Sensor Fusion for Dynamic Privacy Masking

Änis Ben Hamida and Mikael Pendse

Master's thesis
2013:E45

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

Centrum Scientiarum Mathematicarum

Lund University

Spring 2013

Sensor Fusion for Dynamic Privacy Masking

Änis Ben Hamida

Mikael Pendse

August 16, 2013

Abstract

This master thesis investigates the possibility of combining a regular visual surveillance camera with a thermal infrared surveillance camera monitoring the same view of a scene to detect and classify heat radiating objects such as human beings. A method is presented for the application of privacy masking of a window while detecting heat radiating objects between the cameras and the window and preventing them from being masked out. The proposed method is based on determining if an object is present in both the visual and the IR frame. Registration of image pairs is done using thin plate spline interpolation. Foreground segmentation is done using a Mixture of Gaussians method. The proposed method looks at connected components from the foreground segmentation and for each component determines if it should be excluded from the mask. Classification is done by thresholding scores obtained by matching features in corresponding IR-visual frame pairs. Three measures for classifying heat radiating objects and reflections in an IR image are also proposed. The classification routine, when combined with the proposed measures, achieves a 98.9% true positive rate and a true negative rate of 99.7%.

Preface

This is the result of our masters thesis project at LTH, Faculty of Engineering, Lund University. The project was done in association with Axis Communications AB and conducted at their offices in Lund. Axis Communications develops and manufactures network surveillance cameras. We would like to thank everybody at the Product Concepts and New Ideas as well as the Analytics & Systems department that have helped us and made us feel welcome. Especially we would like to thank our supervisors, Willy Sagefalk and Gustav Träff, for all the support and guidance. We would also like to thank our supervisors at LTH, Kalle Åström and Magnus Oskarsson, for all their input and feedback.

Contents

1 Introduction

1.1 Background

All objects with a temperature above absolute zero emit heat radiation. Some of this radiation lies in the long wave infrared (LWIR) spectrum and can be visualized using a thermographic camera. The human body is especially interesting for surveillance and emits plenty of radiation in the LWIR spectrum making the use of thermographic cameras interesting for surveillance purposes. For example, they can make surveillance of a scene possible without any external illumination. These cameras have been quite expensive, but as they become cheaper the concept of combining information from an infrared and a visual camera becomes a commercially viable option.

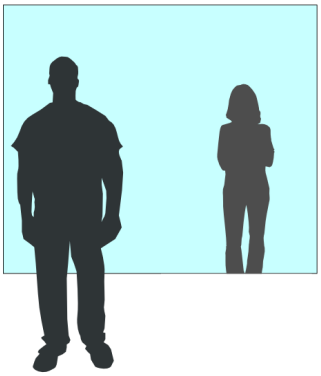
In video surveillance privacy masking is the concept of masking areas by completely blocking or blurring a particular area. A privacy mask can for instance be placed over an ATM keypad to protect pin-codes. Another application is to mask windows where the area on the other side of the window is not allowed to be visible in the video feed. Imagine a lobby with street view windows where you want to have surveillance in the lobby, but need to protect the privacy of passing pedestrians for legal reasons. A privacy mask would then be placed over the window so that the video operator can not identify who is walking on the other side of the window. In the conventional approach everything in the line of sight between the camera and the window is masked out. It would be desirable to not mask people that walk by in front of the window inside the lobby.

The problem is illustrated in figure 1. The scene contains a window over which a privacy mask is placed to blur what is happening behind the window. Figure 1c shows conventional privacy masking and figure 1d shows the desired result.

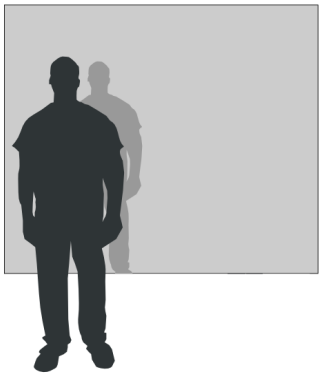
In the LWIR spectrum glass is not transparent with the consequence that a regular window acts as a mirror for these wavelengths, so a person standing on the far side of a window is not visible in a LWIR camera. To illustrate the properties of glass in LWIR, the IR frame of the scene (where only the person in front of the window and the reflection of him are visible) can be seen in figure 1b.

1.2 Aim of the thesis

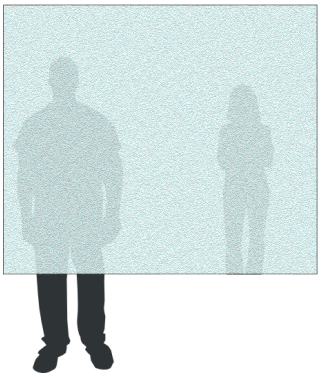
The aim of the thesis is to investigate the possibilities of combining traditional surveillance cameras with thermographic cameras to be used to improve detection of human beings and other heat radiating objects. The application that this thesis focuses on is that of preventing objects on the near side of a privacy masked window to be masked out. The goal is to produce a method to automatically detect humans and heat radiating objects in front of such a privacy mask.



(a) Visual frame



(b) IR frame



(c) Conventional privacy masking

(d) Modified privacy masking

Figure 1: A visual and IR frame of a scene depicting a window, a person between the cameras and the window and a person on the far side of the window. Notice that the person on the far side of the window is not visible in the IR frame. Notice also the thermal reflection of the person in front of the window in the IR frame.

We also want to explore what effect lower resolutions of the infrared camera has on the performance of such algorithms.

1.3 Overview of thesis

In section 2 we introduce some theory and known methods used in the thesis. In section 3 we proceed by presenting the approaches we have taken and the methods we have developed. Section 4 consist of results from applying our methods on gathered test data and evaluation of the methods with some discussion of the results. Lastly we have sections 5 and 6 in which we draw conclusions of the results and discuss improvements, future work and applications.

2 Theory

2.1 Long Wave Infrared Radiation and Thermal Imaging

Infrared radiation is electromagnetic radiation with wavelengths longer than visible light and shorter than microwaves which is in the range of about 700 nm to 1 mm. This corresponds to frequencies of 430 THz to 300 GHz. The IR spectrum is often subdivided into Near Infrared (NIR) — 0.75–1.4 µm, Short Wave Infrared (SWIR) — 1.4–3 µm, Mid Wave Infrared (MWIR) — 3–8 µm, Long Wave Infrared (LWIR) — 8–15 µm and Far Infrared (FIR) — 15–1,000 µm [2]. The IR radiation exhibit some different properties depending on which subspectrum the radiation comes from. The most commonly used spectrum for thermal imaging is the LWIR since objects at around room temperature mostly emit radiation in the 8–25 µm range.

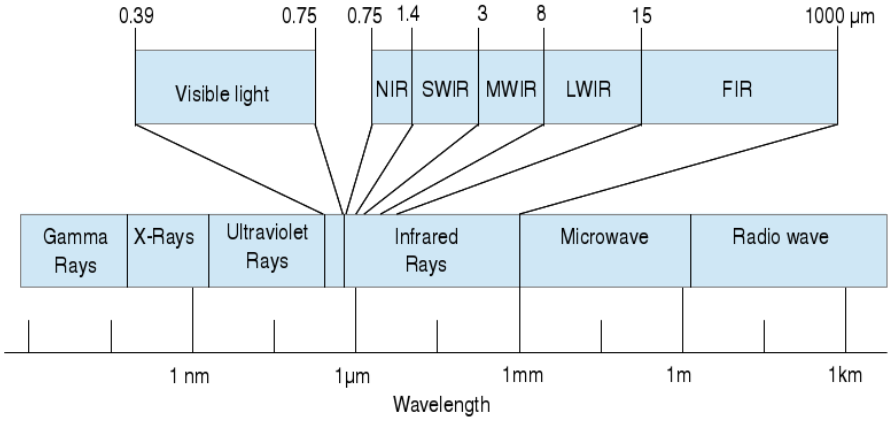


Figure 2: Electromagnetic spectrum

In a thermographic camera a sensor is used to detect the amount of radiation in a scene and produce a thermal image, a so called *thermogram*. False color is used to display the thermogram since the IR radiation itself is not visible to the human eye. Most commonly a gray scale map is used where whiter pixels represent more IR radiation.

All objects above absolute zero temperature emit electromagnetic radiation that comes from the conversion of an object's thermal energy into electromagnetic energy. The process is called thermal radiation and is not restricted to infrared light, although as mentioned most of the radiation at room temperature takes place in the infrared region.

The total amount of detected radiation is not just the thermal radiation from the object itself but also the electromagnetic waves that are transmitted and reflected which does not necessarily mean the object itself is hot. Since the thermographic camera relies on the amount of radiation it detects, it is not always true that warmer objects appear brighter in a thermogram. The amount of thermal radiation from an object itself depends on its *emissivity*, ϵ. The emissivity can be described as the amount of energy radiated by a particular material, where $\epsilon = 1$ means the object has no transmission or reflection, and the detected radiation is only dependent on its temperature [7]. (Such objects are called *black bodies* and are idealized objects, whereas all real world objects have $\epsilon < 1$.)

Many materials that reflect visible light also reflect LWIR radiation such as some plastics with a smooth surface or polished metals. Some materials are significantly more reflective in the LWIR spectrum such as a smooth floor and especially glass. It should be noted that the visibility of the light that is reflected on the glass is dependent on lighting conditions because of the transparency of glass. When it is dark outside more reflections are seen, while if it is bright outside the incoming light dominates the reflected light.

2.2 Image registration

Image registration is the process of aligning two or more images by a geometric transformation so that points in the scene appear at the same image coordinates in all images.

2.2.1 The Parallax Error

When looking at two images of the same scene, taken from different cameras, the so called *parallax error* arises from looking at objects from different angles. The objects appear to be in different positions depending on which line of sight they are viewed along as seen in figure 3. The parallax itself is the separation between the two observers or cameras. This is obviously an important phenomenon when

working with image registration. The parallax error depends on the depth of the objects making it impossible to find a transformation to correctly register images where objects are found at arbitrary distances from the cameras. The projection of scene points to image pixels is not injective and the *projective ambiguity* is illustrated in figure 4.

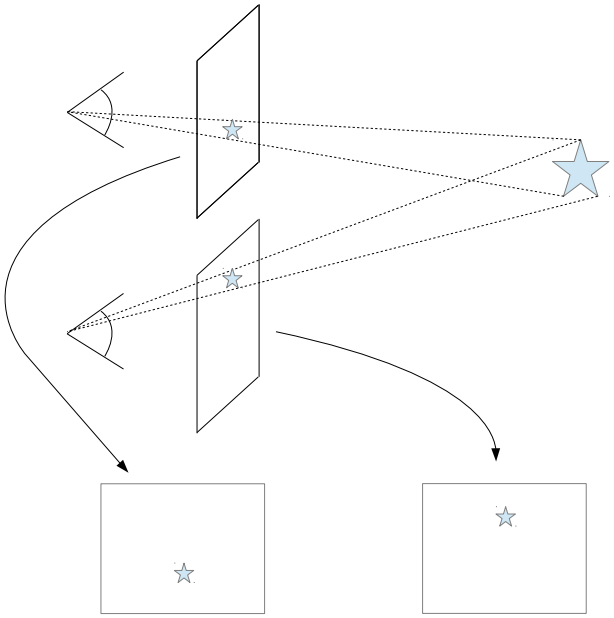


Figure 3: The parallax effect

Without loss of generality consider the parallax problem in two dimensions as depicted in figure 5. The parallax error is often also called *binocular disparity* and refers to the difference in coordinates of a point correspondence in two stereo images. Using the notation in figure 5 (where for instance $\overline{AB}$ denotes the distance from point A to point B) an expression for the binocular disparity is derived. A and B are the two camera centers and f_1 and f_2 are their respective focal lengths. C is the scene point for which the disparity is calculated. h is the parallax (the distance between A and B), d and a are the horizontal and vertical distances between A and C respectively. x_1 and x_2 are the one-dimensional image coordinates of camera A and B respectively that the scene point is projected to. Since the two triangles ΔBIK and ΔBGC are similar the ratios between the length of two sides are equal,

$$\frac{h+a}{d} = \frac{x_2}{f_2} \Leftrightarrow x_2 = f_2 \cdot \frac{h+a}{d}.$$

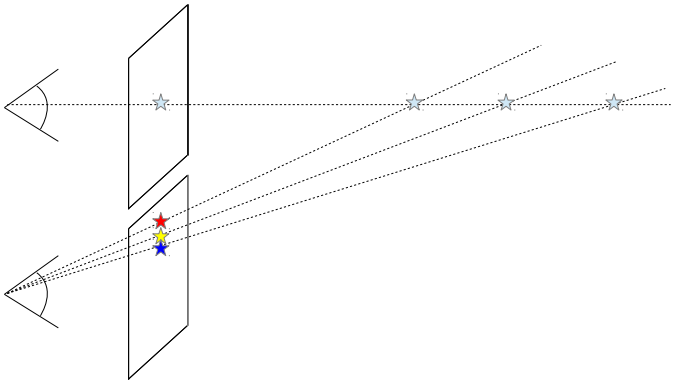


Figure 4: In the upper camera objects at different depths are mapped to the same pixel but the same objects are mapped to different pixels in the lower camera. This is the projective ambiguity.

Using ΔADE and ΔAGC,

$$\frac{a}{d} = \frac{x_1}{f_1} \Leftrightarrow x_1 = f_1 \cdot \frac{a}{d}.$$

The binocular disparity is $|x_2 - x_1| = |f_2 \cdot \frac{h+a}{d} - f_1 \cdot \frac{a}{d}|$. Since the images might have different resolutions, relative positions are calculated as $x_1' = \frac{x_1}{m_1}$ and $x_2' = \frac{x_2}{m_2}$. From the figure

$$\tan\alpha = \frac{m_1}{f_1} \Rightarrow m_1 = \frac{f_1}{\tan\alpha}$$

and

$$\tan\beta = \frac{m_2}{f_2} \Rightarrow m_2 = \frac{f_2}{\tan\beta}.$$

Inserting into x_1'

$$x_1' = \frac{x_1}{m_1} = \frac{a}{d\tan\alpha}$$

and x_2'

$$x_2' = \frac{x_2}{m_2} = \frac{h+a}{d\tan\beta}.$$

The binocular disparity in percentage of the image height (or width) is thus

$$x_2' - x_1' = \frac{(h+a)\tan\alpha - a\tan\beta}{d\tan\beta\tan\alpha}.$$

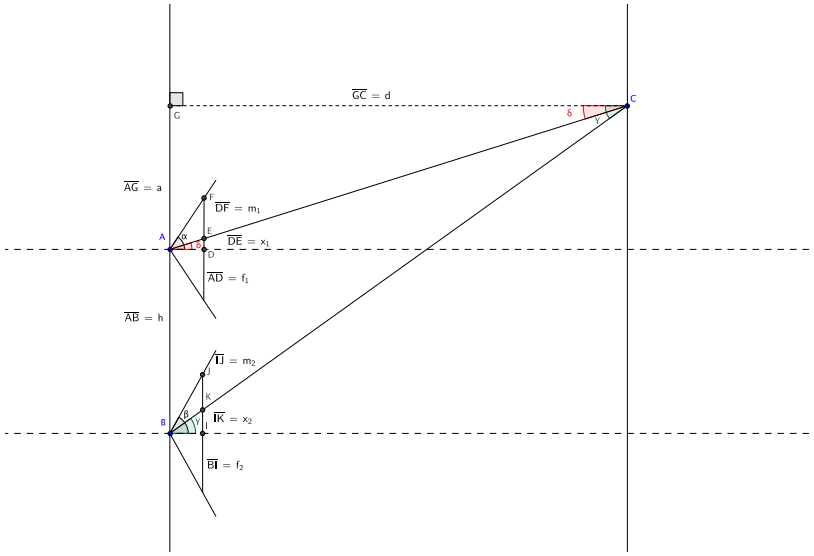


Figure 5: Binocular disparity (parallax error in two dimensions)

2.2.2 Thin Plate Spline Warping

One method for image registration is based on *Thin Plate Splines* (TPS). It has been studied and refined by for example [1]. The method can be physically thought of as the name suggests by considering a thin metal plate which is extended infinitely in space. A finite number of points are pinned to fixed heights and the plate will bend into the shape that minimizes its bending energy.

To construct a TPS mapping, let $P_1 = (x_1, y_1), P_2 = (x_2, y_2), \ldots, P_n = (x_n, y_n)$ be n landmark points in a Cartesian coordinate system. The goal is a function $f(x, y)$ taking fixed values, v_i, at the points P_i, (the pinned heights). This function shall minimize the bending energy

$$E_f = \iint_{\mathbb{R}^2} \left(\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right) \mathrm{d}x \, \mathrm{d}y.$$

As shown in [1] this is done with functions of the form

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i U(|P_i - (x, y)|), \tag{1}$$

where $U(|(x, y)|) = U(r) = r^2 \log r$. It is also necessary that

$$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} w_i y_i = 0.$$

The coefficients a_1, a_x, a_y, w_i are then determined by solving the linear system of equations

$$\begin{bmatrix} K & P \\ P^T & O \end{bmatrix} \begin{bmatrix} W \\ A \end{bmatrix} = \begin{bmatrix} V \\ o \end{bmatrix},$$

where $K_{ij} = U(|P_i - P_j|)$, row i of P is $(1, x_i, y_i)$, O is a 3×3 matrix of zeroes, o is a 3×1 vector of zeroes, W and V are column vectors formed from w_i and v_i respectively and $A = (a_1, a_x, a_y)^T$.

In image registration points are mapped to an image I_1 with coordinates (x, y) from another image I_2 with coordinates (x', y'). A finite number of point correspondences between the images $(x_i, y_i), (x_i', y_i')$ are chosen as landmarks. These landmarks are used to calculate two TPS mappings, one for each coordinate, i.e. first $v_i = x_i'$ is used to calculate the coefficients for a function $f_x(x, y) = \tilde{x}$ of the form in equation 1 and then symmetrically with $v_i = y_i'$ to get a function $f_y(x, y) = \tilde{y}$.

The vector valued interpolating function $f(x, y) = [f_x(x, y), f_y(x, y)] = (\tilde{x}, \tilde{y})$ is used to align the images. As seen in equation 1, the mapping has an affine part defined by A and a nonlinear part.

Since images have discrete pixel coordinate values, the obtained warped image coordinates $(\tilde{x}, \tilde{y})$ are rounded. I_2 is aligned with I_1 by evaluating the pixel intensities of I_2 at the rounded coordinates to get a registered image $\tilde{I_2}$ where $\tilde{I_2}(x, y) = I_2(\lfloor (f(x, y)) \rceil) = I_2(\lfloor \tilde{x}, \tilde{y} \rceil)$[1].

Because of the discretization the mapping is not necessarily injective when warped coordinates are rounded to the same values. This can be dealt with using numerous interpolation techniques.

2.3 Foreground segmentation

Foreground segmentation is the task of classifying an image into foreground objects and a background. We give a theoretical background to two techniques for performing foreground segmentation in a video sequence that we have tried and used.

2.3.1 Mixture of Gaussians method

One of the most used methods for motion detection and real time tracking is background subtraction which in essence means that the current frame is subtracted with an image of the static background to detect the differences. A threshold is then put on these differences and the result can be classified as foreground objects. Some of the problems that arise when obtaining an estimate

[1] $\lfloor a \rceil$ denotes the rounding of a to the nearest integer

of the background image are coping with small movements of the background objects, introduction of new objects in the background and illumination changes just to name a few. There is therefore a need for an adaptive model that can gradually learn the background and adapt to changes in it. One of the most popular models in recent times is the Gaussian mixture model proposed by [8] but the one we will use is the improved version proposed by [3]. We will first present the original version and then present the modified update equations in [3].

Each pixel in the image is modeled by a mixture of K Gaussian distributions. The probability that a particular pixel is observed at time t is then

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}),$$

where K is the number of distributions, $\omega_{i,t}$ is the estimated weight for the ith distribution at time t, $\mu_{i,t}$ is the estimated expectation value for the ith distribution at time t and $\Sigma_{i,t}$ is the covariance matrix of the ith distribution at time t. The probability density function of the Gaussian distribution has the form

$$\eta(X_t, \mu_t, \Sigma_t) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)},$$

where n is the number of color channels(3 for RGB images and 1 for gray-scale images). K is chosen depending on the available memory and computational power. The covariance matrix is assumed to be diagonal, i.e. $\Sigma_{i,t} = \sigma_{i,t}^2 I$ where $\sigma_{i,t}^2$ is the variance parameter of the ith Gaussian at time t. This means that the color channels are independent of each other and have the same variance. This assumption may not be accurate but it saves us computational power since the inverse of the covariance matrix simply becomes $\Sigma_{i,t}^{-1} = \frac{1}{\sigma_{i,t}^2} I$.

The pixel value X_t is at each iteration checked against the K different models for a match which is determined by how close X_t is to the distribution in terms of standard deviations. That is, a match is defined as being within n number of standard deviations of a distribution's (according to [8] n should be chosen as 2.5). If no match is found among any of the K distributions, the least probable distribution is replaced by a new distribution with a mean equal to the current value, a high variance and a low weight.

The weights of the K distributions are updated at each iteration as follows

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}),$$

where α is the learning rate and $M_{k,t}$ is equal to 1 for the model that matched and equal to 0 for the other models. After the update the weights are renormalized to make their sum equal to 1. The distribution parameters are only updated for the model that matched in which case they are updated as follows

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t),$$

where

$$\rho = \alpha\eta(X_t|\mu_{t-1}, \sigma_{t-1}).$$

To estimate the background model the K distributions are ordered by their fitness values ω_k/σ_k and the first B distributions are chosen as the background model where

$$B = \arg\min_b \left(\sum_{k=1}^{b} \omega_k > T \right).$$

It is stated by the original paper [8] that α and T are the only parameters needed to be set. $\frac{1}{\alpha}$ is the time constants which determines change and T is the minimum fraction of the background model.

In [3], the Gaussian mixture model is at first estimated by using the expected sufficient statistic update equations,

$$\omega_{k,t} = \omega_{k,t-1} + \frac{1}{N+1}\left(M_{k,t} - \omega_{k,t-1}\right)$$

$$\mu_{k,t} = \mu_{k,t-1} + \frac{M_{k,t}}{\sum_{i=1}^{t} M_{k,i}}\left(X_t - \mu_{k,t-1}\right)$$

$$\Sigma_{k,t} = \Sigma_{k,t-1} + \frac{M_{k,t}}{\sum_{i=1}^{t} M_{k,i}}\left(\left(X_i - \mu_{k,t-1}\right)\left(X_i - \mu_{k,t-1}\right)^T - \Sigma_{k,t-1}\right).$$

Note that the condition that $\Sigma_{i,t} = \sigma_{i,t}^2 I$ has been dropped since the update equation above corresponding to Σ does not necessarily render it diagonal. After the first L samples have been processed, an L-recent window version of the above is used,

$$\omega_{k,t} = \omega_{k,t-1} + \frac{1}{L}\left(M_{k,t} - \omega_{k,t-1}\right)$$

$$\mu_{k,t} = \mu_{k,t-1} + \frac{1}{L}\left(\frac{M_{k,t}}{\omega_{k,t}} X_t - \mu_{k,t-1}\right)$$

$$\Sigma_{k,t} = \Sigma_{k,t-1} + \frac{1}{L}\left(\frac{M_{k,t}}{\omega_{k,t}}\left(X_i - \mu_{k,t-1}\right)\left(X_i - \mu_{k,t-1}\right)^T - \Sigma_{k,t-1}\right).$$

Another improvement proposed by [3] is the incorporation of shadow detection to avoid shadows in the visual spectrum being detected as foreground.

2.3.2 Eigenbackground Subtraction

A method that is described in [5] is the Eigenbackground subtraction method. It relies on creating an eigenspace model and then using principal component analysis(PCA) to reduce the dimensionality of the space. The idea is that the eigenspace is built in such a way that it models static portions of the image well while dynamic objects are not modeled very well. The way it works is that an average is taken over a number of sample images meaning that moving objects will not contribute much to the model as long as there are enough samples.

The method can be divided into two phases, the learning phase and the classification phase. In the learning phase a sample of N images are used to compute the average image μ_b. All the images are then mean-subtracted and column stacked in a $p \times n$ matrix A, where p is the number of pixels in each image. The covariance matrix is then computed by $C_b = AA^T$. To reduce the dimensionality the covariance matrix is diagonalized using eigenvalue decomposition $L_b = \phi_b C_b \phi_b^T$, where L_b is the diagonal matrix containing the eigenvalues of C_b and ϕ_b is a matrix containing the corresponding eigenvectors (eigenbackgrounds). Then only the M largest eigenvalues and their corresponding eigenvectors are kept to form a $p \times M$ matrix ϕ_{M_b} with lower dimensionality.

In the classification phase the image I that is to be classified is projected onto the eigenspace as $I' = \phi_{M_b}^T(I - \mu_b)$ and then reprojected back onto the image space as $I'' = \phi_{M_b} I' + \mu_b$. Since the eigenspace is a good model for the static scene but not for the small moving objects, I'' will ideally not contain such objects. Foreground objects are then obtained by thresholding the absolute difference between I'' and the original image. In other words foreground pixels are those who satisfy $|I - I''| > T$ for some threshold T.

2.4 Mathematical Morphology

Mathematical morphology is a theory extensively used to process binary images, e.g. obtained from foreground segmentation. Morphology uses set theory to define *morphological operations* which modify geometrical structures found in binary images. These operations are based on a *structuring element*, a binary shape with a defined *reference pixel*.

2.4.1 Erosion and Dilation

The two most basic morphological operators are *dilation* and *erosion*. Erosion is defined as

$$A \ominus B = \{z \in \mathbb{Z}^2 | B_z \subseteq A\},$$

where B is the structuring element and A is the union of the objects in a binary image (pixels with value 1) which also is a subset of the integer grid $\mathbb{Z}^2$. B_z denotes the pixel $z \in \mathbb{Z}^2$ where the reference pixel is placed. In other words, erosion is the set of all pixels for which B placed at that pixel is contained in A. Erosion has the effect on objects (connected components) in the binary image getting thinned in each direction depending on the thickness of the structuring element in that direction. Erosion can be used for instance to remove small objects arising from noise, see figure 6c.

Dilation, which can be seen as the complementary operation to erosion, is the other basic morphological operator and can simply be defined as the erosion of the complement to A, $A \oplus B = (A^c \ominus B)^c$. To define it independently of the erosion first define $\hat{B}$ as the reflection of B , i.e. the rotation of B 180° around

the reference pixel. Dilation is the set defined by the union of all structural elements $\hat{B}_z$ with reference pixel in A,

$$A \oplus B = \bigcup_{z \in A} \hat{B}_z = \{z \in \mathbb{Z}^2 | (\hat{B}_z \cap A \ \neq \emptyset\},$$

Dilation has the opposite effect to erosion in that it thickens objects instead of thinning them. Dilation can be used to give a smoothing effect and to fill out small holes in objects, see figure 6d.

2.4.2 Opening and Closing

Two common operations based on the previous two are *opening* and *closing*. The opening of a binary image A by the structural element B is an erosion followed by a dilation,

$$A \circ B = (A \ominus B) \oplus B.$$

It is often useful to apply an opening to a binary image to remove noise but still keep the approximate size of the larger objects. Erosion removes objects smaller than the structuring element and the remaining objects regain their approximate size during dilation. See figure 6e for an example of an opening.

A closing is a dilation followed by an erosion,

$$A \bullet B = (A \oplus B) \ominus B.$$

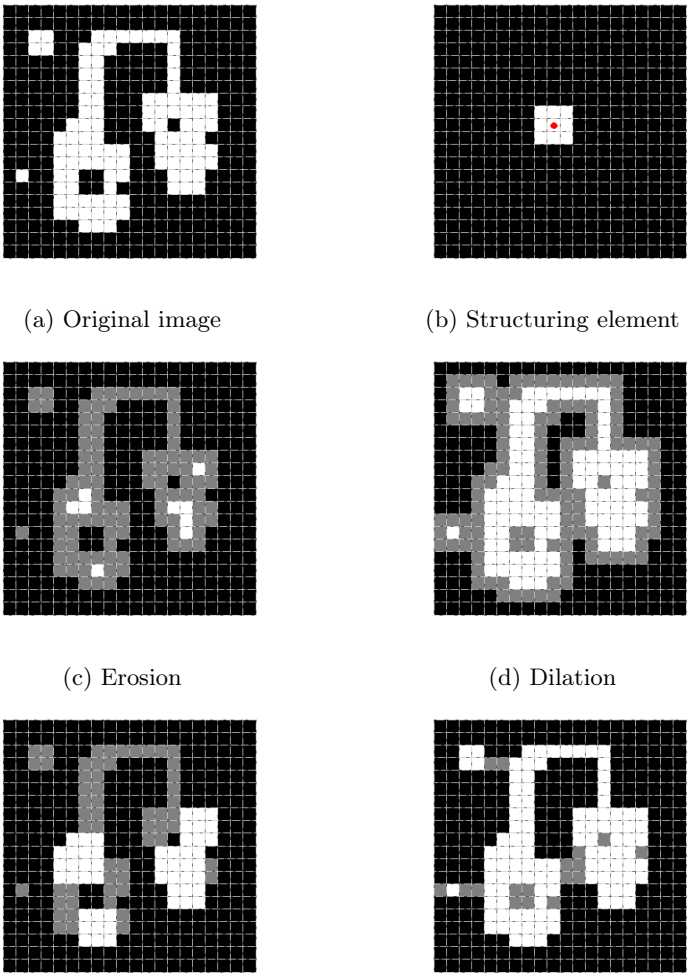
It has a smoothing effect while keeping the approximate size of objects. During dilation, holes smaller than the structuring element are filled and similarly roughness along the border of objects is smoothed. Objects gain in size from dilation and thus an erosion is done to thin them back to approximately their original size. The holes and the roughness do not reappear since all information about the position of the holes and the roughness structure has been lost. See figure 6f for an example of a closing.

2.5 K-Means Clustering Algorithm

Presented here is the K-means clustering algorithm as introduced by James MacQueen in his paper from 1967 ([4]). The K-means algorithm aims to cluster n number of observations, $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ (where $\mathbf{x_i}$ is of any dimension), into a fixed number k clusters, $\{C_1, C_2, \ldots, C_k\}$, where each observation belongs to the cluster with the nearest mean. It does so by trying to minimize the sum of squares,

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2,$$

with respect to the clusters. μ_i denotes the cluster center of C_i and is equal to the mean of the points in C_i.



(a) Original image

(b) Structuring element

(c) Erosion

(d) Dilation

(e) Opening

(f) Closing

Figure 6: Morphological operators. Gray pixels indicate pixels that have changed compared to the original image. In case of erosion and opening gray pixels indicate removed pixels and in case of dilation and closing they indicate added pixels. The red dot in the structuring element indicates the reference pixel.

The algorithm can be initiated by placing the cluster centers at random. In the assignment step, all points are iterated through and assigned to the nearest cluster. In the update step, all the cluster centers are recomputed. The process is repeated until the algorithm is thought to have converged. The algorithm is summarized in algorithm 1.

Algorithm 1 The basic K-means algorithm

Initialize the clusters $\{C_1, C_2, \ldots, C_k\}$

while algorithm has not converged **do**
  for $i = 1$ **to** n **do**
    $C_i = \{x_p \quad | \quad \|x_p - \mu_i\| \leq \|x_p - \mu_j\| \, \forall \, 1 \leq j \leq k\}$
  end for

  for $i = 1$ **to** n **do**
    $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
  end for

end while

return $\{C_1, C_2, \ldots, C_k\}$

There are different termination criteria to determine convergence such as when the clusters no longer change (does not necessarily happen). We have just presented the basic idea for the k-means algorithm, but various modifications for improvements have been proposed through the years. In particular, there is an extensive amount of initialization methods with the most common being The Random Partition method which simply assigns a cluster to each observation at random and then proceeds to the update step.

K-means can be used in image analysis to segment objects in images. Pixel intensities are used as observations and the algorithm tries to assign the pixels with similar color intensities into the same cluster.

2.6 Evaluation Tools

2.6.1 Classification Rates

When using a binary classifier to classify objects as belonging to either A or B where A is defined as the positive class, there are the concepts

- *True positives* (TP) – Objects belonging to A and classified as A.
- *False positives* (FP) – Objects belonging to B and classified as A.
- *True negatives* (TN) – Objects belonging to B and classified as B.

- *False negatives* (FN) – Objects belonging to A and classified as B.

To compare results from such classification, one defines the rates

- *True positive rate* (TPR) $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$.
- *False positive rate* (FPR) $= \frac{\mathrm{FP}}{\mathrm{FP+TN}}$.
- *True negative rate* (TNR) $= \frac{\mathrm{TN}}{\mathrm{FP+TN}}$.
- *False negative rate* (FNR) $= \frac{\mathrm{FN}}{\mathrm{TP+FN}}$.

The TPR measures how good the classifier is at classifying class A objects and the FPR measures how good it is at classifying class B objects. (Since the rates pairwise sum up to one, only two are necessary.) Notice that TPR + FNR = FPR + TNR = 1, thus it is sufficient to use for instance only TPR and TNR since the other two provide no additional information.

2.6.2 Receiver Operating Characteristic

The *receiver operating characteristic* (ROC) is a graph that illustrates the performance of binary classifiers. It plots the TPR against the FPR for various classifiers, for instance using different thresholds. A perfect binary classifier corresponds to a point located at the upper left corner of the ROC space where TPR = 1 and FPR = 0. An example of a ROC curve where each point corresponds to classification using a different threshold is displayed in figure 7.

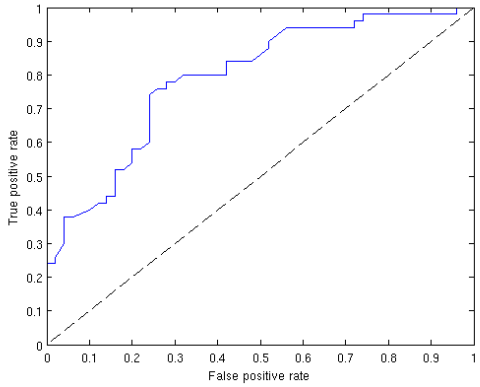


Figure 7: Example of a ROC curve

3 Methods and Approach

In this section we describe in detail our work and the methods we have used as well as approaches that we have taken. The flowchart in figure 8 shows the main steps of the process.

- Firstly, we gathered data on which to test our methods and algorithms, i.e. we created film sequences.
- The frames that the data consist of are synchronized in order to match the IR frames with the visual frames in time.
- The frames are registered so that image pixels in the IR and the visual frames correspond to the same scene pixels.
- Foreground segmentation is performed to segment out objects of interest in both the IR and visual frames.
- Classification of the segmented objects into those that should be masked (objects on the near side of the window) and into those that should not (IR reflections and objects on the far side of the window).

Figure 8: Flowchart illustrating the steps in the process of solving our problem

3.1 Equipment

We have used two Axis network surveillance cameras, one thermographic camera, Axis Q1921, which detects long wave infrared light with wavelengths between 8-14 μm and one regular camera, Axis P1346, capturing light in the visible spectrum. The Axis P1346 will be referred to as the visual camera and the thermographic camera is simply referred to as the IR camera. The IR camera records the relative temperatures in the scene which means that the object with the highest temperature gets maximum intensity and the one with the lowest temperature gets zero intensity.

The IR camera has a micro bolometer image sensor with the resolution of 384×288 pixels. It has a 10 mm lens and a focal length of 12 mm. Its horizontal angle of view is 51° and vertical angle of view is 40°.

The visual camera has a varifocal lens with a focal length between 5.6 mm and 16 mm. We have used the resolution 640×480 pixels when recording. Its horizontal angle of view ranges between $27° - 72°$ and its vertical angle of view between $23° - 53°$.

Both cameras are set to film with their full frame rate capacity of 30 frames per second.

We mounted the two cameras on top of each other as shown in figure 9 and stabilized them using pieces of cardboard and duct tape. We positioned them to get the optical centers as close as possible and adjusted the optics to make the fields of view as similar as possible to ensure that the images obtained from the two cameras display the scene from approximately the same view.

Figure 9: The cameras that were used, the upper white one being the visual camera and the lower black one being the IR camera.

3.2 Datasets

The data is in the form of video sequences. The aim is to have datasets comprising different situations and scenarios that arise in reality, i.e. video sequences resembling what could be caught by a surveillance camera. Two scenes we focused on was the inside of a store and a lobby/front desk. These types of scenes are ones that would be expected to be encountered in surveillance. Furthermore, we wanted the video sequences to comprise scenarios that were particularly problematic. Some scenarios that we tried to include in our datasets were

- Someone in front of the cameras in a regular manner (figure 10a).
- Someone on the far side of a window.
- Someone at a distance so that only the reflection of him is visible in the IR-camera (and possibly reflections in the visual camera as well) (figure 10b).
- Someone in front of the camera such that both him and his reflection is visible in IR (figure 10c) including cases when he is overlapping his own

reflection.

- Someone in front of a window simultaneously as someone is on the other side of the window, including the case when the IR reflection of the person in front of the window overlaps the person behind the window (figure 10d).

- Someone is wearing a jacket or something similar that blocks heat radiation (figure 10e).

- Combinations of the above with several people simultaneously (figure 10f).

In figure 10 are examples of some frames from the collected data in the scene inside of a store. In figure 10b the effect of the IR camera measuring relative temperatures as opposed to absolute temperatures can be seen. A reflection does not radiate as much heat as a person in IR and the camera rescaling the image intensities results in lower contrast.

(a) Person walking, similar visibility in IR and visual

(b) Reflection of a person, only visible in IR

(c) A person and his reflection

(d) A person in front of a window whose reflection overlaps with another person who is standing behind the window (if we combine the images)

(e) Person with a jacket

(f) Combination of scenarios with multiple individuals

Figure 10: Some samples of frames from inside of a store where some different scenarios are caught

3.3 Synchronization of frames

The cameras were set to film with a frame rate of 30 frames per second (fps). However, for reasons such as not perfect network connections the received frame rate varied depending on how many frames the network were able to transmit at that moment. In our datasets, it typically varied between 25 and 30 fps. The frame rates of the cameras did not vary synchronously causing the frames from the visual camera not to match the frames from the IR camera, creating the need for an algorithm to synchronize the film sequences.

Each frame filmed with an Axis camera in the MJPG format has an encoded timestamp. A method of synchronizing the films is making sure that each second there are exactly 30 frames in both sequences by assigning a frame to each 1/30 of a second, namely the one that lies closest in time. The algorithm in detail can be seen in algorithm 2 where s_1 and s_2 denotes the sequences of frames from camera 1 and 2 respectively, t_1 and t_2 are the corresponding encoded timestamps for these frames and d is the duration of the films in seconds.

Algorithm 2 Synchronize frames from two camera sources with frame rates 30 fps

for $i = 0$ **to** $30 \cdot (d-1)$ **do**
  $j_1 \leftarrow \arg\min_j |t_1[j] - \frac{i}{30}|$
  $j_2 \leftarrow \arg\min_j |t_2[j] - \frac{i}{30}|$
  $s_1'[i] \leftarrow s_1[j_1]$
  $s_2'[i] \leftarrow s_2[j_2]$
end for
return s_1' and s_2'

s_1' and s_2' are two new sequences where the frames in s_1 and s_2 have been duplicated and arranged to be at approximately the "right time". For this algorithm to work well, it is assumed that the received frame rates do not vary too much between the cameras or one of the sequences will appear to lag. A solution is to throw away frames from the camera with the higher frame rate. Consequently, when there is a lag in one of the sequences, a lag is created in the other sequence as well. Even if this produces a sequence of frames where nothing is moving there is a closer match in the timestamps between the two sequences which is preferable for our applications. s_1 denotes the sequence with higher frame rate and s_2 the sequence with lower frame rate, t_1 and t_2 are the corresponding encoded timestamps, n is the number of frames in s_2 and algorithm 3 shows the pseudocode of how this procedure looks.

t_1' (s_1) matches t_2 (s_2) better than t_1 (s_1). The frame rates are still variable but now the frame rates of camera 1 and 2 vary synchronously causing frames to be discarded to obtain a better match. A problem is determining which camera has the lower frame rate. A sliding window of width w is introduced in where it is determined which of the sequences has a lower frame rate. An improved version of algorithm 3 is shown in algorithm 4.

Algorithm 3 Throw away frames from the camera that has the higher frame rate

for $i = 0$ **to** n **do**
  $j' \leftarrow \arg\min_j |t_1[j] - t_2[i]|$
  $s'_1[i] \leftarrow s_1[j']$
  $t'_1[i] \leftarrow t_1[j']$
end for
return s'_1 and t'_1

Algorithm 4 Adaptive version of algorithm 3

$k \leftarrow 0$
while $s_1[k] < n_1$ **and** $s_2[k] < n_2$ **do**
  $q \leftarrow$ **false**
  if mean frame rate of $s_1[k$ to $k + w]$ $<$ mean frame rate of $s_2[k$ to $k + w]$ **then**
    $q \leftarrow$ **true**
  end if
  for $i = k$ **to** $k + w$ **do**
    if q **then**
      $j' \leftarrow \arg\min_j |t_2[j] - t_1[i]|$
      $s'[i] \leftarrow s_2[j']$
      $t'[i] \leftarrow t_2[j']$
    else
      $j' \leftarrow \arg\min_j |t_1[j] - t_2[i]|$
      $s'[i] \leftarrow s_1[j']$
      $t'[i] \leftarrow t_1[j']$
    end if
  end for
  $k \leftarrow k + w$
end while
return s' and t'

3.4 Registration of frames

To fuse the information in the frames from the visual and the infrared spectra in a meaningful way image registration is needed to align them. Registration is done as preprocessing for all frame pairs and it is desirable to define a single mapping used for all frames to keep the computational time low.

3.4.1 Parallax Error

The cameras were set up to minimize the effect of the optical centers not being identical. However, there still is projective ambiguity as illustrated in figure 4 meaning any static registration mapping used will only really be valid for some given depth for each pixel. There is also a parallax effect when overlaying images from the two cameras resulting in binocular disparity as described in section 2.2.1. Using the notation from section 2.2.1, let's assume the cameras are positioned so the optical centers differ by the parallax h only in a direction parallel to the image plane and are adjusted to have equal angles of view, i.e. $\alpha = \beta$. A simplified formula for the parallax error in percentage at depth d becomes

$$x_2' - x_1' = \frac{h}{d \tan \alpha}.$$

In this case, the error depends only on the depth of the scene point and not on the height a. For our camera setup the parallax h, i.e. the vertical distance between the two lens centers, was 6 cm and the vertical angle of view $2 \cdot \beta$ for the IR-camera was 40°. A plot of the binocular disparity in percentage for these values at scene depths of 1-20 m is shown in figure 11, e.g. for depths greater than ∼8 m the error is below 2% corresponding to ∼10 pixels in the visual frames.

If the fields of view are not adjusted to be equal the binocular disparity depends not only on the depth, d, but also on the height, a. In figure 12 is a plot of the absolute value of the binocular disparity in percentage when the angles of view are miscalibrated by 4° for different values of d and a.

To correct for the binocular disparity we first tried registering the frames using a homography estimated from manually selected landmark points visible in both frames. A general homography describes a mapping taking a plane to another plane [9], meaning that if only parallax error was present it could be corrected for along a plane in the scene. The plane is determined by the landmark points.

Using a homography for registering frames the alignment was reasonable near the center of the images but poor for most other pixels. A reason being that in addition to the parallax error there is also radial distortion from the lenses present in images from both cameras. Radial distortion increases further away from the image center and causes large error near the edges.

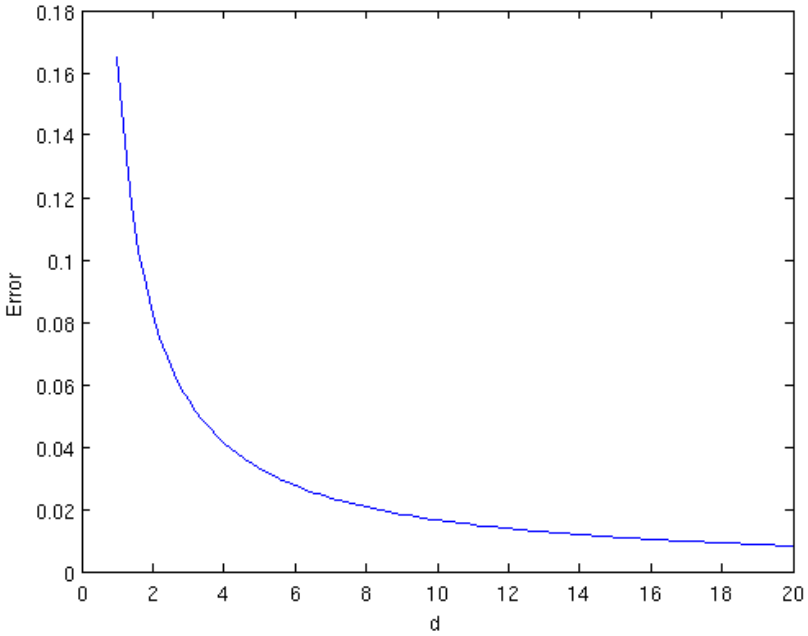


Figure 11: The theoretical binocular disparity when the two cameras have the same angle of view. Notice that the binocular disparity in this case only depends on the depth, d, of the scene point. The further away the scene point is from the camera, the larger the error.

3.4.2 Thin Plate Spline Mapping

To correct for the radial distortion as well, Thin Plate Spline (TPS) warping was used, as described in section 2.2.2. The mapping is done by manually selecting landmark points used to calculate the TPS mapping function.

An example of a TPS mapping computed from 12 landmark points can be seen in figures 13 and 14 where one IR frame is registered onto the corresponding visual frame. In the superimposed frames the effect of the mapping is most visible on the lamps in the ceiling and on the person to the far right in the frames. To illustrate the distortion between the corresponding frames the same mapping is done in figure 15 with an artificial grid instead of the actual frames. In the mapped grid a pincushion distortion effect is quite clear which indicates that the radial distortions differ for the two cameras.

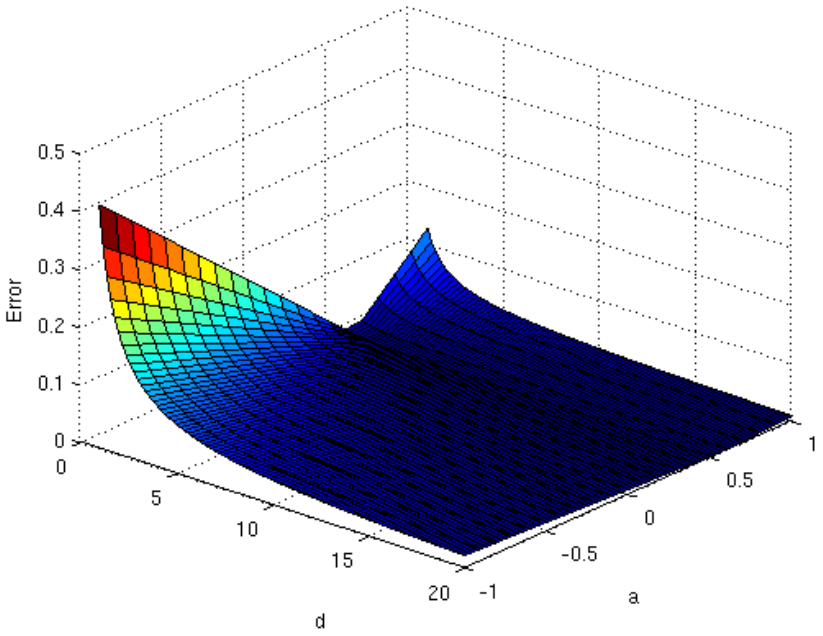
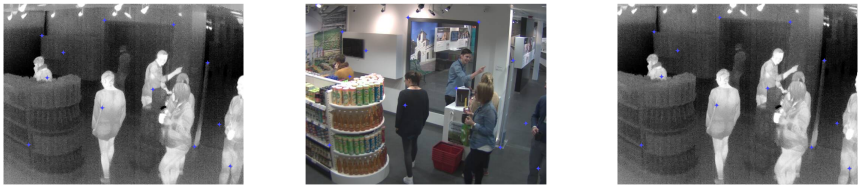


Figure 12: The theoretical b1inocular disparity when the cameras have different angles of view, namely, $\alpha = 20°$ and $\beta = 22°$. In this case the binocular disparity does not only depend on the depth, d, of the scene point (as in figure 11), but also on the height, a.

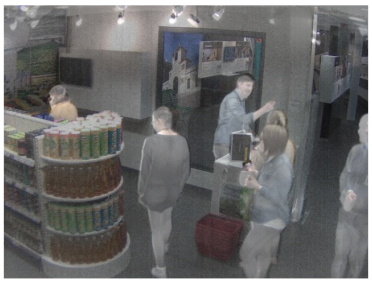


(a) IR frame with landmark points

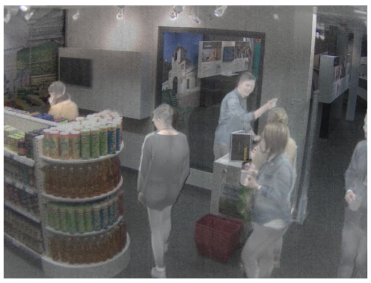
(b) Visual frame with landmark points

(c) IR-frame after mapping

Figure 13: Example of a TPS mapping on one pair of frames.

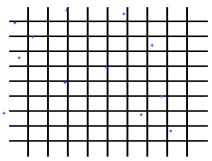


(a) Superimposed frames before mapping

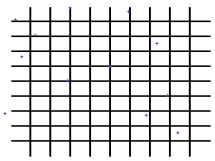


(b) Superimposed frames after mapping

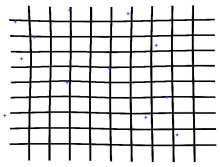
Figure 14: Example of TPS mapping performed on a frame pair.



(a) Grid corresponding to IR-frame with landmark points



(b) Grid corresponding to regular frame with landmark points



(c) Grid after mapping

Figure 15: Example of a TPS mapping on two artificial grid frames. Notice the deformation on the grid in the right image that results from the TPS mapping.

3.5 Foreground Segmentation

When the frames are synced and registered the next step is to find objects of interest using foreground segmentation. This section gives an overview of some of the techniques used for segmentation of the frames.

The segmentations consist of binary matrices of the pixels with ones to represent detected foreground. To deal with noise a morphological opening is performed on the binary image to remove isolated or small regions. A morphological closing is used to connect components that are spatially close and to smooth the edges of the segmentation.

3.5.1 Thresholding

For our datasets and most typical scenarios the temperature of the objects of interest (such as humans) is higher than the surrounding ambient temperature. As a result in the IR images these objects will in general have higher pixel intensities than the background. This motivates *thresholding* as a simple segmentation method for the IR images where all pixels with intensity higher than some threshold are classified as foreground.

The intensities in the IR images represent the relative temperatures in the scene meaning there is not a fixed threshold or interval that will segment out objects of interest. Instead the threshold, θ, is computed for each frame by weighting the mean intensity of the IR image, I_{mean}, and the maximum intensity of the IR image, I_{max}, by the weight γ:

$$\theta = \gamma \cdot I_{mean} + (1 - \gamma) \cdot I_{max}.$$

Furthermore, an *extended thresholding* is done for each resulting connected component in the segmentation. A bounding rectangle with some padding is found and a second, local thresholding with a lower threshold value is performed within this rectangle to capture more of the object. An example of thresholding and extended thresholding of an IR frame is seen in figure 16. The bounding rectangles with the padding are shown in gray. Note that since thresholding only uses pixel intensities the relatively warm lamps in the ceiling are segmented as foreground even though they are part of the background.

3.5.2 Background Subtraction

To make use of the temporal information in the videos, the more dynamic *Eigenbackground subtraction* and *Mixture of Gaussians* (MoG) methods are used. They are described in section 2.3. The methods estimate a background image and then determines which pixels deviate enough from the background to be

(a) IR frame

(b) Segmentation using simple thresholding with $\gamma = 0.6$

(c) Segmentation after extended thresholding

Figure 16: An IR frame and the segmentation using thresholding and extended thresholding. The gray rectangles in the rightmost image indicate the area where further thresholding is done.

classified as foreground. This for example prevents the stationary warm lamps being segmented as foreground and removes the need to assume the foreground has higher intensity than the background. In figure 17 the same IR frame and the segmentations using the dynamic methods are shown. The IR image is downsampled by a factor of 4 in both dimensions in the Eigenbackgrounds method to avoid working with excessively large matrices when computing the eigenvectors, explaining the grainier look of the Eigenbackgrounds segmentation. Note that in the MoG segmentation the reflection is segmented as a foreground object. The remaining small noise objects are mainly due to intensity rescaling when multiple people entered the scene and a slow learning rate. The methods are discussed further in section 4.1.

(a) IR frame

(b) Segmentation using Eigenbackgrounds

(c) Segmentation using Mixture of Gaussians

Figure 17: An IR frame and the segmentation using dynamic segmentation methods.

3.6 Object Matching

Having obtained IR and visual frame correspondence with segmented objects of interest in each frame the next step is to determine which of these objects to mask and unmask when there is a privacy mask over a window. Objects can be divided into three principal classes:

- Class A: Heat radiating objects (such as humans) on the near side of the

window

- Class B: IR reflections of heat radiating objects
- Class C: Objects on the far side of the window

Examples of the three classes can be seen in figure 1. Objects of class A should be unmasked while those of class B and C should be masked. The three classes have some different properties that should make it possible to distinguish them as can be seen in table 1. It should be possible in most cases to determine if the object is to be unmasked or not by looking at in which segmentation they appear. In principal, if a particular object exists in both the IR and the visual segmentations it should be unmasked.

Foreground objects segmented as a result of noise or changes in the background and objects of ambient temperature do not fit into any of the three classes. However, such objects will most likely be segmented in only one of the segmentations. A possible situation is when a reflection is prominent enough in the visual frame to be segmented in both segmentations resulting in the reflection area being unmasked. Although, it is not sure this is unwanted since the reflection might still block the view through the window.

Frame type	Class A	Class B	Class C
Visual	Detected	Not detected (mostly[2])	Detected
IR	Detected	Detected	Not detected

Table 1: Properties in IR and visual for different classes

3.6.1 Segmentation Matching

The main idea is to match objects in the segmentation of the IR frame with objects in the segmentation of the visual frame, and if a match is found for an object then it should be put in class A and thus be unmasked. If we would have perfect segmentations and perfect registration of the IR and visual frames, then it would be straightforward to for every object in the IR segmentation look at the corresponding pixels in the visual segmentation and determine that it is a match if the object also exists there. However, the registration performed is typically not perfect and the error may be as large as 5 pixels in some areas of the image. Even more problematic is the segmentations being different making them harder to match.

[2]Of course there are usually some reflections that are more or less visible (depending on the light conditions on the scene) in the visual frames as well but normally they are not as dominant as the IR reflection and will as such not be segmented to as large an extent as the corresponding IR reflections.

Since objects from different classes may overlap it is not as straightforward as matching pixels in the corresponding segmentations. There is a need of a more sophisticated method to determine if two objects indeed are the same and not just objects from different classes overlapping. An idea is to create a score to measure the correspondence between an object in the IR segmentation and the corresponding area in the visual segmentation. For each object in the IR segmentation, each pixel is gone through and it is determined if there is a corresponding foreground pixel in the visual segmentation by looking in some neighborhood. The number of pixel matches are added and divided by the total number of pixels to form a score, s_1.

3.6.2 Gradient Matching

To increase the robustness another score, s_2, is used which matches the gradients of the original corresponding images as opposed to the segmentations of them. The procedure to compute s_2 is,

1. Compute gradients in both images.

2. Threshold the gradient in the IR frame to get a search area.

3. Iterate through the pixels in the search area.

4. Give a match score of how close the gradient directions are in magnitude, look in a neighborhood.

5. Sum all the match scores, normalize and threshold to determine if the object is a match.

For heat radiating objects the gradients are sharper and larger in magnitude in the IR image compared to the gradients in the visual image, making it somewhat difficult to match the magnitudes so instead something that is less dependent on the sharpness of the transitions is matched, namely the directions of the gradients. By matching the gradients, it is not enough that two objects overlap at a pixel, but their edges should also be directed the same way.

Since there is a need to take into account the registration errors that might exist the matching is done in a neighborhood. So the gradient direction difference that is chosen as a match score for a particular pixel is the smallest difference of the gradient direction in the IR pixel and the gradient directions in the pixels in the corresponding neighborhood in the visual image.

3.6.3 Object matching algorithm

An algorithm to determine if objects are of class A by weighting s_1 and s_2 can be seen in algorithm 5.

Algorithm 5 Algorithm for determining if objects in an image should be masked or not. O_{IR} and O_v are the segmentation images of the IR and visual image respectively. k is the number of objects in the IR segmentation (connected components in O_{IR}). n is the neighborhood size. s_1 is the score when searching through the segmentation pixels. s_2 is the score when searching through the gradients.α is the weight given to the scores. T is the threshold used to determine a match.

M is an empty mask

Compute gradient directions G_{IR} and G_v

for $i = 1$ **to** k **do**
  $o \leftarrow O_{IR}[i]$ {Current object}
  $m \leftarrow 0$ {Number of matches}
  for $j = 1$ **to** NumberOfPixels(o) **do**
    Set N as a $n \times n$ neighborhood centered at j
    if $\exists\, l \in N$ s.t $O_v[l] = 1$ **then**
      $m++$
    end if
  end for

  $s_1 = 1 - \frac{m}{\text{NumberOfPixels}(o)}$
  $s_2 \leftarrow 0$

  Search area S is equal to pixels where $G_{IR} > 0.1$ in o

  for $j = 1$ **to** NumberOfPixels(S) **do**
    Set N as a $n \times n$ neighborhood centered at j
    $s_2 = s_2 + \min_{l \in N} |G_{IR}[j] - G_v[l]|$
  end for

  if $\alpha s_1 + (1 - \alpha) s_2 < T$ **then**
    Add o_c to M
  end if

end for
return M

3.7 Classifying Reflections in IR

In section 3.6 an algorithm is presented to determine if an object in the segmentations should be masked or not by matching information from the visual and IR frames. In this section we investigate some measures of the information in the IR images alone to help determine whether an object is of class A or B

(objects of class C are not visible in the IR frames).

From physics we expect the image of a reflection to be more diffuse and lower in intensity than the image of the object itself as seen in figures 18 and 10. One thing to note again is the pixel intensities represent the relative temperature content in the scene meaning the intensity of a reflection is increased when there is no object of class A in the scene as can be seen by comparing figures 10b and 10d, however, it can also be noted that the entire intensity image is rescaled.

Figure 18: IR images of a human (left) and a reflection of a human (right)

Let I be the IR intensity image and o the current object pixels, i.e. $I(o)$ are the IR intensities at the current object pixels. M is the mean of I and $I_M = \frac{I}{M}$ is the "mean normalized" intensity image. Let G be the image gradients and $|G|$ the magnitude of the gradients.

To get a measure of the pixel intensities consider $I_M(o)$, the mean normalized image at the current object. Threshold $I_M(o)$ to get the normalized intensities that are greater than 1, (i.e. the mean intensity of the entire image), and take the mean of this, if there are no intensities greater than 1 set the measure to be 1. The first measure, m_1, is defined as

$$m_1 = \max(\text{mean}(I_M(o) > 1), 1).$$

To get a measure to distinguish between the diffuseness of a reflection and the higher contrast in the image of a human a second measure is defined, m_2, as

the standard deviation of the pixel intensities at the current object,

$$m_2 = \text{std}[I(o)].$$

The gradients of the contour of a class A-object are expected to be larger in magnitude than those of class B-object. To include the contours of the current object, a morphological dilation of the binary image o is performed to get a slightly larger object o'. Consider the gradient magnitudes in the dilation of the object, $|G|(o')$. The dominating edges are extracted by selecting the gradients that are more than two times the standard deviation of $|G|(o')$ in magnitude and their mean are defined as a third measure, m_3,

$$m_3 = \text{mean}(|G|(o') > 2 \cdot \text{std}[|G|(o')]).$$

3.8 Object separation

A particular situation which is an issue for the methods described in sections 3.6 and 3.7 is when objects of classes A and B in the IR segmentation are concatenated as the same object. Such a situation can be seen in figure 19. If the methods decide that the object should be unmasked this leads to areas outside of a class A-object being unmasked, meaning the window can be seen through when it should not.

Figure 19: The left image shows when a person is concatenated with his reflection and the right is the corresponding IR segmentation

An idea is to use the k-means algorithm presented in section 2.5 to divide the concatenated object since the intensities of the class B-pixels are usually lower

than those of class A. The k-means algorithm should be able to divide them if the pixel intensities are used as observations and the number of classes is set to two.

When an object is not concatenated, i.e. only class A or B, the algorithm still tries to separate the object into two classes even though there should be only one. The result depends on how homogenous the pixels of the object are but most likely both classes will get pixels assigned to them as opposed to the ideal result where all the pixels are assigned to the same cluster. Examples can be seen in the results in section 4.3. Needed is a way to determine if an object is a concatenation or not to decide if the object should be separated.

One way would be by inspecting the histogram of the object pixels, specifically looking at the number of peaks. Usually there is one peak when only one of the classes is present and two peaks when it is a concatenation. Another way is to run the k-means algorithm and analyze the resulting clustering. If there are mainly two connected components, then most likely the separation is satisfying and should be used to divide the object. If there on the other hand are many fragmented connected components, then most likely the object consists of only one class and should not be separated.

4 Results and Evaluation

4.1 Segmentation Algorithms

In this section we discuss the segmentation methods that were used, how they perform and their benefits and drawbacks. The methods considered were (extended) thresholding, Eigenbackgrounds and Mixture of Gaussians (MoG).

In figure 20 the segmentations from the methods on three different images are shown, and in table 2 the corresponding true positive and true negative rates are displayed. Figure 21 shows the ROC's for the methods on the different images.

4.1.1 Thresholding

In the second column of figure 20 are the results of using the thresholding method (explained in section 3.5). It has the advantage of being fast, but on the downside it is very crude. As can be seen in the figure, there are a lot of unwanted areas that get segmented as a result of the heat produced by the lamps in the ceiling. The method seems to segment the humans fairly well (86.2% true positive rate) except in the second image where all the methods have trouble finding the entire bodies. As opposed to the other methods it has no learning time and may be a good algorithm to use during start up of the system. For a

more robust system however, more dynamic methods are required.

4.1.2 Eigenbackgrounds

The Eigenbackgrounds method in the form described in section 2.3.2 collects samples of the background during initialization to compute the eigenspace and to avoid working with huge matrices the images are downsampled. The eigenspace is still computationally expensive to recompute meaning the method relies on having good samples of the background during the learning time. (An adaptive version of the method is presented in [6] where the eigenspace is updated continuously without recomputing the entire eigenspace.) The strength of the Eigenbackgrounds approach on our data lies in that it projects the IR image onto the eigenspace which efficiently deals with the image intensities rescaling when the scene content changes rapidly, e.g. when multiple persons enter the frame as can be seen in the second image of figure 20d compared with MoG. The Eigenbackgrounds method with a 99.6% true negative rate which is the highest of the four methods. As can be seen in the ROC in figure 21 this is consistent through the three images. On the other hand, it has the lowest true positive rate of 85.0% although the second image is mainly responsible for that.

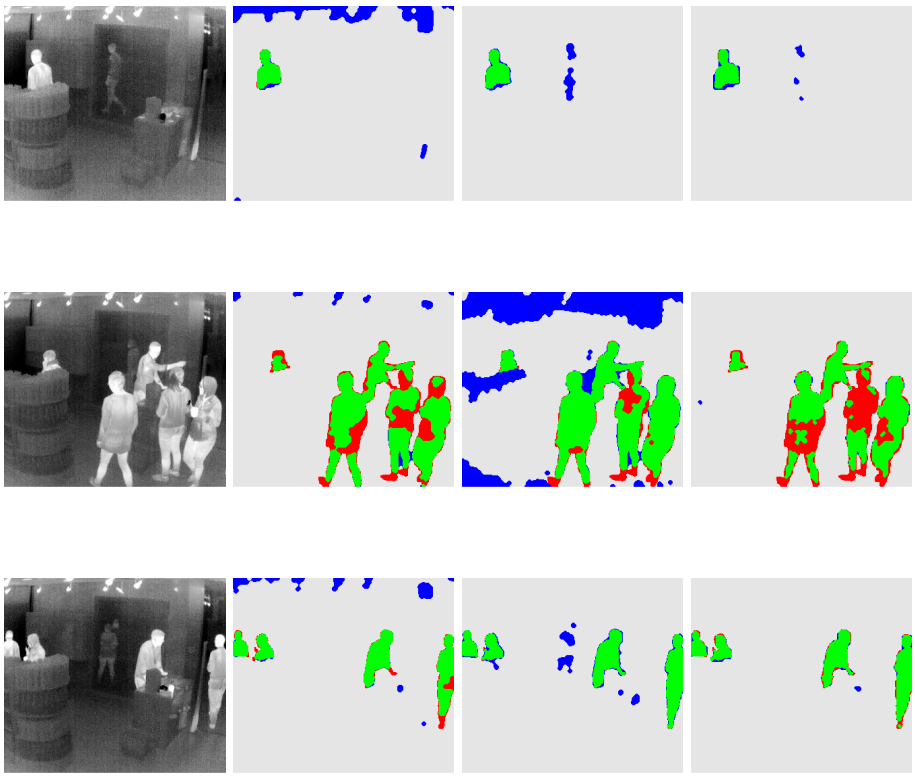
4.1.3 Mixture of Gaussians

The Mixture of Gaussians approach (as described in section 2.3.1) also has a learning time during which it is favorable to have frames clear of foreground objects. However, the method is adaptive and will eventually learn the static background. How quickly it does so depends on the learning rate parameter affecting how fast it deals with the IR intensities rescaling. In the second image of figure 20c we see when it is dealing with such a rescaling. This frame is responsible for the relatively low true negative rate of 92.6% which is also apparent in figure 21. It has a true positive rate of 94.9% which is the second highest.

Of the methods we consider the MoG method to be the most suitable with a high true positive rate and true negative rate (except when the rescaling phenomenon occurs). It is also the method which segments the most reflections which is useful when evaluating the object matching and measures.

Method	True positive rate	True negative rate
Thresholding	86.2%	96.8%
Extended thresholding	96.8%	92.8%
Mixture of Gaussians	94.9%	92.6%
Eigenbackgrounds	85.0%	99.6%

Table 2: True positive and true negative rate for the different segmentation methods



(a) IR image (b) Thresholding (c) MoG (d) Eigenbackgrounds

Figure 20: The results of the three different segmentation methods where we have marked the true segmentation manually. Green represents correct segmentation (true positives), blue represents false positives (areas that have been classified as foreground but are part of the background), red represent false negatives (areas that have been classified as background but are part of the foreground) and light gray is anything that is part of the background and classified as such (true negatives).

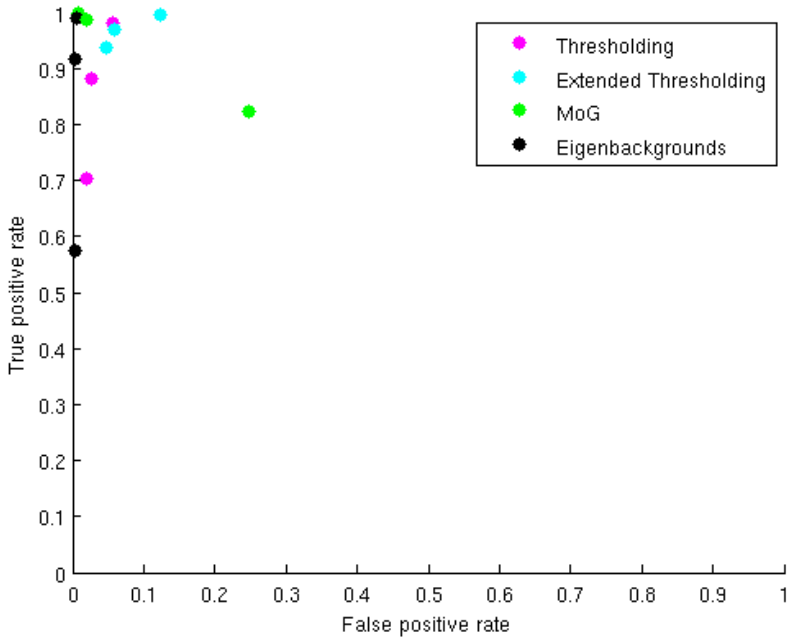


Figure 21: Receiver operator characteristic for the different segmentation methods and the 3 different images in 20

Measure\Class	A	B	A & B concatenated
s_1	276 (98.2%)	2 (1.3%)	78 (87.6%)
s_2	263 (93.6%)	10 (6.5%)	83 (93.3%)
m_1	252 (89.7%)	16 (10.5%)	58 (65.2%)
m_2	262 (93.2%)	12 (7.8%)	51 (57.3%)
m_3	236 (84.0%)	17 (11.1%)	62 (70.0%)

Table 3: The number of objects from each class that are classified as A for each measure.

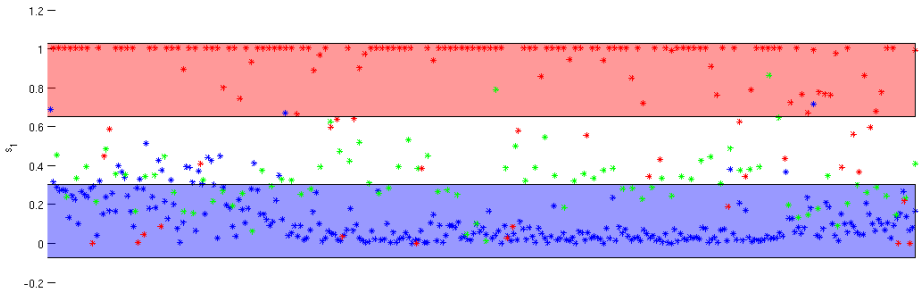
4.2 Object Matching and Classification

The object matching algorithm presented in section 3.6 makes use of two matching scores that are combined and thresholded to determine if an object is a match. These can be seen as measures just like the ones presented in section 3.7 and can be analyzed in the same way. We have gone through a number of frames from the video sequences and manually marked the true classes of the objects in the segmentation. The measures are computed and the correlation between a particular measure and a class is investigated. The measures are illustrated by plotting the measure values for the manually marked objects and the points are colored according to what class they belong. Such plots for the two scores, s_1 and s_2, and the three measures m_1, m_2 and m_3 can be seen in figures 22 and 23 respectively. The two colored bars in the plots have a width of 3 times the standard deviation of the observations in the corresponding class centered at their mean. Heuristically, this is approximately the interval within the observations of the same class stay.

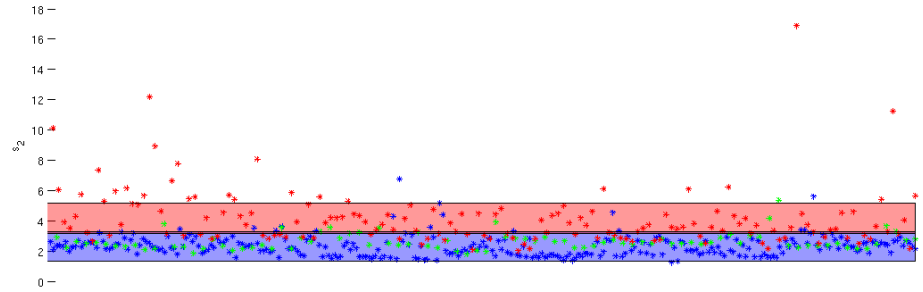
4.2.1 Thresholding the measures

Thresholds to binary classify the observations as A (positive) or B (negative) are obtained by looking at the ROC curve for each score/measure. On each point of the curve a particular threshold is used to classify the observations. Ideally a point on the ROC curve should lie in the upper left corner where true positive and true negative rates both are equal to one, thus the threshold is chosen as the one corresponding to the point closest to the upper left corner. The ROC curves for s_1, s_2, m_1, m_2 and m_3 are displayed in figure 24.

In table 3 one can see how the obtained thresholds classifies observations for the different measures. Classifying based on s_1 provides best results correctly classifying 98.2% of class A and 98.7% of class B. Among the measures m_2 performs best correctly classifying 93.2% of class A and 92.2% of class B.

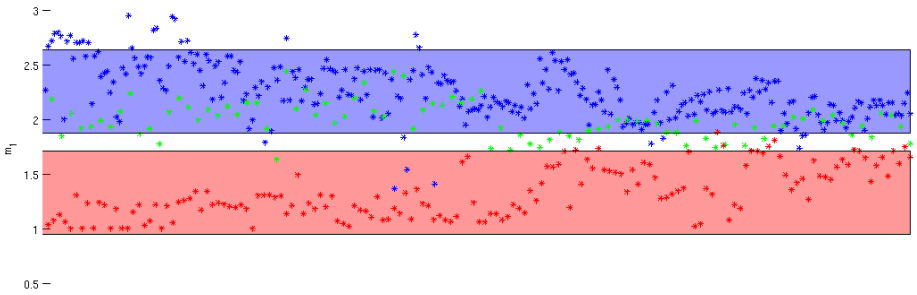


(a) Score 1

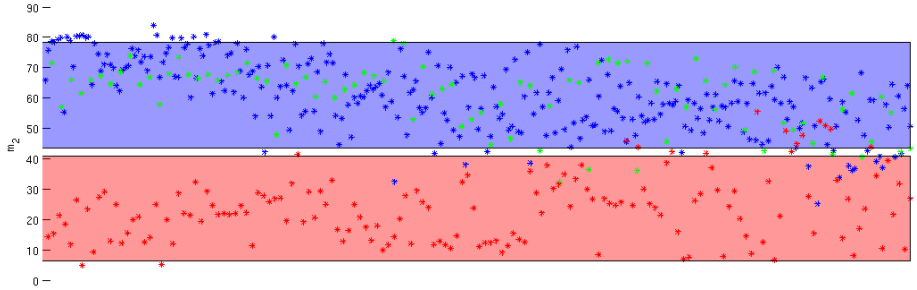


(b) Score 2

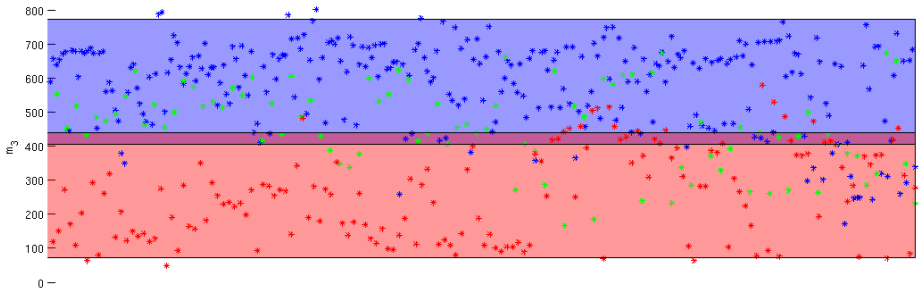
Figure 22: Matching scores (as defined in section 3.6) for different classified observations where blue points represent class A, red points represent class B and green points represent objects where the two classes are concatenated.



(a) Measure 1

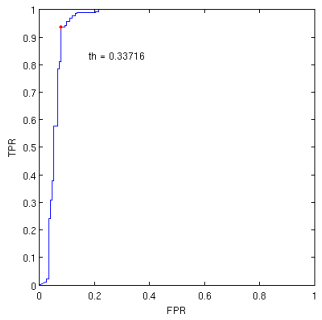


(b) Measure 2

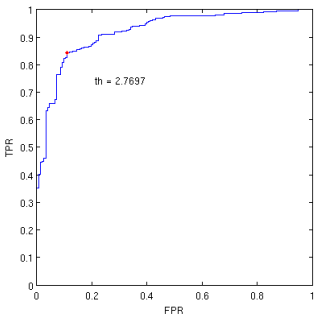


(c) Measure 3

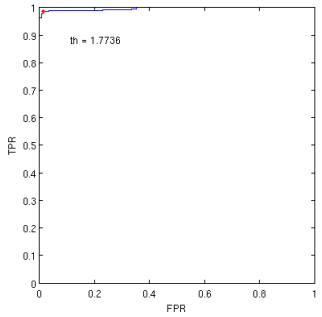
Figure 23: Measures values (as defined in section 3.7) for different classified observations where blue points represent class A, red points represent class B and green points represent objects where the two classes are concatenated.



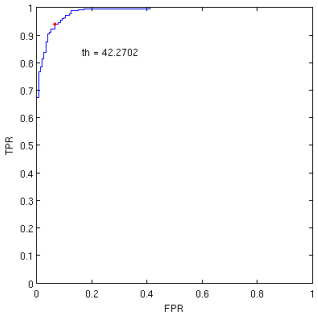
(a) Score 1



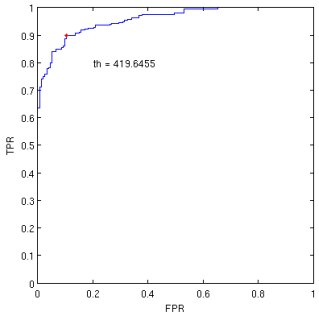
(b) Score 2



(c) Measure 1



(d) Measure 2



(e) Measure 3

Figure 24: Receiver operator characteristic curves for the two scores and the three measures. The red dot is the point on the curve closest to the upper left corner and the corresponding threshold is written out.

Number of measures\Class	A	B	A & B concatenated
0	0 (0%)	103 (67.2%)	0 (0.0%)
1	3 (1.1%)	44 (28.8%)	5 (5.6%)
2	4 (1.4%)	5 (3.3%)	10 (11.2%)
3	18 (6.4%)	1 (0.7%)	17 (19.1%)
4	56 (19.9%)	0 (0.0%)	29 (32.6%)
5	200 (71.2%)	0 (0.0%)	28 (31.5%)
Total	281	153	89

Table 4: The number of objects from each class that are classified as A for a specific number of measures.

n\Class	A	B	A & B concatenated
1	281 (100%)	50 (32.7%)	89 (100%)
2	278 (98.9%)	6 (3.9%)	84 (94.4%)
3	274 (97.5%)	1 (0.7%)	74 (83.2%)
4	256 (91.1%)	0 (0.0%)	57 (64.0%)
5	200 (71.2%)	0 (0.0%)	28 (31.5%)

Table 5: The number of objects from each class that are classified as A for at least a specific number of measures.

4.2.2 Combining the measures

Table 4 shows the distribution of the number of measures for which the observations are classified as A. For instance, there are 18 class A observations that are correctly classified in exactly three measures. A way of combining all the measures (the two scores and three measures) is by requiring that an observation has to be classified as A in at least n number of measures for it to be definitively classified as A. A result of this for different n can be seen in table 5 and the corresponding ROC curve in figure 25. The point on the ROC curve closest to the upper left corner corresponds to $n = 3$. For $n = 3$ we get that 97.5% of class A observations are classified correctly (true positive rate) and that 99.3% of the class B observations are classified correctly (true negative rate). The true positive rate is slightly lower than when using only s_1 (-0.7 difference in percentage points) but the true negative rate is higher (+0.6 difference in percentage points). It is also a matter of what to prioritize, one can easily obtain a 100% true positive rate by choosing $n = 1$ but this at the cost of increasing the false negative rate to 32.7%, which probably would be considered excessive. More reasonable could be the choice of $n = 2$ in which case the true positive rate is as high as 98.9% and the false negative rate increases to 3.9%. In different applications and scenarios one prioritizes the true positive and the true negative rate differently so n can be chosen accordingly, but the more reasonable choices of n seem to be 2,3 or 4.

Another way of combining the measures is by taking a weighted sum of them to obtain a total score for each observation. The thresholds θ_i obtained from the ROC curves are used as weights to "normalize" the measures. The weighted

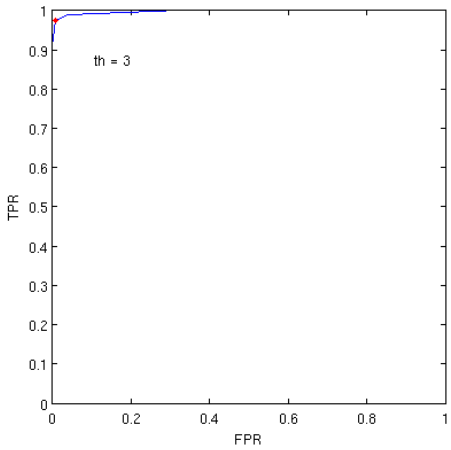


Figure 25: ROC curve for the data in table 5. The threshold here represents the least number of measures for which an object has to be classified as A for it to be positively classified.

measures m_1, m_2 and m_3 are added and the weighted s_1 and s_2 are subtracted since they have a lower score for class A objects to get the total score m as

$$m = \frac{m_1}{\theta_{m_1}} + \frac{m_3}{\theta_{m_3}} + \frac{m_3}{\theta_{m_3}} - \frac{s_1}{\theta_{s_1}} - \frac{s_2}{\theta_{s_2}}.$$

The values of the total weighted score m are shown in figure 26 and the corresponding ROC curve in figure 27. Combining the measures in this way it is possible to increase the true positive rate to 98.9% with the same true negative rate, 99.3%, as when having $n = 3$ as threshold when counting the number of fulfilled measures.

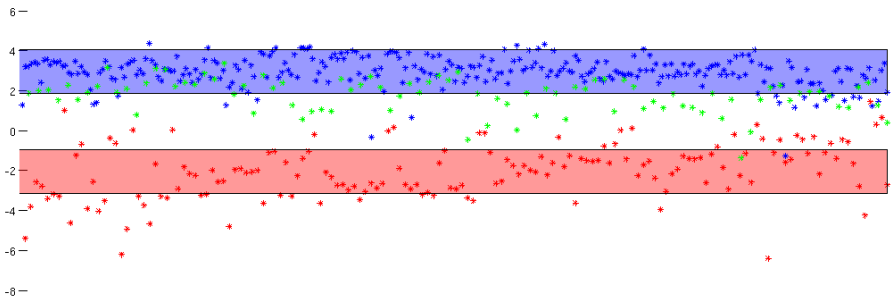


Figure 26: Total weighted score m for the different classified observations where blue points represent class A, red points represent class B and green points represent objects where the two classes are concatenated.

So far we have not discussed the third class, i.e. when the segmented object is a concatenation of class A and B. It is unclear whether it is desirable to classify

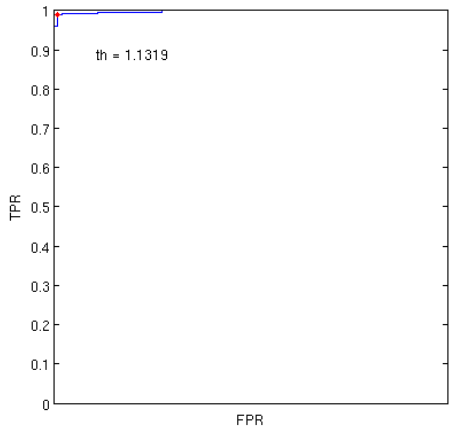
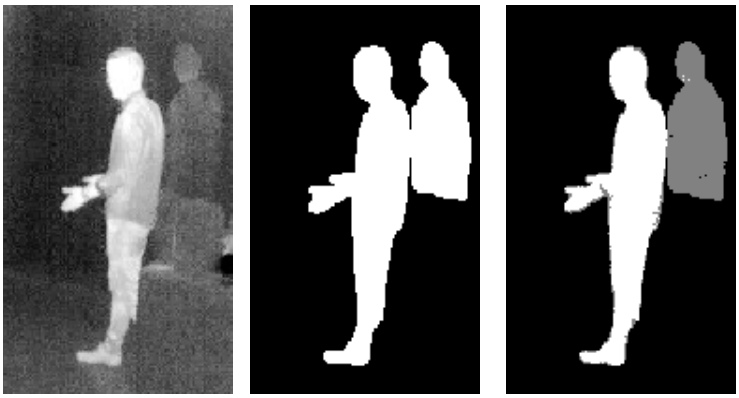


Figure 27: ROC curve for the total weighted score m. The red dot is the point on the curve that is closest to the upper left corner and the corresponding threshold is written out.

these as A or B and this might depend on the application. Commonly, the class A part of the object is more dominant and the object will most likely be labeled as such. This can be observed in table 5 where more than 50% gets labeled A for all n except the most restrictive $n = 5$. The issue of concatenated objects is further discussed in section 3.8.

4.3 Object separation

As can be seen in figure 28c k-means clustering works quite well to separate a person from a reflection. With the use of some morphological operations it is possible to improve the clustering further.



(a) IR image (b) Segmentation (c) K-means clustering

Figure 28: The result of k-means clustering when a person is concatenated with his reflection

As suspected and discussed in section 3.8, the results are not as satisfactory when clustering is used on objects that should not be separated. This can be seen in figure 29c where the k-means algorithm has been performed on a reflection and on a human separately. Preferably such clustering should be avoided and the original segmentation used.

(a) IR image (b) Segmentation (c) K-means clustering

Figure 29: The result of k-means clustering when done on a the pixels of a person (left segmentation) and the pixels of a reflection (right segmentation) separately

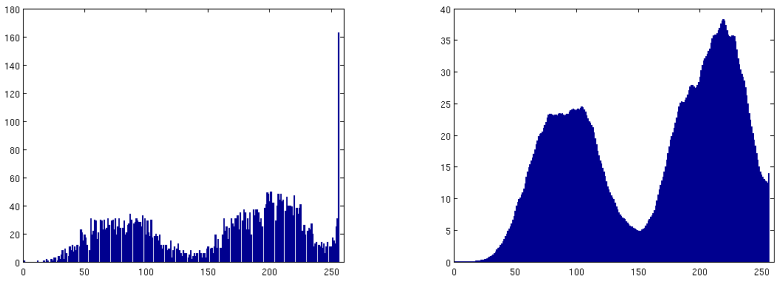
For determining when clustering should be used we investigated the approach of counting peaks in the histogram of an object. In figure 30a is the histogram of a concatenation. It is pretty clear for a human observer that there are two peaks but to make it easier to identify the number of peaks automatically the histograms are smoothed with an averaging filter, shown in figure 30b. In figure 31 are the corresponding histograms of a human and a reflection. In both

of these cases, there is only one peak present. Commonly, there are two distinct peaks when a class A and a class B object are concatenated and only one peak when it is only one class. Therefore, it should be possible to use some kind of peak counting algorithm to determine if the k-means algorithm should be used.

It should be mentioned that there are various situations in which more than one peak will be present even though there is only one of the classes, for instance when a person is wearing a thick jacket causing the heat radiation of the object to be unevenly distributed. Another case would be when the segmentation covers an area larger than the actual object in which case there might be multiple peaks originating from the surrounding area. This is not necessarily a problem since clustering then ideally divides the area into object and "surroundings" as in figure 32. The algorithms will then hopefully not classify the surroundings as A.

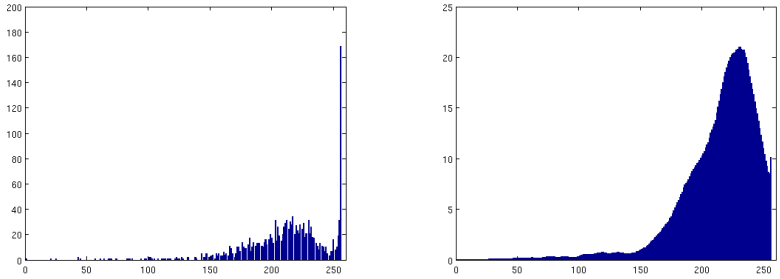
If it is assumed that class A has higher intensity mean, another idea is to keep the cluster with higher mean as a new segmentation and disregard the other cluster which would be assumed to be of class B or the "surroundings", i.e. objects that should be masked regardless.

Table 6 shows the distribution of the number of peaks (found using the algorithm from [10]) among the different classes where classes has been marked manually. It shows that on our data for class B there is never more than one peak found, and when there is a concatenation of both classes there is always at least one peak but mostly (79.4%) more than one. Since there in general are more intensity variations in class A objects there are more peaks for those but they tend to have fewer peaks than concatenations (70.9% with one peak for class A as opposed to 20.6% with one peak for concatenations). For the class A objects with more than one peak the extra peaks are usually caused by the segmentations including surroundings.



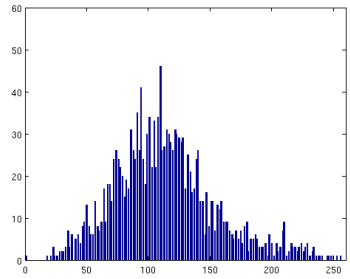
(a) Original histogram (b) Smoothed histogram

Figure 30: Histogram for the pixels of the concatenated object in figure 28a

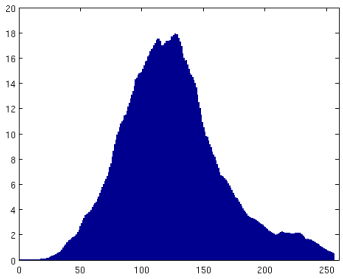


(a) Original histogram

(b) Smoothed histograms

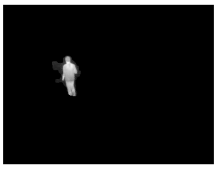


(c) Original histogram

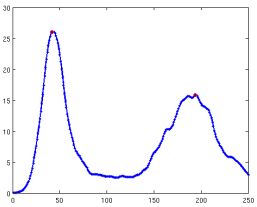


(d) Smoothed histogram

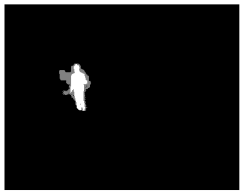
Figure 31: Histogram for the pixels of the person in figure 29a (upper plots) and his reflection (lower plots)



(a) Poorly segmented human



(b) Smoothed histogram



(c) K-means clustering

Figure 32: Poor segmentation leads to two peaks in the histogram. After k-means clustering we can throw away the gray area and keep the white area as the segmentation.

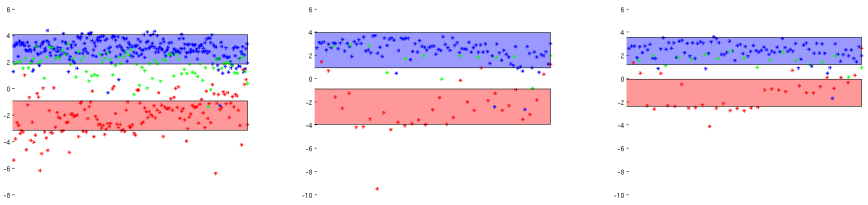
Number of peaks\Class	A	B	A & B concatenated
0	12 (4.2%)	106 (71.6%)	0 (0.0%)
1	205 (70.9%)	42 (23.4%)	22 (20.6%)
2	67 (23.2%)	0 (0%)	82 (76.6%)
3	5 (1.7%)	0 (0%)	3 (2.8%)
Total	289	148	107

Table 6: The number of objects for which a specific number of peaks was found for each class

4.4 Dependence on Resolution

One of the main issues with using thermographic cameras in surveillance is the cost of the cameras. While the prices of these cameras are expected to drop in the near future, the cost is still of concern at the moment. Though the cameras with higher resolution are quite expensive, it is possible to acquire ones with lower resolution for a significantly lower price. Therefore, it is of interest to investigate how well our methods fair when lower resolutions are used and if the expected performance drop is significant enough to motivate purchase of expensive cameras with higher resolutions.

Of course, the quality of the segmentation is affected when having lower resolution but even for resolutions down to 36×48 we achieved reasonable segmentation of humans. This segmentation can then be used as initial guess for segmentation in the visual frame with higher resolution so the segmentation is not the main concern. We have done similar analysis as in section 4.2 but with the IR images downsampled to measure how much the relevant information is degraded with lower resolution. When looking at the different measures the values for s_2, the gradient direction matching, seemed to degrade the most with lower resolutions. The other measures held up fairly well still indicating a mean difference between the classes, however, the gap between class A and B decreased as the resolution got lower making classification less accurate. This is illustrated in figure 33 with the values of m, the total weighted measure, calculated as in section 4.2 for the original IR resolution 288×384 and two downsampled versions with resolutions 144×192 and 36×48. (Not as much ground truth was labeled for the downsampled versions because of difficulties in tuning the segmentation parameters for different resolutions and data sets.) It is interesting to see that for a resolution as low as 36×48 the measures seem to be useful.



(a) Original resolution, 288×384

(b) Lower resolution, 144×192

(c) Lower resolution, 36×48

Figure 33: Total weighted score m for the different classified observations when using original and downsampled IR frames. Blue points represent class A, red points represent class B and green points represent objects where the two classes are concatenated.

5 Conclusions

The use of a thermographic camera improves the detection of humans and other heat radiating objects since they (in room temperature) have larger contrast to the background than in a visual camera. A drawback is that objects of room temperature are much harder to detect, but assuming living creatures are the objects of interest it is a great improvement. A second drawback is the possibility of reflections of warm objects being segmented which then can be taken care of with the methods presented in this thesis. The application of privacy masking would not be possible with the visual camera alone, assuming it is not a 3D camera that gives the possibility to compute the depth of scene points.

The registration of the images using thin plate splines was the most computationally expensive operation, especially when interpolating the holes resulting from the transformation. It is possible to use a homography instead which is much faster, but it did not perform as well as the thin plate spline mapping for the camera setup used in this thesis. The use of registration is almost inevitable since the object matching algorithm is dependent on the objects to be matched being registered. It is possible not to use the object matching algorithm and only use the reflection classifier, which only uses the IR images, also reducing the computational complexity. However, omitting the registration of the images would cause the unmasking to be slightly shifted meaning that some parts of the window would be visible and seen through. The interpolation part of the TPS mapping can be omitted and morphological dilation can be used to fill the holes in the segmentation but it would have a negative impact on the scores from the object matching algorithm, although not a crucial one.

Of the segmentation methods we investigated, the Mixture of Gaussians method performed the best overall and was the one we used. It finds most of the areas of interest in the IR images while ignoring background objects. It was superior to the other segmentation methods on the visual image.

The results from the object matching algorithm were highly promising, especially the first score which gave a 98.2% true positive rate and a 98.7% true negative rate. The second score also provided reasonable results with a 93.6% true positive rate and a 93.5% true negative rate. While it may seem that score one could be used without score two without any loss of performance, there are some special cases where score two should perform better. For instance, when there is a large segmentation area in the visual image but a smaller one in the IR image that fits within the one in the visual image. Score one will then get a maximum value while score two will hopefully give a smaller value provided that the edges between the images do not correspond.

The three presented measures also performed well, producing true positive and true negative rates at around 90%. Using the three different measures together with the scores from the object matching algorithm the idea is that we get a robust method where different measures complement each other. When weighting

all the measures and thresholding on our manually classified objects we got a 98.9% true positive rate and a 99.7% true negative rate. This is quite satisfying and shows the usefulness of the methods. However, it should be noted that the scenarios used to obtain these values were similar, making the parameters specific for our data.

Methods for separating objects concatenated in the segmentation, which we did not investigate in as much depth, seems promising and produced good results for separating humans from other objects. The problem was to determine if there were more than one class of objects in a segmented area. Using the approach of counting peaks in the intensity histogram we got 75.1% of the class A, 95% of the class B and 70.9% of the concatenated objects classified correctly. Although, for many of the class A objects classified as concatenations it was still desirable to run the clustering algorithm since the areas included surroundings as well and it resulted in finer segmentation. It is not a perfected method but it could improve the performance of the overall method significantly.

One overall conclusion we draw from working on this thesis is that the idea of combining visual cameras with thermographic cameras has the potential of leading to algorithms to improve video surveillance.

6 Future Work

There are many things that can be improved and developed related to the fusion of a visual camera and a thermographic camera that were not within the scope of this thesis. In this thesis we have mainly focused on the application of privacy masking a window and unmasking human beings that walk in front of the window (but not reflections), but there are other applications as well.

Another use of our sensor fusion that is related to the privacy masking application is the intriguing idea of identifying the mask area, i.e. the windows, automatically. This is a complex problem but an approach would be to try to identify the windows by trying to find the areas in the IR frames where there are reflections occurring (assuming reflections can be identified) and where there could be any sort of movement in the visual frames. This could be combined with edge detection in both IR and visual to get an initial guess of where a window could be. While this is an abstract idea, our perception is that it could be feasible.

In trying to classify humans and reflections separately we were able to get around 99% correct classification. Assume that we have 30 frames per second and an average of three objects per frame, meaning that on average there will be a misclassification of an object approximately every second. If these misclassifications are (unrealistically) assumed to be equally distributed across time they will appear as flickering in the masking/unmasking of an object, meaning that every second (in one of the 30 frames) there is an object that gets incorrectly

masked/unmasked. This might be prevented by implementing some form of object tracking so that an object is not allowed to suddenly shift class. This could be linked to the measures that are the basis of the classification, if the measures only move slightly above/under the threshold one could require them to stay there for a longer time before the class label is changed. Implementing object tracking should greatly improve the robustness of the methods presented in this thesis. Tracking could also be used as a mean of joining areas that have been incorrectly separated in the segmentation. For instance, when a head has been separated from a body in the segmentation this could be detected and corrected for.

There are many other possible improvements to our methods such as finding better and more robust measures and optimizing the parameters further. Larger datasets and more varying test scenarios would allow deeper analysis on how specific the parameters need to be for a scene. A further development could be to make the parameters adaptive by estimating them continuously.

In a future product two sensors, one that is sensitive to the visual spectra and one that is sensitive to the LWIR spectra, could be built into the same camera housing receiving the same light. This would greatly improve the performance of the algorithms as there is no need for registration.

References

[1] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[2] James Byrnes, editor. *Unexploded Ordnance Detection and Mitigation*. Springer, 2009.

[3] P. KaewTraKulPong and R. Bowden. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. September 2001.

[4] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[5] N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[6] J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G.A. Jones. Adaptive eigen-backgrounds for object detection. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1847–1850 Vol. 3, 2004.

[7] Robert Siegel and John R. Howell. *Thermal radiation heat transfer*. Washington : Hemisphere, cop. 1992, 1992.

[8] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 246–252 Vol. 2, Los Alamitos, CA, USA, August 1999.

[9] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 1st edition. edition, November 2010.

[10] N Yoder. *PeakFinder*. MATLAB Central File Exchange, June 2011.

Master's Theses in Mathematical Sciences 2013:E45
ISSN 1404-6342

LUTFMA-3253-2013

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/