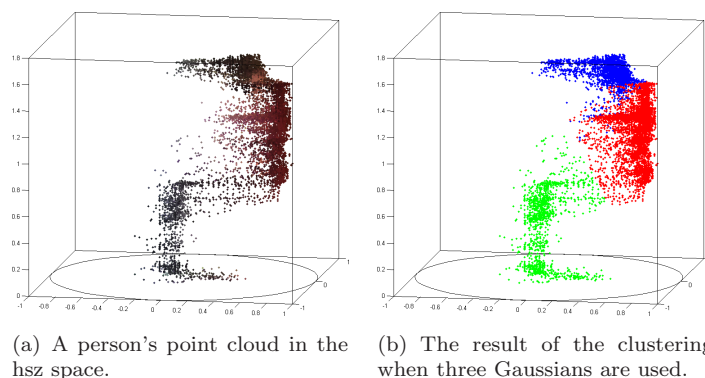
FIGURE 4.8: Two plots in hsz space. Note how the clusters of colours occur on different heights, but are not correlated to the height.

The hypothesis is that blobs of colours occur at different heights but are not linearly correlated to the height, see figure 4.8. This information can be used to improve the general mixture of Gaussians. We allow the mean vectors $\vec{\mu}_i$ to take any value, that is a cluster can appear at any position in the hsz space. But we limit the covariance matrices

Σ_i to be of the form:

$$\Sigma = \begin{bmatrix} \sigma_{aa} & \sigma_{ba} & 0 \\ \sigma_{ab} & \sigma_{bb} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix} \tag{4.26}$$

Now the colours can be correlated as before, but the height is disconnected. Graphically this means that the Gaussian density functions have become an upright ellipsoid in the hsz space. I expect that this will improve the separability of the colours.

4.8   Semi correlated mixture of Gaussians with separation planes

As described in the previous section, we assume the colours of a person to appear at different heights. Indeed, when the point clouds would be perfect, some hard edges can be found, for example at the transition from trousers to shirt. The appearance model of this section is an extension of the previous one that tries to find these transitions.

Normally each pixel has a probability for each of the cluster centres. In the appearance model proposed in this section that probability is positive for the nearest cluster and zero for all the others. The distance measure that is used to determine the nearest cluster is:

$$\Delta_\Sigma^\updownarrow (\vec{\mu}, \vec{x}) = \sqrt{\frac{(x_z - \mu_z)^2}{\sigma_{zz}^2}} \tag{4.27}$$

The result of this assignment is as if there are horizontal separation planes (parallel to the ab-plane) that divide the space dynamically into layers. This is similar to the Disks model, only with dynamic disks instead of disks that are fixed at a certain height and using a Gaussian density function instead of an uniform distribution.

I expect this model to perform well when a person's colours are well separable, but it will fail when people wear clothes with different colours at the same height.

Chapter 5

Method

While the future implementation will be a *tracking application*, the models are tested by *data association* using a *classifier*. The motivation for this simplification is that tracking involves a great number of parameters which can add noise to the comparison of the colour models.

The eventual tracker will use tracking by location within one field of view and recognition by colour when a person re-appears somewhere else (see Future work, page 44). In other words, the colour model will be used to *improve* the existing tracker rather than to *replace* it. Therefore, the results of this study will form a lower bound with respect to the performance of the eventual implementation.

5.1 Data association

To compare the appearance models objectively, they were isolated from the tracking application and tested in an off-line classifier. Two recordings were made with respectively 4 and 5 Eagle Eyes. A recorded 'frame' of one camera consisted of a point cloud of the scene and a list of locations where the Eagle Eye had detected a person in that frame.

For each detected person, a cylindrical point cloud was extracted as illustrated in figure 1.2. Each point cloud was manually assigned to one of the people featuring in the recording. The point clouds were treated as if they were independent measurements, meaning that the detection location and time were removed.

The point clouds of one of the cameras were chosen as test set, the point clouds of all other cameras formed the training set. Then, one of the appearance models is trained on the training set to form an *accumulated representation* for each individual in the recording.

Subsequently, each point cloud in the test set is associated with one of the accumulated representations. When all point clouds in the test set are classified, the results are evaluated to score the concerned appearance model.

This process is repeated k times, where k is the number of cameras, assigning another camera as test set each iteration. Repetitive testing like this is called k-fold cross validation [7]. Each appearance model is tested as described above. The next sections will elaborate on the training, classification and evaluation process.

5.1.1   Training

For each point cloud in the training set, an appearance model is created that forms a representation of the detected person. The simplest model described in this paper is the mean RGB colour of all pixels in a person's point cloud. The feature vector of a representation created by that model is $\vec{\mu} = [\overline{r}\ \overline{g}\ \overline{b}]$, a row vector where $\overline{r}$, $\overline{g}$ and $\overline{b}$ correspond to the mean red, green and blue values respectively. In other words, for the point cloud $\mathcal{P} = [\mathbf{x}\ \mathbf{y}\ \mathbf{z}\ \mathbf{r}\ \mathbf{g}\ \mathbf{b}]$ of each detection, the means of the data vectors $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{b}$ (column vectors) are calculated and stored in the feature vector $\vec{\mu}$ that serves as abstract representation of the detected person.

All representations of a single person in the training set are averaged to create an accumulated representation of that person. Consequently than accumulated representation has the same structure as a representation based on a single point cloud. How the representations are constructed and averaged depends on the type of model and colour/geometry space. See also chapter 3 and 4.

5.1.2   Classification

If the appearance model is descriptive and discriminative enough, it should be possible to assign newly detected people to one of the accumulated representations. This association is done by calculating the distance between a the representation of this detected person and each trained accumulated representation. The accumulated representation with the smallest distance is chosen and the detected person is classified accordingly. The most suitable distance measure differs per colour space and per appearance model, they are explained in chapter 3 and 4.

In the real tracking application the classification process would be similar, but *on-line*, meaning that there is no *training* and *test set*. All representations in the past are used as the '*training set*' and all new representations are the 'test set'. Each new representation is assigned to the termtrack with the best matching accumulated representation, or a

new track is initiated when there is no appropriate match. The best matching track is updated by the new representation. Thus a representation is first classified and then used as training data to alter the accumulated representation of the track.

More thoughts on improvements can be found in the Future work section (page 44).

5.1.3   Evaluation

A descriptive and discriminative appearance model would be able to assign each detection in the test set to the correct person, based on the accumulated representation created from all previous representations of that person. The results of such a data association can be displayed in a confusion matrix. The rows of the confusion matrix denote the actual classes (i.e. the person of whom the detection was recorded), the columns are the predicted classes (i.e. the person as which the representation was classified). The cells in the matrix contain the number of detections q_{ij} that are person i and are classified as person j, see table 5.1. By convention the actual classes and the predicted classes are in the same order, thus a perfect classification would result in a diagonal confusion matrix.

$$
\begin{array}{cc|ccc}
 & & \multicolumn{3}{c}{\text{Predicted class}} \\
 & & 1 & \cdots & n \\
\hline
\text{Actual} & 1 & q_{11} & \cdots & q_{1n} \\
 & \vdots & \vdots & \ddots & \vdots \\
\text{class} & n & q_{n1} & \cdots & q_{nn}
\end{array}
$$

TABLE 5.1: Confusion matrix

The *accuracy* a_i is the portion of detections $\mathcal{X}_i$, that are correctly classified as person i, for $i \in \{1, \ldots, n\}$. The *overall accuracy* is defined as the total number of correctly classified instances divided by the total number of instances. This might look like a useful measure for the performance of an appearance model. However, when the sizes of the classes are not equal, the overall accuracy is not accurate. In that case, it is better to calculate the *average accuracy per person*, $\overline{a}$.

$$\mathcal{X}_i = \{x | x \text{ is person } i\} \qquad \mathcal{Y}_i = \{x | x \text{ is classified as person } i\} \qquad (5.1)$$

$$a_i = \frac{|\mathcal{X}_i \cap \mathcal{Y}_i|}{|\mathcal{X}_i|} = \frac{q_{ii}}{\sum_{j=1}^{n} q_{ij}} \qquad\qquad \overline{a} = \frac{1}{n} \sum_{i=1}^{n} a_i \qquad (5.2)$$

As described before, k-fold cross validation is used to get a reliable accuracy measure. The accuracy of all of the k passes are averaged to obtain an *average accuracy per person per pass*:

$$\bar{\bar{a}} = \frac{1}{k} \sum_{i=1}^{k} \bar{a}_i \tag{5.3}$$

5.2   Implementation

All models and methods described in this thesis have been implemented in MATLAB® for testing purposes. Bear in mind that MATLAB® is a scripting language and thereby it is important to pay attention to the way the algorithms are implemented. The best way to use MATLAB® is to avoid for-loops and use as much linear algebra as possible.

5.2.1   Assumptions

This study focusses on the *appearance models* rather than the stereo vision or the tracking mechanism. It is assumed that the point clouds and the exact location of each detected person are provided. Furthermore it is assumed that the cameras are placed in an indoor environment with constant lighting. That is, no lights are turned on or off during the recording and the colour of all light sources is approximately equal. Especially the latter can upset the system, as we will see in the Results section (page 39).

5.2.2   Preprocessing

The Eagle Grid is built by *Eagle Vision Systems* (EVS) to track people by their location only. For a detected person $\mathcal{D}_{t,i}$ at position $\vec{x}_{t,i}$ which is very close to the end of track $\mathcal{T}_{t-1,j}$ it is likely that the $\mathcal{D}_{t,i}$ is the same person as the one that created track $\mathcal{T}_j$.

Because the actual appearance of a person is currently not used by the tracker, the point clouds do not have to be very precise. It is good enough to have clear and distinct blobs in the z-map (see figure 1.1(e)). However, when building an appearance model these point clouds should be more accurate; this section describes how the quality of the point clouds is improved without altering the point cloud generation procedure. Indeed the point clouds are given, so only post-processing is possible.

Removing background pixels

The main problem of the current point clouds is an effect called *foreground fattening*. The result is that foreground objects in point clouds appear too *fat*. As shown in figure 5.1(a), pixels that belong to the background are classified as object pixels, which introduces noise in the point clouds that pollutes the appearance models.

To correct this, the pixels that are considered background in the flat main image are removed from the point cloud. A distinction between foreground and background is made by subtracting a background image from the main image and calculating the difference. All pixels with a difference less than half the maximum measured difference are considered background and therefore they are removed from the point cloud. This is a basic background subtraction method, more advanced methods exist, but this one is fast and effective enough. See figure 5.1(b) for an example.

(a) The originally extracted point cloud.

(b) The point cloud after removing foreground fattening.

(c) The cylinder has been shifted to fit the person.
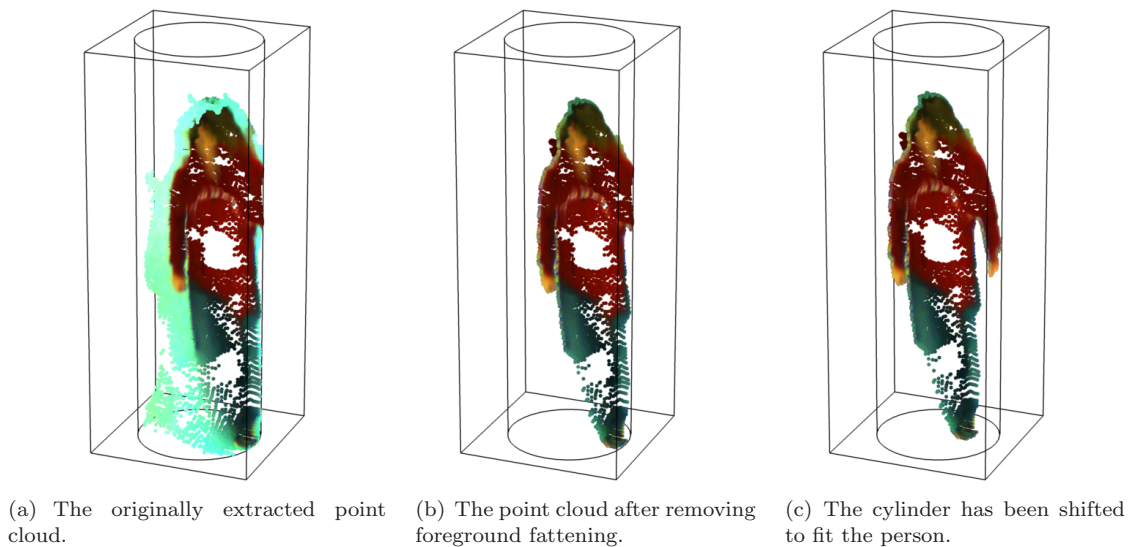
FIGURE 5.1: Removal of the foreground fattening effect and re-centring the cylinder

Refining the position detection

The position of a person is provided by the Eagle Eye together with a point cloud of the scene. My algorithm extracts a cylinder with a radius of 30 centimetres and a height of 2.20 metres. These values are heuristically chosen to contain the point cloud of a person. This cylinder contains a subset of the original point cloud and should contain the person entirely. Because the location returned by the Eagle Eyes is not always accurate, the person often appears at the edge of a the cylinder. This is corrected by iteratively shifting the cylinder to fit the person. Every iteration the cylinder is moved to the *centre of mass*

of the points inside it. Then a new subset is extracted from the original point cloud and the cylinder is shifted again.

There are no scientific results of this method, but it was tested on ± 50 randomly selected people and point clouds, and every time it gave similar results as shown in figure 5.1(c).

Chapter 6

Experiments

The previous chapters explained how 38 appearance models have been developed and how they should be evaluated. This chapter will provide the parameters and circumstances under which the experiments are conducted and finally the results will be presented.

6.1 Setup

When building any computer vision system, the danger of over-specialisation always looms ahead. One of the first thing a vision researcher does when a problem has to be solved is looking at the data. This means that whatever he or she will develop, it will be based on assumptions about the data. Some of those assumptions are explicit (steady lighting, indoor environment, using stereo cameras), but some are more implicit, or latent. To guard the integrity of the results, the models of this study are evaluated on two different datasets. One was recorded at *Eagle Vision Systems* (EVS) and was used to develop the models, the other was recorded at *Philips Shop Lab* (PSL) and has remained unseen until all models were finished. Not only were the recordings held at different locations, also different cameras were used, different lighting was applied and different people featured in the recordings.

The first recording was held at EVS's test bed, this is EVS's test room where there system is constantly running to test its performance and to try new features. The testbed dates from before the development of the Eagle Eye and is equipped with six commercial stereo cameras, called *Bumblebees*. The Bumblebees are permanently mounted on a frame such that their fields of view overlap, four of them were used for the first recording. Nowadays the six Bumblebees are accompanied by four Eagle Eyes.

In the first recording three people were walking around, one of them changed clothes twice (out of sight); they were treated as 5 'individuals'. Their clothing was relatively simple, shirts and pants with one clear dominant colour. The people entered and exited the area several times and they were captured at least once by each camera. The area was illuminated by fluorescent tubes, which caused a little flickering in the recorded frames.

The second recording was held at Philips Shop Lab, a laboratory built by Philips to test all kinds of technology applicable to shops. This time 5 Eagle Eyes were temporarily mounted such that their fields of view did *not* overlap. Now two people were walking around, and again one of them changed clothes twice; they were treated as 4 'individuals'. The fabric of their clothing contained more structure, like stripes and patches across their shirts and jackets. The people were walking around the lab acting as if they were shopping. They were captured at least once by four of the cameras, one camera captured only two of them. The Shop Lab area was illuminated by all kinds of lighting; daylight, fluorescent light, LED light and coloured spot lights.

Refer to table 6.1 for an overview of the recording parameters.

	Dataset A EVS.5	Dataset B PSL.2
Location	Eagle Test Bed	Shop Lab
Company	Eagle Vision	Philips
Date/time	28-10-2011 13:22:17	20-03-2012 14:02:52
Camera type	Bumblebee2	EagleEye
Lighting	fluorescent	various
Overlap	yes	no
Cameras	4	5
Individuals	5	4
Height	2.82m	2.51m
Camera fov	2.1m $\times$ 2.9m	1.8m $\times$ 2.6m
Covered area	3.6m $\times$ 4.8m	10.1m $\times$ 8.0m
Frames	4 $\times$ 799	5 $\times$ 2112
Duration	1m20s	7m05s

TABLE 6.1: Parameters of the two recording sessions.

The recordings have been manually annotated; that is, each detection made by the existing software of EVS is manually assigned to the correct person (or marked as noise if it was a misclassification).

6.2 Parameters

For each model the parameters are heuristically chosen, so that similar models have similar numbers of parameters, and a meaningful comparison can be performed. The number of histogram bins is around 100, the single mean model has no parameters, the disk model uses 3 disks in accordance to Zajdel et al. [1], the ring models uses 4 cylinders $\times$ 9 disks = 36 rings and the kernel models use 4 kernels (based on the expectation that a person has around 4 distinct colours; legs, torso, face, hair).

Section 4.1 describes the histogram models. The number of bins can not be arbitrarily chosen, for instance the RGB histogram will always have n^3 bins for $n \in \mathbb{N}_{\geq 1}$. The exact number of bins differs per colour space, table 6.2 provides an overview which I also will explain in text. Firstly the RGB histogram is three-dimensional and contains $5 \times 5 \times 5$ bins (figure 4.1(a)), the rgb histogram is divided into 14×14 squares and almost half of the squares is unused (figure 4.1(b)). The HSV space is divided in 4 layers perpendicular to the value axis, each layer is split up using the radial histogram of figure 4.3(b) with 1 to 4 rings (figure 4.4(b)). Since the hue is one-dimensional, it can be divided in exactly 100 bins. The hs space is divided like figure 4.4(b), only with 6 rings and the hsz space is divided into 4 layers and each layer was divided as the radial histogram of figure figure 4.4(b) with 3 rings.

colour space	histogram shape	composition of bins/kernels	bins or kernels
RGB	cube	5^3	125
rgb	triangle	$14 \times (14+1)/2$	105
HSV	cone	$3 \times 1^2 + 3 \times 2^2 + 3 \times 3^2 + 3 \times 4^2$	90
hue	circle	1×100	100
hs	disk	3×6^2	108
hsz	cylinder	$4 \times (3 \times 3^2)$	108

TABLE 6.2: Composition of the histograms

6.3 Results

All results of the automated evaluation process are shown in appendix D. A selection of these results are the *average accuracies per person* shown in table 6.3 and 6.4.

	bins or kernels	RGB	HSV	hue	rgb	hs	hsz
Histogram	±100	0.452	0.708	0.918	0.897	0.933	0.929
Single mean	1	0.675	0.804	0.515	0.830	0.793	0.703
Disks	3	0.713	0.888	0.688	0.890	0.891	0.890
Rings	36	0.751	0.932	0.845	0.897	0.923	0.928
k-Means	4	0.643	0.901	0.662	0.840	0.899	0.891
Mixture of Gaussians (MoG)	4	0.848	0.860	0.666	0.878	0.856	0.774
MoG - Semi correlated	4						0.750
MoG - Separation planes	4						0.512

0.5 ▮▮▮▮▮▮▮▮▮ 0.9

TABLE 6.3: Average accuracy of each model / colour space combination in the Eagle Vision recording, averaged over 4-fold cross validation. The average of all values displayed in this table is 0.764.

	bins or kernels	RGB	HSV	hue	rgb	hs	hsz
Histogram	±100	0.382	0.367	0.352	0.370	0.327	0.423
Single mean	1	0.340	0.326	0.356	0.336	0.382	0.385
Disks	3	0.399	0.400	0.301	0.296	0.316	0.307
Rings	36	0.350	0.396	0.297	0.340	0.378	0.365
k-Means	4	0.353	0.334	0.346	0.395	0.388	0.431
Mixture of Gaussians (MoG)	4	0.283	0.251	0.322	0.325	0.222	0.232
MoG - Semi correlated	4						0.291
MoG - Separation planes	4						0.111
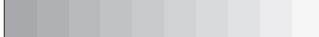
0.2 ▮▮▮▮▮▮▮▮▮ 0.4

TABLE 6.4: Average accuracy of each model / colour space combination in the Shop Lab recording, averaged over 5-fold cross validation. The average of all values displayed in this table is 0.321.

Overall the models perform much better in the EVS dataset than in the PSL dataset. The people in the EVS dataset wore very distinctly coloured shirts and the illumination is quite constant, compared to the PSL dataset. This makes the latter dataset more complex, but also more realistic.

The chromaticity models rgb, hs and hsz perform best. There is no significant difference in performance between them. The reason that they perform so well is that they describe the actual colour of an object without any luminance influences. Highlights and shadows, which occur much more often than we are aware of (figure 3.1), have less effect on these chromaticity spaces.

Due to variance the luminance, the RGB colour space is expected to perform badly on these datasets. The results in table 6.3 and 6.4 show that this is indeed the case. HSV, which is also a full colour space, performs just as good as the hs space. An explanation is that the effect of the luminance is only reflected in the value dimension, which is better than when it is reflected in all three red, green and blue dimensions.

Furthermore, as explained in section 3.4 the hue space is expected to perform badly because the images are poorly saturated which causes noise in the hue space. Secondly, there is no correct way to calculate the mean hue, which causes problems in the mean colour based appearance models. And indeed, both table 6.3 and 6.4 show a column with low values in the hue column. Except for the histogram models in the EVS recording because the histograms do not use a mean and because the people of the EVS recording wore very distinctly coloured shirts.

The expectation was that the hsz space would perform best because some of the geometrical information is included in this space. however, the results provide no proof that it performs better than other spaces. While examining the point clouds, they seemed quite noisy, especially in the z direction (i.e. the height). This noise might be the reason that the geometrical information does not add any value. The reason that this noise was never reduced by EVS is simply that it was not relevant before. During the development of the Eagle Grid they focussed on building a robust detection method rather than a detailed description of a person's appearance.

As one should expect, the Single mean model performs the worst of all. The appearance of a person is simply too complex to fit into a single colour. The histogram models are, as expected, among the best performing models (at least in the chromaticity spaces). The reason for this is that the are able to capture complex distributions. As discussed in section 4.1 the histograms use a relatively long feature vector of approximately 100 elements.

The disks model performs surprisingly well with a feature vector of only 3, 6 or 9 elements (depending on the colour space) and is very easy (i.e. fast) to calculate. It performs much better in the EVS recording than in the PSL recording, probably because the PSL recording contains to much noise. The rings model performs even better. A person's shape is implicitly embedded in these models, which might be interesting to investigate further.

The hypothesis was that a mixture of Gaussians would perform best among the non-histogram models. During the development it seemed to work well, but in this comparison one can easily see that it is not as good as some others. The suggested improvements perform slightly worse than the original mixture of Gaussians. This might be caused by the noise in the point cloud. For the mixture of Gaussians model with separation planes, the assumption that different colours would occur at different heights was made, which is not always true.

Chapter 7

Conclusion

Looking at the results of the EVS dataset, one may say that it is indeed possible to build an appearance model that can be used to recognise individuals. Some of the accuracy values are above 0.9, meaning that the predictions were more than 90% correct. The PSL dataset however, shows a maximum accuracy of around 0.4, which is unacceptable for a real-world application. Nevertheless, as discussed before, the point clouds are noisy and this is something that could be improved.

Please note that a random data association would result in an average accuracy of 0.200 for the EVS recording and 0.250 for the PSL recoding. That is because the EVS and PSL recordings featured 5 and 4 people respectively.

The results of the first data set show that RGB and hue are not suitable for the re-identification of people. Despite that there is not enough significant evidence to point out the best colour model from these experiments, the results of both data sets indicate that the histogram models, the ring models and the k-means models perform best, especially in the hue-saturation space and the hue-saturation-height space.

The three variants of the mixture of Gaussians should have been an improvement on k-Means, but they are not. The normal mixture of Gaussians was introduced to capture the variance in all directions, but it suffers greatly from the noise in the point clouds. The two other mixture of Gaussians models use the z information to improve the separation between the colours and again precisely because of the noise in this dimension they perform worse than any other.

The best conclusion to draw is that the hue-saturation-height space should be further investigated and improved. The point-cloud creation should be revised or at least its parameters should be adjusted such that an accurate point cloud is created. For now the hue-saturation-height space should be used in a histogram or a k-means model.

One should choose one of tested appearance models depending on the requirements. Histograms can be constructed very fast and in constant time, whereas the k-means requires many more calculations. Furthermore, the execution time of k-means depends on the distribution of the data and the initialisation of the cluster centres. On the other hand, generally histograms produce feature vectors that are much longer (e.g. 108 elements) than the feature vectors constructed by k-means (e.g. 8 or 12 elements). The current results show that k-means outperforms the mixture of Gaussians in almost every colour space, especially when the amount of noise is high. But still, if the level of noise is reduced, the mixture of Gaussians should be able to describe the distribution of the data better than k-means (see figure 4.7).

7.1 Future work

This section will describe a number of thoughts for future research. Some of them are just ideas, others are important implementation notes. The latter are tuned for implementation within the Eagle Grid.

7.1.1 Classify tracklets instead of detections

This research focussed on finding the best appearance model to describe a person. Nevertheless, in a real world implementation one should use tracking by position within a single field of view. Such a sub track is called a *tracklet* and all tracklets of a person form its *track*. The tracking by position algorithm of *Eagle Vision Systems* (EVS) is very robust and the colour model should be an extension to that algorithm, not a replacement. The tracklet could be assigned to one of the known tracks after the person leaves the field of view. By then there is enough information to update the track reliably.

A tracklet does not only contain more information, but also better information. At the edge of the field of view, the detections are very noisy, so the moment a person enters the view is not the right moment to classify it. On the other hand, when a person is exactly underneath a camera, his or her clothes can barely be seen. Classifying a complete track at once would reduce the effect of these artifacts.

Ultimately, a tracklet should save the point clouds of each detection and create a single representation of all those combined point clouds at once. Unfortunately this is not possible in the current implementation of the Eagle Grid. The Eagle Eyes have no knowledge about previous detections, tracklets or tracks, they only send a list of representations per frame to the central tracker. Sending the entire point cloud would result in too much data traffic and implementing a local tracker in the Eagle Eye would break the current design.

If we want to keep the design intact, a solution would be buffer all representations of a tracklet in the global tracker and average them when a tracklet is completed. Then classify the entire tracklet at once and update the assigned track. Another option would be to break the design and allow for local tracking within the Eagle Eyes.

Moreover, the representations could be extended with a reliability measure. This can be done dynamically by the Eagle Eyes or statically by the Eagle Tracking Engine. The latter would involve a recency/latency function that assigns a lower reliability to the first and last couple of representations. They are expected to contain much noise because they are probably taken from the edge of the field of view. It could also use a heuristic value based on the location in the field of view. However, in the current implementation, the tracker has no knowledge of the position or angle of view of the Eagle Eye.

7.1.2 Classify partial tracklets

Classifying a tracklet after it is completed, as suggested in the previous section, is suitable for statistical analysis, but not if you want to know who is walking around at this very moment. A variation would be to classify the partial tracklet in real time, acting every iteration as if the tracklet was completed. This online classification is only temporary, to display to the user, the track should only be updated when the tracklet is completed.

7.1.3 Improve the point clouds

As mentioned several times before, the point clouds are not very accurate and an improvement in the point clouds could have a major effect of the performance of the colour models. Because the point clouds were provided externally, this study has not focused on building point clouds and I do not have any suggestions how to improve them. But is it clear that the point clouds are noisy and that people are inaccurately distinguished from the background. The first problem result in an inaccurate height estimation of the pixels (used by some models / colour spaces). The last problem results in background pixels being added to the foreground, disturbing the colour measurements.

7.1.4 Use an illumination-invariant chromaticity space

In this research some of the most common colour spaces are used, however, there do exist more advanced colour/chromaticity spaces. For example [2] describes a chromaticity space which is invariant under the colour and intensity of the illuminating light source. This is especially interesting when the sensors are distributed over a large area where the light sources differ per sensor.

There are more papers that conducted experiments on colour spaces, Gevers and Smeulders [11] for instance compared 8 colour spaces on 7 different aspects and shows a clear overview of which model to use under which circumstances. And Gevers [4] even describes 14 colour spaces on 9 different conditions. These could be used as an improvement on the models discussed in this thesis.

7.2   Final conclusion

To conclude, this study introduced a novel way of dividing a circle-area for the purpose of uniformly binning a radial space. These histograms formed the baseline for testing other methods like calculating one or more means, fitting a mixture of Gaussians and others. k-Means performed just as well as the histograms (in the chromaticity spaces), producing a much smaller representation, but using more computational resources. Some other models perform a little less, but improving the quality of the point clouds will contribute to their performance.

Appendix A

Arctangent for vectors of 2 elements

The arctan2($\vec{x}$) is a slightly adapted version of the arctan2(x_1, x_2) function that works on a row vector $\vec{x} = [x_1 \ x_2]$ instead of two scalars x_1 and x_2 and whose range is $[0, 1)$ instead of $[-\pi, \pi)$.

$$\text{arctan2}(\vec{x}) = \begin{cases} \frac{1}{2\pi} \arctan\left(\frac{x_1}{x_2}\right) & \text{if } x_2 > 0 \\ \frac{1}{2\pi} \arctan\left(\frac{x_1}{x_2}\right) + \frac{1}{2} & \text{if } x_2 < 0 \\ \frac{1}{4} & \text{if } x_2 = 0, x_1 > 0 \\ \frac{3}{4} & \text{if } x_2 = 0, x_1 < 0 \\ 0 & \text{if } x_2 = 0, x_1 = 0 \end{cases} \tag{A.1}$$

.

In MATLAB®:

```
1  function a = arctan(x)
2      a = mod(atan2(x(1), x(2)) / (2*pi), 1);
3  end
```

Appendix B

Radial Bins

B.1   Proof of equally sized bins

The concept of equally sized radial bins is firstly proven for a circle with radius $r = n \in \mathbb{N}_{\geq 1}$ which is divided into n rings and $k = n^2$ bins. Subsequently this will be scaled to match the actual case for the hue-saturation space where the radius is fixed at $r = 1$ and there are three times as many bins to make them more square like, $k = 3n^2$.

B.2   The basic circle area distribution
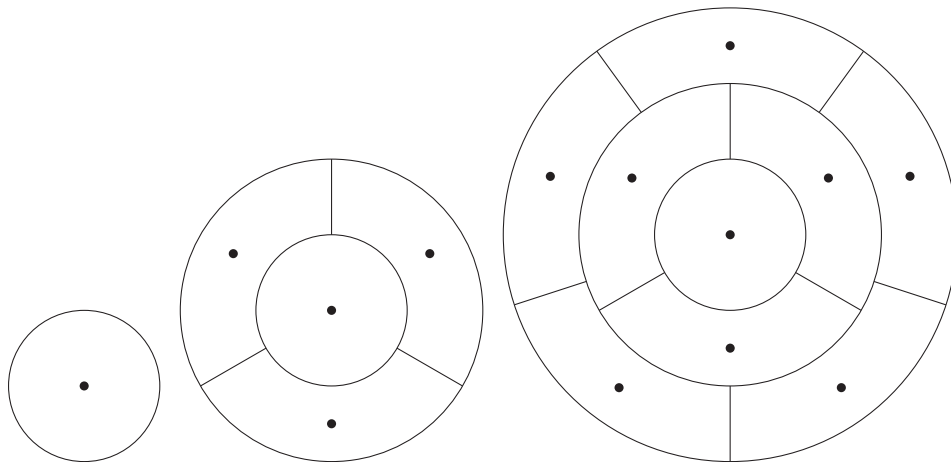
FIGURE B.1: Three circles with radius $r_n = n \in \{1, 2, 3\}$, all divided in n^2 equally sized bins.

To prove: When a circle with radius $r_n = n \in \mathbb{N}_{\geq 1}$ is divided in n rings of equal width and each ring i (counted from the centre) is divided into $k_i = 2i - 1$ equally sized bins, the area of each bin in each ring is equal, π to be precise. See figure B.1.

Proof. The circle-area of the ith circle is $a_i = \pi r_i^2 = \pi i^2$ and the ring-area b_i is:

$$b_i = a_i - a_{i-1} \tag{B.1}$$

$$= \pi i^2 - \pi (i-1)^2 \tag{B.2}$$

$$= (2i - 1)\pi \tag{B.3}$$

Thus, if each ring i with area $b_i = (2i - 1)\pi$ is divided into $k_i = 2i - 1$ bins, the area of each bin in each ring is $\frac{b_i}{k_i} = \pi$.

$\square$

B.3 The adapted bin distribution
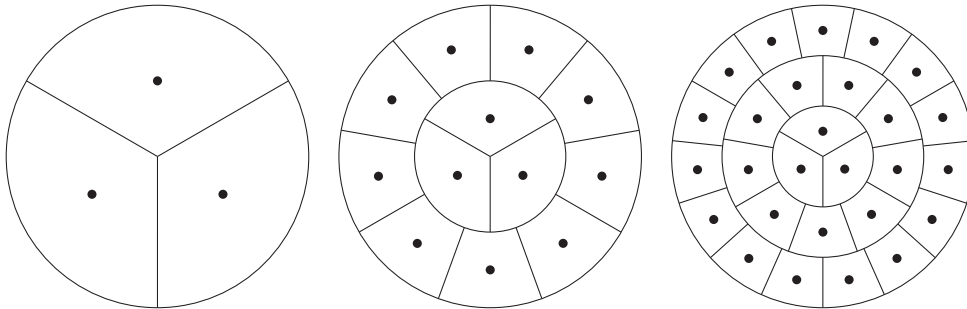
FIGURE B.2: Three circles width radius $r_n = 1$, $n \in \{1, 2, 3\}$, each divided in $3n^2$ equally sized bins.

To prove: When a circle with radius $r = 1$ is divided in n rings, $n \in \mathbb{N}_{\geq 1}$, of equal width and each ring i (counted from the centre) is divided into $k_i = 3(2i - 1)$ equally sized bins, the area of each bin in each ring is equal, $\frac{1}{3n^2}\pi$. See figure B.2.

Proof. The previous proof showed that a circle with radius $r = n, n \in \mathbb{N}_{\geq 1}$ can be divided in $k = n^2$ equally sized bins. If r is fixed at $r = 1$ then $r_i = \frac{i}{n}r = \frac{i}{n}$, i.e. the entire circle is scaled. Furthermore if each bin would be divided in 3 equal parts a circle can also be divided in $k = 3n^2$ equally sized bins (see figure B.1 and B.2). Similar to equation B.1 to

B.3:

$$b_i = a_i - a_{i-1} \tag{B.4}$$

$$= \pi(r_i)^2 - \pi(r_{i-1})^2 \tag{B.5}$$

$$= \pi\left(\frac{i}{n}\right)^2 - \pi\left(\frac{i-1}{n}\right)^2 \tag{B.6}$$

$$= \pi\frac{2i-1}{n^2} \tag{B.7}$$

The number of bins has been multiplied by 3, so $k_i = 3(2i-1)$ and the size of a single bin in ring i is $\frac{b_i}{k_i} = \frac{\pi(2i-1)/n^2}{3(2i-1)} = \frac{\pi}{3n^2}$, which is independent of i, so again, all bin-areas are equal. $\square$

Appendix C

k-Means++

This appendix shows the pseudo code of the algorithm mentioned in section 4.5 (k-Means). It is an alternative for random initialisation of k-means. The cluster centres are chosen by random sampling, but the chance a data point is chosen depends on its distance to the existing centres. See "k-means++: the advantages of careful seeding" [8] for more details.

Algorithm 2 k-Means++ initialisation

Input: A pixel set $\mathbf{X}$ with n pixel vectors x_j of length m.

   $\vec{\mu}_1 \leftarrow \text{RANDOMSAMPLE}(\mathbf{X})$

  for $i = 2 \rightarrow k$ **do**

      for $j = 1 \rightarrow n$ **do**

         $\delta_j = \min_{\vec{\mu}_a \in M} \Delta^{\mathsf{E}}(\vec{x}_j, \vec{\mu}_a)$

      end for

      for $j = 1 \rightarrow n$ **do**

         $p_j = \frac{\delta_j^2}{\text{SUM}(\delta^2)}$

      end for

      $\vec{\mu}_i \leftarrow \text{WEIGHTEDRANDOMSAMPLE}(\mathbf{X}, \mathbf{p})$

  end for

Appendix D

All Results

The next two pages show the mean average accuracy per person of all tested appearance models in both data sets. The nine subsequent pages show the number of correct classifications per person per fold.

		Rgb	Nrgb	Hsv	Hue	Hs	Hsz
GroundTruth		1.000	1.000	1.000	1.000	1.000	1.000
Random		0.202	0.202	0.202	0.202	0.202	0.202
Histogram	10	0.294	0.801	0.629	0.746	0.824	0.756
Histogram	30	0.400	0.732	0.677	0.828	0.875	0.860
Histogram	100	0.452	0.897	0.707	0.918	0.933	0.929
Histogram	300	0.600	0.906	0.707	0.878	0.929	0.938
Mean		0.675	0.830	0.804	0.515	0.792	0.703
Disks	2	0.700	0.885	0.879	0.638	0.877	0.889
Disks	3	0.713	0.890	0.888	0.688	0.891	0.890
Disks	4	0.706	0.895	0.883	0.681	0.861	0.865
Disks	6	0.713	0.907	0.900	0.661	0.892	0.889
Rings	10	0.704	0.863	0.886	0.700	0.868	0.884
Rings	30	0.751	0.897	0.932	0.845	0.923	0.928
Rings	100	0.754	0.913	0.924	0.845	0.925	0.926
Rings	300	0.761	0.919	0.931	0.874	0.931	0.932
Kmeans	2	0.661	0.870	0.885	0.520	0.877	0.771
Kmeans	3	0.636	0.807	0.889	0.577	0.882	0.876
Kmeans	4	0.643	0.840	0.901	0.662	0.899	0.891
Kmeans	6	0.693	0.895	0.910	0.582	0.915	0.894
Mog	2	0.809	0.892	0.862	0.557	0.827	0.813
Mog	3	0.826	0.875	0.836	0.553	0.878	0.767
Mog	4	0.848	0.878	0.860	0.666	0.856	0.774
Mog	6	0.788	0.862	0.818	0.605	0.899	0.754
MogSC	2						0.786
MogSC	3						0.762
MogSC	4						0.749
MogSC	6						0.765
MogSP	2						0.687
MogSP	3						0.631
MogSP	4						0.512
MogSP	6						0.393

0.5 ▮▮▮▮▮▮▮▮ 0.9

TABLE D.1: Average accuracy of the models in the EVS recording.

		Rgb	Nrgb	Hsv	Hue	Hs	Hsz
GroundTruth		1.000	1.000	1.000	1.000	1.000	1.000
Random		0.212	0.212	0.212	0.212	0.212	0.212
Histogram	10	0.375	0.310	0.356	0.338	0.307	0.347
Histogram	30	0.399	0.386	0.322	0.355	0.359	0.293
Histogram	100	0.382	0.370	0.367	0.352	0.327	0.423
Histogram	300	0.373	0.363	0.368	0.342	0.331	0.369
Mean		0.340	0.336	0.326	0.356	0.382	0.385
Disks	2	0.412	0.270	0.421	0.315	0.312	0.281
Disks	3	0.399	0.296	0.400	0.301	0.316	0.308
Disks	4	0.372	0.310	0.372	0.287	0.334	0.330
Disks	6	0.352	0.316	0.365	0.293	0.359	0.369
Rings	10	0.337	0.332	0.369	0.308	0.389	0.394
Rings	30	0.350	0.340	0.396	0.297	0.378	0.365
Rings	100	0.352	0.352	0.379	0.314	0.368	0.369
Rings	300	0.336	0.324	0.364	0.305	0.354	0.355
Kmeans	2	0.326	0.330	0.287	0.199	0.390	0.376
Kmeans	3	0.331	0.322	0.298	0.302	0.369	0.388
Kmeans	4	0.353	0.395	0.334	0.346	0.388	0.431
Kmeans	6	0.358	0.393	0.364	0.304	0.409	0.406
Mog	2	0.331	0.341	0.350	0.263	0.413	0.335
Mog	3	0.291	0.336	0.251	0.323	0.390	0.263
Mog	4	0.283	0.325	0.252	0.322	0.222	0.232
Mog	6	0.252	0.324	0.178	0.302	0.327	0.304
MogSC	2						0.379
MogSC	3						0.302
MogSC	4						0.291
MogSC	6						0.297
MogSP	2						0.205
MogSP	3						0.172
MogSP	4						0.111
MogSP	6						0.174

0.2 ▬▬▬▬▬▬▬ 0.4

TABLE D.2: Average accuracy of the models in the PLS recording.

	Rgb					Nrgb					Hsv					Hue					Hs					Hsz				
GroundTruth	6	88	92	28	30	6	88	92	28	30	6	88	92	28	30	6	88	92	28	30	6	88	92	28	30	6	88	92	28	30
Random	1	18	18	5	8	1	18	18	5	8	1	18	18	5	8	1	18	18	5	8	1	18	18	5	8	1	18	18	5	8
Histogram 10	6	0	3	12	4	6	88	54	21	28	6	56	41	12	9	0	68	92	22	30	6	88	89	21	22	6	67	49	18	18
Histogram 30	6	12	7	20	15	6	65	77	22	30	6	56	41	20	9	0	82	92	22	30	3	87	92	22	30	6	87	86	18	28
Histogram 100	0	32	53	22	15	6	86	89	12	30	6	54	44	22	8	6	85	92	22	30	6	87	92	22	30	6	87	92	17	30
Histogram 300	0	44	57	22	15	6	85	92	13	30	6	62	48	22	12	6	84	92	2	30	6	86	92	21	30	6	87	92	20	30
Mean	4	57	66	17	10	6	87	63	21	22	6	81	62	20	20	0	77	40	2	8	6	88	52	20	19	6	48	48	18	7
Disks 2	3	43	71	19	15	6	86	92	20	29	6	84	92	20	30	0	79	62	25	22	5	87	92	21	29	6	87	92	18	29
Disks 3	4	47	79	20	13	6	84	92	22	29	6	80	92	22	29	0	78	65	25	21	6	87	91	23	29	6	86	91	23	28
Disks 4	5	48	76	20	13	6	84	92	22	28	6	79	92	21	25	0	75	55	25	13	3	86	90	23	21	4	86	90	23	21
Disks 6	5	52	79	20	15	6	85	92	23	28	6	79	92	23	28	0	72	57	23	18	3	84	91	23	27	3	83	91	23	27
Rings 10	4	45	80	23	14	6	87	73	22	30	6	84	77	22	30	4	77	56	23	18	4	87	75	23	30	6	87	75	22	30
Rings 30	4	45	79	23	14	6	87	84	23	30	6	87	92	23	30	6	79	73	23	29	6	87	89	24	30	6	87	89	24	30
Rings 100	4	50	76	23	16	6	88	92	25	30	6	87	92	23	30	6	78	82	23	30	5	87	92	25	30	5	87	92	25	30
Rings 300	4	49	75	23	16	6	88	92	26	30	6	87	92	23	30	6	79	83	23	30	5	87	92	26	30	5	87	92	26	30
Kmeans 2	6	25	73	19	20	6	88	80	22	27	6	75	81	22	30	0	81	30	6	24	6	78	79	22	30	6	73	54	18	18
Kmeans 3	3	24	73	19	19	6	87	46	21	26	6	83	82	22	28	0	73	21	27	12	6	86	64	21	30	6	76	78	18	28
Kmeans 4	4	30	72	19	17	6	87	48	21	30	6	80	90	22	29	0	78	49	22	29	6	87	66	22	30	6	69	85	19	30
Kmeans 6	5	35	79	17	20	6	87	76	21	30	6	81	91	21	30	0	85	20	28	23	6	86	82	20	30	6	76	85	18	26
Mog 2	6	88	58	14	25	6	88	71	21	30	6	82	89	21	25	2	71	28	22	24	6	87	92	21	30	6	26	88	22	30
Mog 3	6	87	62	17	25	6	88	72	22	30	6	83	91	22	23	0	86	41	22	19	6	86	92	22	30	6	22	87	22	28
Mog 4	6	86	65	19	26	6	88	74	22	30	6	79	87	22	30	0	87	51	26	29	6	86	91	6	30	6	23	83	22	29
Mog 6	6	81	67	20	28	6	85	82	22	30	6	79	87	22	27	0	82	56	20	21	6	87	91	22	30	6	81	85	22	24
MogSC 2																										6	39	76	21	30
MogSC 3																										6	28	89	22	28
MogSC 4																										6	25	71	22	29
MogSC 6																										6	82	84	22	22
MogSP 2																										6	39	69	19	22
MogSP 3																										6	55	68	17	22
MogSP 4																										5	27	58	14	13
MogSP 6																										4	40	51	1	5

TABLE D.3: Correctly classified instances per prototype in the EVS recording, using camera 1 as test set.

GroundTruth	Rgb					Nrgb					Hsv					Hue					Hs					Hsz				
	49	142	106	115	16	49	142	106	115	16	49	142	106	115	16	49	142	106	115	16	49	142	106	115	16	49	142	106	115	16
Random	10	32	19	25	3	10	32	19	25	3	10	32	19	25	3	10	32	19	25	3	10	32	19	25	3	10	32	19	25	3
Histogram 10	40	20	15	47	1	47	142	56	114	16	49	137	72	46	12	35	142	105	101	16	48	142	106	114	10	49	106	79	111	11
Histogram 30	34	37	7	66	4	49	10	75	113	15	49	140	72	65	12	42	142	106	115	16	46	142	106	115	16	49	142	103	113	11
Histogram 100	23	60	71	106	10	49	142	106	114	16	49	140	71	98	12	49	142	106	115	16	49	142	106	115	16	49	142	106	114	16
Histogram 300	26	93	81	97	11	49	142	106	113	16	49	126	72	96	13	49	142	106	115	16	49	142	106	115	16	49	142	106	114	16
Mean	48	84	94	67	12	48	142	92	114	12	49	137	99	83	11	29	141	62	11	8	49	142	79	114	12	49	139	67	113	13
Disks 2	42	89	99	74	11	47	141	103	114	14	49	135	103	75	14	13	141	64	35	13	48	141	106	111	14	49	141	106	109	14
Disks 3	45	94	98	67	11	47	142	103	113	14	49	140	103	68	14	21	142	73	115	12	48	142	105	113	13	49	142	105	112	13
Disks 4	44	87	97	66	10	44	142	106	113	13	49	141	106	68	13	29	142	76	115	11	49	142	106	113	13	49	142	106	112	13
Disks 6	46	88	98	67	11	43	142	106	114	16	49	139	106	76	14	37	131	66	115	12	47	142	105	114	14	47	142	104	114	14
Rings 10	40	73	101	67	12	38	142	102	115	15	49	142	106	75	15	43	136	57	115	11	48	142	106	115	15	49	142	105	115	15
Rings 30	43	87	97	66	13	38	142	106	115	16	49	142	106	86	16	46	141	88	115	14	49	142	105	115	16	49	142	105	115	16
Rings 100	43	97	94	65	13	38	142	106	115	16	49	142	106	79	16	45	136	80	115	16	48	142	106	115	16	48	142	106	115	16
Rings 300	43	95	101	65	12	38	142	104	115	16	49	142	106	80	16	47	133	82	115	15	48	142	105	115	16	48	142	105	115	16
Kmeans 2	49	29	102	66	9	49	142	104	114	13	46	137	103	86	14	15	127	56	79	16	49	142	103	115	14	48	139	58	114	14
Kmeans 3	45	45	100	65	9	49	142	72	115	13	49	140	106	81	15	1	141	31	115	11	49	142	100	114	16	49	142	90	114	14
Kmeans 4	36	59	104	64	7	49	142	59	115	16	49	139	100	82	16	31	141	48	103	16	49	142	91	115	16	49	142	97	114	15
Kmeans 6	43	73	97	65	9	48	142	85	114	16	49	138	106	77	16	0	136	75	115	11	49	142	100	115	16	49	142	89	114	16
Mog 2	48	138	76	76	14	48	138	99	106	15	49	142	100	70	16	19	120	60	89	15	48	142	106	97	14	48	139	104	105	10
Mog 3	48	138	79	84	14	48	138	98	108	15	49	141	98	51	15	4	142	49	115	7	49	142	106	89	14	49	137	106	21	11
Mog 4	49	141	72	89	14	47	140	100	111	16	49	139	96	60	16	6	141	55	115	16	49	142	106	100	14	49	142	99	21	15
Mog 6	49	138	70	90	14	46	141	98	111	14	49	138	95	27	14	5	138	44	107	13	49	141	106	89	15	49	142	91	9	13
MogSC 2																										49	135	97	97	10
MogSC 3																										49	133	103	21	13
MogSC 4																										49	139	88	21	15
MogSC 6																										47	139	94	9	13
MogSP 2																										48	120	79	26	16
MogSP 3																										46	99	55	10	10
MogSP 4																										44	34	36	1	10
MogSP 6																										26	97	20	5	12

TABLE D.4: Correctly classified instances per prototype in the EVS recording, using camera 2 as test set.

GroundTruth		Rgb 8	Rgb 109	Rgb 107	Rgb 17	Rgb 32	Nrgb 8	Nrgb 109	Nrgb 107	Nrgb 17	Nrgb 32	Hsv 8	Hsv 109	Hsv 107	Hsv 17	Hsv 32	Hue 8	Hue 109	Hue 107	Hue 17	Hue 32	Hs 8	Hs 109	Hs 107	Hs 17	Hs 32	Hsz 8	Hsz 109	Hsz 107	Hsz 17	Hsz 32
GroundTruth		8	109	107	17	32	8	109	107	17	32	8	109	107	17	32	8	109	107	17	32	8	109	107	17	32	8	109	107	17	32
Random		3	30	14	2	7	3	30	14	2	7	3	30	14	2	7	3	30	14	2	7	3	30	14	2	7	3	30	14	2	7
Histogram	10	8	0	6	0	1	8	109	46	3	10	8	101	54	0	13	0	108	49	12	32	4	109	107	4	0	8	105	60	4	22
Histogram	30	8	18	34	0	4	8	96	64	0	20	8	101	55	0	13	0	109	87	9	32	4	109	107	4	31	8	109	107	4	0
Histogram	100	0	86	13	3	16	7	109	107	0	32	8	99	55	4	14	8	109	94	4	32	8	109	107	4	32	8	109	107	2	32
Histogram	300	3	103	62	3	19	8	109	107	0	31	8	107	48	4	13	8	109	78	4	32	8	109	107	3	32	8	109	107	4	31
Mean		8	102	53	2	17	8	109	105	2	12	8	109	75	2	18	8	102	57	2	13	8	109	75	2	12	8	75	42	2	10
Disks	2	8	108	58	0	30	8	109	107	0	21	8	109	107	0	24	4	109	31	10	23	8	109	107	0	21	8	109	107	0	21
Disks	3	8	107	59	0	29	8	109	107	0	20	8	109	107	0	25	0	109	36	10	25	8	109	107	1	21	8	109	107	0	21
Disks	4	8	105	59	0	26	8	109	107	1	25	8	109	107	1	28	0	109	33	11	23	8	109	107	2	23	8	109	107	1	22
Disks	6	8	104	58	0	25	8	109	107	3	21	8	109	107	3	25	0	109	18	12	21	8	109	107	8	22	8	109	107	8	21
Rings	10	8	106	65	3	15	8	109	103	3	7	8	109	107	3	23	4	109	40	4	8	8	109	107	4	13	8	109	107	4	13
Rings	30	8	106	82	3	29	8	109	100	4	15	8	109	107	4	31	8	105	66	4	28	8	109	107	4	23	8	109	107	4	26
Rings	100	8	105	83	3	29	8	109	98	4	22	8	109	107	4	30	7	107	60	4	29	8	109	107	4	28	8	109	107	4	28
Rings	300	8	104	86	3	31	8	109	95	4	25	8	109	107	4	32	8	99	65	4	29	8	109	107	4	30	8	109	107	4	30
Kmeans	2	8	70	37	2	32	6	109	96	0	26	8	109	101	3	32	2	108	14	3	22	8	109	99	2	32	8	104	67	2	12
Kmeans	3	8	85	46	2	27	6	109	68	0	26	8	109	91	2	29	0	108	20	14	2	8	109	76	2	30	8	103	101	3	25
Kmeans	4	8	93	54	2	25	8	109	60	2	31	8	108	100	2	32	3	107	30	4	14	8	109	80	2	31	8	99	102	2	26
Kmeans	6	8	89	65	2	27	8	109	91	2	32	8	109	103	3	32	0	93	52	16	26	8	109	103	2	32	6	99	102	3	27
Mog	2	8	109	69	3	26	8	109	84	4	32	8	109	105	3	32	6	107	10	2	21	4	109	107	4	15	7	108	106	4	22
Mog	3	8	109	74	3	28	8	109	77	4	32	8	109	99	2	28	0	109	26	6	1	8	109	104	3	23	8	109	105	4	20
Mog	4	8	109	87	3	29	8	109	75	3	32	8	109	104	4	32	0	104	43	10	27	8	109	107	2	28	8	108	104	4	21
Mog	6	8	109	87	3	30	8	109	77	3	32	8	109	102	3	32	3	109	36	3	15	8	109	105	3	32	6	109	106	4	20
MogSC	2																										8	108	106	4	27
MogSC	3																										8	105	105	4	15
MogSC	4																										8	108	101	4	18
MogSC	6																										8	107	101	4	20
MogSP	2																										8	103	83	3	20
MogSP	3																										7	96	84	4	16
MogSP	4																										6	68	54	2	16
MogSP	6																										5	67	46	0	13

TABLE D.5: Correctly classified instances per prototype in the EVS recording, using camera 3 as test set.

	Rgb					Nrgb					Hsv					Hue					Hs					Hsz				
GroundTruth	51	147	172	64	49	51	147	172	64	49	51	147	172	64	49	51	147	172	64	49	51	147	172	64	49	51	147	172	64	49
Random	4	32	31	14	10	4	32	31	14	10	4	32	31	14	10	4	32	31	14	10	4	32	31	14	10	4	32	31	14	10
Histogram 10	2	32	14	28	42	44	109	148	63	44	51	78	56	28	41	38	147	137	12	44	46	147	167	63	41	46	116	85	58	44
Histogram 30	26	25	52	59	11	51	74	14	64	44	51	82	55	59	40	46	147	150	63	43	46	139	156	63	41	51	139	166	59	49
Histogram 100	12	21	95	63	10	51	135	160	64	42	51	83	58	61	39	49	147	150	64	32	45	147	153	64	43	51	146	160	61	49
Histogram 300	47	106	104	61	12	51	147	145	64	44	51	121	91	61	7	50	147	150	64	35	47	147	153	64	42	51	147	164	60	48
Mean	51	118	101	61	15	50	94	153	63	43	51	106	159	61	40	37	67	155	5	36	50	81	162	63	39	51	80	146	62	41
Disks 2	51	115	139	58	17	40	142	169	62	44	51	143	163	58	43	13	137	164	29	44	37	142	167	62	44	45	139	168	59	48
Disks 3	51	113	133	61	17	38	147	165	64	48	51	147	168	61	47	18	143	160	53	45	32	146	170	64	47	36	146	170	64	47
Disks 4	51	115	121	61	18	39	147	162	64	48	51	147	168	61	45	28	132	152	64	41	37	145	170	64	45	37	143	170	64	46
Disks 6	51	113	102	61	20	43	147	155	64	48	51	145	167	63	46	38	102	142	64	25	39	147	170	64	46	39	147	170	64	46
Rings 10	45	111	114	63	30	46	127	155	64	49	51	128	160	62	44	44	126	145	63	23	41	128	161	63	45	43	125	162	63	44
Rings 30	47	129	136	60	27	47	145	155	63	49	51	145	163	62	48	43	135	140	63	26	43	145	162	63	46	43	145	163	63	46
Rings 100	45	123	139	60	27	44	147	155	63	49	51	144	159	62	46	44	131	129	63	31	43	147	157	63	47	43	147	158	63	47
Rings 300	45	132	145	60	27	46	147	156	63	49	51	147	160	62	48	47	136	149	63	44	43	147	159	63	48	43	147	160	63	48
Kmeans 2	44	109	55	56	29	51	124	162	64	40	51	124	152	63	42	14	39	121	51	22	23	124	166	64	42	47	101	147	62	43
Kmeans 3	48	108	49	61	22	48	81	157	64	46	51	141	163	63	41	45	99	146	64	32	47	107	160	64	46	51	135	148	62	45
Kmeans 4	42	117	58	61	26	47	83	157	64	46	51	142	154	63	42	48	50	150	64	31	50	133	167	64	46	50	139	160	63	47
Kmeans 6	47	121	63	61	30	42	133	165	64	44	51	142	157	63	44	5	58	98	62	10	47	126	164	64	47	51	143	163	62	48
Mog 2	51	101	162	52	46	48	125	168	58	43	50	140	159	18	45	11	51	125	16	23	45	147	165	9	45	51	119	168	15	46
Mog 3	51	102	159	54	45	46	127	168	39	43	51	140	153	16	45	40	94	126	64	22	44	146	161	39	43	40	141	159	8	47
Mog 4	51	109	161	57	46	46	127	168	39	43	51	135	153	12	47	45	79	147	64	9	41	146	163	36	44	41	134	156	5	47
Mog 6	51	77	128	60	2	47	131	168	20	44	51	127	139	10	44	47	87	146	61	12	47	142	166	39	44	37	135	143	1	40
MogSC 2																										51	121	166	12	46
MogSC 3																										41	138	150	8	46
MogSC 4																										37	140	141	5	46
MogSC 6																										41	129	136	1	46
MogSP 2																										38	66	118	2	44
MogSP 3																										42	110	74	1	37
MogSP 4																										45	99	53	35	25
MogSP 6																										44	80	24	0	8

TABLE D.6: Correctly classified instances per prototype in the EVS recording, using camera 4 as test set.

	Rgb				Nrgb				Hsv				Hue				Hs				Hsz			
GroundTruth	96	51	31	18	96	51	31	18	96	51	31	18	96	51	31	18	96	51	31	18	96	51	31	18
Random	25	13	6	4	25	13	6	4	25	13	6	4	25	13	6	4	25	13	6	4	25	13	6	4
Histogram 10	2	0	31	6	39	1	6	10	64	8	31	4	68	0	9	13	62	18	10	4	52	4	30	3
Histogram 30	5	6	31	5	73	9	7	1	9	4	31	4	75	15	2	0	79	13	0	4	59	10	24	3
Histogram 100	11	6	31	5	76	20	10	7	18	4	31	4	77	17	0	0	81	15	0	3	71	11	16	6
Histogram 300	3	7	31	5	81	18	1	0	23	5	31	3	78	18	0	0	84	10	0	0	69	10	18	5
Mean	7	3	31	4	59	9	24	5	14	3	30	4	41	15	26	0	60	16	25	3	71	18	25	3
Disks 2	6	7	31	4	34	11	24	0	31	12	30	4	23	9	21	2	29	18	28	4	28	16	26	4
Disks 3	5	4	31	4	39	9	26	5	27	3	31	4	25	9	23	1	36	13	27	5	38	16	27	6
Disks 4	5	2	31	3	45	9	26	5	28	1	31	3	41	7	22	0	44	11	26	8	43	13	26	7
Disks 6	7	4	31	3	48	8	26	4	27	2	31	3	49	4	24	0	46	11	26	7	47	12	26	8
Rings 10	5	4	31	4	73	12	22	0	17	7	31	6	55	17	17	0	66	25	24	2	63	22	24	4
Rings 30	2	4	31	4	68	6	15	8	14	10	31	4	25	3	12	7	72	12	17	9	73	15	17	6
Rings 100	1	9	31	6	68	13	6	4	14	13	31	6	32	21	6	3	68	20	6	3	68	20	7	3
Rings 300	2	10	31	3	60	13	5	5	12	11	31	4	31	23	4	5	63	18	4	3	63	19	4	3
Kmeans 2	2	6	30	4	59	25	2	4	34	7	21	4	7	4	4	6	59	30	12	0	62	21	23	2
Kmeans 3	2	1	31	4	64	14	11	1	57	5	10	4	61	8	1	7	63	12	5	8	66	26	14	2
Kmeans 4	2	3	31	4	70	15	20	7	56	7	14	4	73	3	0	10	64	16	6	6	72	18	22	2
Kmeans 6	2	4	31	4	54	15	27	1	24	7	30	4	32	1	6	8	59	19	22	3	70	23	21	4
Mog 2	7	5	29	0	22	17	25	0	17	5	31	1	13	6	24	3	30	14	27	2	69	6	27	3
Mog 3	4	4	31	0	41	12	22	0	9	8	31	0	48	23	1	4	63	18	11	3	58	9	15	3
Mog 4	2	4	31	0	55	11	15	0	10	4	31	2	71	22	1	3	74	22	5	4	64	9	13	6
Mog 6	3	5	31	0	62	12	5	2	5	5	0	3	42	4	5	4	63	25	16	3	52	11	11	3
MogSC 2																					61	6	27	2
MogSC 3																					53	8	14	2
MogSC 4																					61	9	15	3
MogSC 6																					57	6	9	1
MogSP 2																					56	2	19	2
MogSP 3																					28	9	21	1
MogSP 4																					11	17	13	1
MogSP 6																					23	5	8	1

TABLE D.7: Correctly classified instances per prototype in the PSL recording, using camera 21 as test set.

	Rgb			Nrgb			Hsv			Hue			Hs			Hsz		
GroundTruth	117	0	35	117	0	35	117	0	35	117	0	35	117	0	35	117	0	35
Random	31	0	8	31	0	8	31	0	8	31	0	8	31	0	8	31	0	8
Histogram 10	21	0	15	36	0	9	2	0	27	11	0	19	0	0	32	13	0	29
Histogram 30	21	0	15	4	0	27	2	0	25	7	0	21	2	0	34	12	0	28
Histogram 100	19	0	19	5	0	25	2	0	27	9	0	22	9	0	24	44	0	28
Histogram 300	17	0	20	2	0	30	5	0	26	8	0	21	7	0	28	32	0	30
Mean	14	0	14	2	0	30	12	0	20	4	0	31	20	0	29	44	0	30
Disks 2	19	0	24	4	0	32	10	0	23	1	0	28	3	0	33	3	0	33
Disks 3	17	0	20	6	0	29	14	0	23	2	0	27	4	0	29	11	0	29
Disks 4	15	0	18	7	0	31	8	0	25	3	0	29	7	0	31	6	0	31
Disks 6	11	0	18	12	0	30	6	0	27	1	0	29	12	0	32	13	0	32
Rings 10	16	0	18	7	0	29	22	0	25	3	0	28	32	0	30	34	0	31
Rings 30	22	0	21	10	0	28	19	0	24	3	0	31	30	0	29	30	0	28
Rings 100	23	0	22	13	0	29	20	0	25	4	0	30	30	0	28	30	0	28
Rings 300	24	0	21	11	0	27	19	0	25	5	0	30	26	0	28	26	0	28
Kmeans 2	4	0	24	15	0	28	4	0	18	47	0	2	11	0	23	35	0	25
Kmeans 3	30	0	15	2	0	28	29	0	5	43	0	4	64	0	10	43	0	19
Kmeans 4	9	0	23	7	0	28	4	0	21	58	0	5	18	0	24	50	0	27
Kmeans 6	4	0	23	8	0	30	16	0	24	63	0	1	34	0	22	47	0	26
Mog 2	22	0	22	46	0	24	13	0	20	46	0	1	44	0	23	48	0	19
Mog 3	19	0	23	47	0	24	8	0	15	57	0	5	36	0	19	21	0	8
Mog 4	16	0	25	46	0	22	7	0	16	54	0	8	4	0	0	13	0	6
Mog 6	14	0	25	46	0	18	3	0	11	71	0	3	23	0	17	50	0	18
MogSC 2	50	0	19	50	0	19	50	0	19	50	0	19	50	0	19	50	0	19
MogSC 3	60	0	14	60	0	14	60	0	14	60	0	14	60	0	14	60	0	14
MogSC 4	28	0	4	28	0	4	28	0	4	28	0	4	28	0	4	28	0	4
MogSC 6	24	0	14	24	0	14	24	0	14	24	0	14	24	0	14	24	0	14
MogSP 2																33	0	12
MogSP 3																14	0	16
MogSP 4																9	0	3
MogSP 6																30	0	22

TABLE D.8: Correctly classified instances per prototype in the PSL recording, using camera 29 as test set.

	Rgb				Nrgb				Hsv				Hue				Hs				Hsz			
GroundTruth	103	68	13	14	103	68	13	14	103	68	13	14	103	68	13	14	103	68	13	14	103	68	13	14
Random	22	16	1	2	22	16	1	2	22	16	1	2	22	16	1	2	22	16	1	2	22	16	1	2
Histogram 10	0	2	13	4	90	21	0	1	103	0	0	3	103	0	0	3	87	0	0	0	28	0	0	11
Histogram 30	14	41	12	0	102	6	0	5	102	0	0	4	102	0	0	4	103	0	0	4	39	0	3	0
Histogram 100	27	17	13	0	102	15	0	0	102	0	0	5	102	0	0	5	103	0	0	0	84	0	6	6
Histogram 300	30	15	12	3	102	2	0	4	102	1	0	3	102	0	0	4	102	0	0	0	82	0	7	0
Mean	5	25	13	0	45	1	0	14	35	23	11	2	1	0	0	14	57	1	0	14	54	0	0	12
Disks 2	10	41	11	0	23	1	1	10	59	39	8	4	21	0	0	13	24	2	1	13	21	4	1	11
Disks 3	10	26	11	1	34	1	1	12	57	25	9	4	14	0	0	14	29	2	1	13	30	3	1	11
Disks 4	9	19	11	1	46	1	1	12	54	18	8	5	9	0	0	12	40	2	1	13	41	3	1	11
Disks 6	7	16	11	2	50	0	1	10	45	10	9	5	6	0	0	12	43	0	1	12	45	1	1	12
Rings 10	3	23	13	0	51	2	0	12	30	25	10	3	6	1	0	13	52	1	0	12	58	1	0	12
Rings 30	0	23	11	0	74	4	0	12	46	31	9	3	11	1	0	12	64	4	0	12	63	3	0	12
Rings 100	0	19	10	1	63	4	0	12	32	25	9	3	9	1	1	12	67	0	0	12	66	0	0	12
Rings 300	5	19	10	1	57	9	0	11	33	27	9	3	17	2	0	12	72	5	0	12	72	5	0	12
Kmeans 2	0	17	13	0	82	4	0	7	62	19	5	0	32	0	0	14	81	3	1	13	58	6	6	11
Kmeans 3	0	26	13	0	58	10	0	7	88	28	4	0	60	3	1	11	85	5	0	13	65	5	5	11
Kmeans 4	3	24	13	0	65	13	0	14	48	22	6	2	28	3	2	10	81	6	1	13	69	3	3	10
Kmeans 6	3	29	13	0	65	3	1	12	39	30	7	1	30	0	0	13	66	3	0	14	65	5	0	12
Mog 2	8	26	12	0	30	22	11	4	24	27	11	3	2	0	1	14	47	19	6	14	61	4	6	1
Mog 3	1	29	12	0	36	19	8	5	8	42	0	2	51	0	0	13	59	18	1	14	45	2	2	10
Mog 4	2	28	12	0	45	17	5	5	12	41	12	1	59	0	1	3	37	9	0	6	38	3	1	8
Mog 6	3	25	12	0	80	16	0	5	8	33	1	0	47	1	1	7	46	4	0	9	49	1	1	12
MogSC 2																					77	12	5	11
MogSC 3																					43	5	2	7
MogSC 4																					48	3	1	10
MogSC 6																					43	4	0	11
MogSP 2																					8	6	2	4
MogSP 3																					8	0	2	3
MogSP 4																					11	8	2	3
MogSP 6																					9	0	0	1

TABLE D.9: Correctly classified instances per prototype in the PSL recording, using camera 30 as test set.

	Rgb				Nrgb				Hsv				Hue				Hs				Hsz			
GroundTruth	97	77	12	7	97	77	12	7	97	77	12	7	97	77	12	7	97	77	12	7	97	77	12	7
Random	22	19	3	2	22	19	3	2	22	19	3	2	22	19	3	2	22	19	3	2	22	19	3	2
Histogram 10	11	1	12	7	0	77	0	0	5	2	0	7	0	0	0	7	16	7	0	1	8	10	0	7
Histogram 30	21	7	11	5	6	37	0	7	46	27	0	2	2	37	0	7	0	19	0	7	6	22	0	0
Histogram 100	37	43	0	5	0	70	0	1	26	18	0	7	2	18	2	7	2	71	1	1	1	16	0	4
Histogram 300	20	37	0	4	5	62	0	1	31	34	0	5	1	22	2	7	5	52	0	1	7	33	0	0
Mean	20	77	2	0	10	20	0	0	5	76	0	0	0	54	0	4	6	44	0	0	10	20	0	0
Disks 2	19	4	7	7	12	6	0	0	41	15	0	7	5	27	0	7	4	9	0	1	5	14	0	0
Disks 3	12	8	7	7	16	6	0	0	21	7	0	7	2	18	0	7	6	13	0	1	6	13	0	1
Disks 4	13	13	6	7	19	6	0	0	15	10	0	7	5	16	0	5	7	17	0	0	10	19	0	1
Disks 6	9	56	6	1	28	5	0	0	12	45	0	3	10	19	0	4	10	21	0	1	10	23	0	1
Rings 10	5	46	3	0	27	5	0	0	4	48	0	0	7	25	0	4	10	28	0	0	10	23	0	0
Rings 30	8	31	5	1	23	4	0	0	2	40	0	1	2	27	0	4	11	19	0	0	12	19	0	0
Rings 100	19	21	4	0	28	15	0	0	9	33	0	0	5	39	0	4	12	32	0	0	12	32	0	0
Rings 300	23	18	4	0	26	14	0	0	12	23	0	0	3	32	0	3	11	30	0	0	11	30	0	0
Kmeans 2	7	59	1	2	3	39	0	1	2	67	1	0	12	7	1	0	8	64	0	3	9	34	0	0
Kmeans 3	13	42	0	3	4	39	0	4	0	54	0	4	8	8	7	0	5	64	0	0	9	30	0	0
Kmeans 4	12	48	1	4	5	44	0	2	0	43	0	4	0	31	0	7	3	63	0	3	7	41	0	3
Kmeans 6	12	56	0	4	2	56	0	3	1	52	0	5	26	31	0	4	4	56	0	1	5	37	0	1
Mog 2	1	44	0	2	11	21	0	0	10	69	0	0	11	8	7	2	9	67	0	0	17	46	0	0
Mog 3	2	17	0	2	11	30	0	0	8	65	0	0	36	13	6	3	8	77	0	0	16	55	0	0
Mog 4	1	8	0	0	11	39	0	0	0	8	0	0	3	7	0	7	7	75	0	0	20	48	0	0
Mog 6	3	51	0	0	11	61	0	0	1	60	0	0	39	0	3	4	3	73	0	0	26	47	0	0
MogSC 2																					15	46	0	0
MogSC 3																					15	59	0	0
MogSC 4																					20	55	0	0
MogSC 6																					21	57	0	0
MogSP 2																					19	14	0	0
MogSP 3																					17	1	0	0
MogSP 4																					5	2	0	0
MogSP 6																					5	30	0	1

TABLE D.10: Correctly classified instances per prototype in the PSL recording, using camera 33 as test set.

GroundTruth	Rgb					Nrgb					Hsv					Hue					Hs					Hsz				
	111	43	33	11	1	111	43	33	11	1	111	43	33	11	1	111	43	33	11	1	111	43	33	11	1	111	43	33	11	1
Random	27	11	2	2	1	27	11	2	2	1	27	11	2	2	1	27	11	2	2	1	27	11	2	2	1	27	11	2	2	1
Histogram 10	6	7	30	4		59	19	1	7		93	0	0	4		68	1	0	10		58	0	0	11		62	1	0	5	
Histogram 30	32	9	28	4		70	28	0	7		79	5	0	4		68	12	0	11		81	0	0	7		59	0	0	9	
Histogram 100	15	10	24	5		82	15	0	7		80	7	0	6		72	2	1	11		91	2	1	8		83	0	2	11	
Histogram 300	23	10	23	6		75	12	1	11		84	10	2	6		72	2	0	10		86	8	0	11		71	1	1	9	
Mean	19	9	29	4		45	31	0	2		44	11	2	4		36	13	0	9		59	8	0	9		85	28	0	0	
Disks 2	37	6	30	4		23	14	1	4		52	17	13	3		16	3	0	8		21	8	1	6		18	12	1	2	
Disks 3	41	8	29	6		29	12	1	4		50	10	15	6		13	3	0	7		27	11	0	7		27	17	0	2	
Disks 4	29	9	29	5		35	20	0	1		45	12	7	5		17	6	0	6		34	11	0	6		35	14	0	4	
Disks 6	29	11	29	4		42	16	1	3		36	16	11	4		24	6	0	7		45	17	0	6		45	15	0	7	
Rings 10	19	10	29	6		73	29	0	1		35	21	6	7		28	24	0	3		69	23	0	5		71	34	0	2	
Rings 30	14	15	31	5		70	19	0	3		37	32	21	5		18	9	0	8		68	21	0	4		70	24	0	2	
Rings 100	12	15	31	6		78	26	0	5		38	26	13	7		27	27	0	4		73	27	0	5		73	27	0	5	
Rings 300	6	18	30	4		75	16	0	5		27	26	20	5		23	15	0	7		71	22	0	5		71	22	0	5	
Kmeans 2	8	4	31	2		54	16	0	5		45	14	0	3		2	11	10	3		53	24	1	5		28	23	3	3	
Kmeans 3	12	6	29	4		38	22	0	3		37	16	1	4		72	1	1	10		59	30	0	3		76	24	1	6	
Kmeans 4	11	6	30	4		75	19	0	3		68	21	1	4		52	9	0	11		61	14	0	6		81	25	1	3	
Kmeans 6	11	6	32	4		54	23	2	4		38	20	2	4		32	9	0	11		49	29	1	8		77	26	1	2	
Mog 2	13	18	24	4		4	31	4	3		24	25	15	4		5	15	6	5		32	34	1	4		49	13	1	2	
Mog 3	4	16	25	0		4	31	1	3		12	23	12	0		53	6	4	4		49	32	1	4		39	10	1	2	
Mog 4	5	16	26	2		9	30	3	3		16	15	13	0		78	4	0	10		18	13	0	4		7	11	0	3	
Mog 6	0	10	0	0		15	29	2	4		2	18	20	1		43	4	0	11		58	14	0	4		7	9	2	3	
MogSC 2																										41	17	1	3	
MogSC 3																										38	14	1	2	
MogSC 4																										81	18	0	3	
MogSC 6																										84	16	2	3	
MogSP 2																										18	8	0	2	
MogSP 3																										16	3	5	1	
MogSP 4																										9	2	0	2	
MogSP 6																										15	0	0	2	

TABLE D.11: Correctly classified instances per prototype in the PSL recording, using camera 39 as test set.

List of Figures

List of Tables

Bibliography

[1] Wojciech Zajdel, A. Taylan Cemgil, and Ben Kröse. A hybrid graphical model for online multicamera tracking. Technical report, Intelligent Autonomous Systems, University of Amsterdam, Kruislaan 403, 1098SJ Amsterdam, The Netherlands, 2004. URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.4450&rep=rep1&type=pdf`.

[2] Mark S. Drew, Jie Wei, and Ze-Nian Li. Illumination-invariant color object recognition via compressed chromaticity histograms of color-channel-normalized images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 533–540, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9. URL `http://www.cs.sfu.ca/~li/papers-on-line/Drew-ICCV-98.pdf`.

[3] Gwenn Englebienne, Tim van Oosterhout, and Ben Krose. Tracking in sparse multi-camera setups using stereo vision. pages 1–6, Augustus 2009. URL `http://dx.doi.org/10.1109/ICDSC.2009.5289371`.

[4] T. Gevers. Color in image search engines. In *Principals of Visual Information Retrieval*, London, UK, 2001. Springer-Verlag. ISBN 1-852333-391-2. URL `http://staff.science.uva.nl/~gevers/pub/survey_color.pdf`.

[5] Edward H. Adelson. *Checkershadow Illusion*, 1995. URL `http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html`.

[6] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *Proceedings of the 8th International Conference on Database Theory*, ICDT '01, pages 420–434, London, United Kingdom, 2001. Springer-Verlag. URL `http://bib.dbvis.de/uploadedFiles/155.pdf`.

[7] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[8] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5. URL `http://www.stanford.edu/~darthur/kMeansPlusPlus.pdf`.

[9] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, April 1936. URL `http://www.new.dli.ernet.in/rawdataupload/upload/insa/INSA_1/20006193_49.pdf`.

[10] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:49–86, 1951. URL `http://www.csee.wvu.edu/~xinl/library/papers/math/statistics/Kullback_Leibler_1951.pdf`.

[11] T. Gevers and A. Smeulders. Color based object recognition. *Pattern Recognition*, 32:453–464, 1997. URL `http://staff.science.uva.nl/~gevers/pub/GeversPR99.pdf`.