Segmentation and Scene Content in Moving Images

Problem presented by

Glynn Wright and Laurence Broadbent

Aralia Systems Ltd.

ESGI107 was jointly hosted by
The University of Manchester
Smith Institute for Industrial Mathematics and System Engineering

with additional financial support from
KTN Ltd
Natural Environment Research Council
Manchester Institute for Mathematical Sciences

Report author

Dmitry Badziahin (Durham University)
Ferran Brosa Planella (University of Oxford)
Marina Ferreira (Imperial College London)
Silvia Gazzola (University of Padova)
James G. Herterich (University of Oxford)
Armin Krupp (University of Oxford)
Sean W. X. Lim (University of Oxford)
Stan Miklavcic (University of South Australia)
Rafal Pronko (Centre for Industrial Applications of Mathematics and Systems)
Jim Skinner (University of Warwick)

Executive Summary

The problem of scene content in moving images was brought by Aralia. The goal in this study group was to consider two problems. The first was image segmentation and the second is the context of the scene. These problems were explored in different areas, namely the Bayesian approach to image segmentation, shadow detection, shape recognition and background separation.

Version 1.0
May 1, 2015
iv+29 pages

Contributors

Niall Bootland (University of Oxford)
Sophia Coban (University of Manchester)
Asbjørn Nilsen Riseth (University of Oxford)

Contents

1 Introduction

(1.1) Automated scene content analysis is a software analysis programme that is widely used in the surveillance and security industry. It is the detection and determination of events, both spatially and temporally, from a scene. The motivation for the software is the ability of the biological visual cortex to separate a spatio-temporal event of interest (the foreground) from the uneventful scene in which it takes place (the background). The primary application of this software, for Aralia, is to provide high-tech security solutions in airports, rail networks, cities, ports and many more locations.

(1.2) Aralia's technology is specifically designed for object detection and tracking, perimeter surveillance, left luggage detection, crowding, loitering, infractions and counting statistics. The scene analysis consists of a number of steps:

1. comparison of the first-order derivatives of current and average scenes to determine regions that may contain objects that have moved in the scene,
2. identification of all pixels in the scene that are significantly 'different' from the average background,
3. rigorous segmentation of the changed pixels,
4. a coalescing process on the segments that reconstructs moving objects in the scene,
5. classification of the objects from their characteristic signatures deduced from the coalesced segments, and
6. testing of the classification results against their scene context.

(1.3) This strategy has a number of advantages. By comparing the first-order derivatives, that is, the changes from one frame to another, it is only the moving regions of the scene that are analysed. Hence, moving objects are easily detected and separated from the more static background. Furthermore, by working solely with the moving images, this may significantly increase the efficiency of the analysis as this often represents a small proportion of the scene. Analysis, including segmentation, of the changing pixels due to a moving object allows for object classification without resorting to more complex algorithms. Finally, by testing the classifications against the scene context, Aralia report a reduction in the incidence of false classifications by an order of magnitude.

(1.4) However, there are also a number of drawbacks to this strategy. An object that is brought into the scene may be detected by its movement. However, if such an object remains stationary in the scene over a long period of time, it becomes difficult to detect. In effect, it may become part of the background.

In many cases it becomes difficult to determine the scene context, the background. Shadows, reflections and weather play an important role in a scene, both spatially and temporally, *e.g.*, shadow in an outdoor scene moves and changes in size slowly during the day, yet a sudden snowfall changes a scene dramatically in a short time interval. In addition to this, the intensity of natural light changes during the day, distorting the colour information. Moving surveillance cameras present a further problem by removing a fixed reference frame for the background. Finally, segmentation loses its precision for objects that are close to one another.

2 Problem statement

2.1 Scope of the Problem

(2.1) The goal of the study group is to consider some proposed techniques to improve the information extracted from a scene, as well as exploring new techniques for the problem.

(2.2) There are two main areas of the problem that Aralia wish to focus on. The first is on image segmentation. Currently, it is performed by maintaining two moving averages of the scene. Aralia propose segmenting the moving images and then coalescing them using additional information such as a 3D depth map, a 2D or 2.5D texture of principal component analysis of illumination. An important part of image segmentation to be considered is shadow detection.

(2.3) The second area is the context of the scene. Aralia apply scene content rules, typically to prevent impossible situations from arising. This includes people moving at too high a speed or objects changing their classification. Aralia wish to extend context analysis by identifying specific objects within the scene. Identification of the types of static objects in a scene may then be useful as a method to identify or narrow-down the full context of the scene.

2.2 Aspects considered during the Study Group

(2.1) We consider a new technique for image segmentation, using a Bayesian inverse problem approach. This requires a prior, *i.e.*, an idea of the background context of the scene. This is achieved using an averaging process of the raw video data. A further method for image segmentation is via a low-rank sparse matrix decomposition, specifically a Robust Principal Component Analysis.

(2.2) Shadow detection and removal is considered using a variety of techniques. We discuss their merits and drawbacks, and consider improvements using a combination of techniques.

(2.3) Object classification is explored via Deep Learning algorithms.

3 Image Segmentation

3.1 Inverse Problem Formulation

(3.1.1) We consider the simple problem of background and foreground separation in image segmentation. Given the foreground at time n, denoted by a column vector u_f^n, the foreground at time $n+1$ is given by

$$u_f^{n+1} = u_f^n + \eta^{n+1}, \tag{1}$$

for some η^{n+1}. The image, denoted by a column vector u^n, is being observed, and is given by

$$u^n = u_f^n + u_b^n, \quad n = 1, \cdots, N, \tag{2}$$

where u_b^n is a column vector which denotes the background at time n. We model the background as a random term, governed by a Gaussian distribution, that is, we have

$$u_b^n \sim N(\overline{u_b^n}, W_n^{-1}), \tag{3}$$

where $\overline{u_b^n}$ and W_n^{-1} is the mean and covariance of the background at time n. The task is then to estimate u_f^n for $n = 1, \cdots, N$.

(3.1.2) Since it is unclear how η^{n+1} is determined, it would seem reasonable to model η^{n+1} as a random variable governed by a probability density function. Nevertheless, for simplicity in this problem, we use an averaging technique described in Section 3.3 to compute η^{n+1}.

3.2 Solution using Bayesian Inference

(3.2.1) The Bayesian approach to solve an inverse problem involves treating the parameters and variables as random variables. In this sense, the task is to determine the probability density for u_f^n given the image.

(3.2.2) In order to do this, we first specify a prior density for the initial state, u_f^0. We construct a Gaussian for the prior density, with mean $\overline{u_f^0}$ and covariance L^{-1}. The probability density of u_f^0, $\pi(u_f^0)$, is given by

$$\pi(u_f^0) = z \exp\left(-\frac{1}{2}(u_f^0 - \overline{u_f^0})^T L (u_f^0 - \overline{u_f^0})\right), \tag{4}$$

where z is a normalisation constant.

(3.2.3) Then, it follows that the foreground at each time step, u_f^n, is Gaussian, and the mean and precision (or inverse covariance) updates are given by

$$L_{n+1} = W_n + L_n, \tag{5}$$

$$L_{n+1}\overline{u_f^{n+1}} = W_n(u^{n+1} - \overline{u_b^{n+1}}) + L_n(\eta^{n+1} + \overline{u_f^n}). \tag{6}$$

For more details, we refer the reader to [1].

3.3 Updating the Background

(3.3.1) The background is required as a prior for the Bayesian inverse formulation of the problem. We consider two approaches to calculating the background image by way of first-order changes in the scene. First-order changes are significant differences, at the pixel level, between subsequent frames.

3.3.1 First Approach

(3.3.2) Our first approach to separating the background and foreground (1) arises from considering how the background scene changes due to a moving object. An object of interest in one frame appears as a still object. It covers up a part of the background scene, blocking all information behind it in that one frame. In a subsequent frame, this moving object now covers up a different section of the background scene. We can track the movement of the foreground object by comparing subsequent frames, whereby the moving object of the foreground is the first-order change in the scene between two frames,

$$u_f^{n+1} = u^{n+1} - u^n. \tag{7}$$

(3.3.3) Working with this sequence of foreground frames, $\{u_f^n\}$, tracking the moving image, we can extract a sequence $\{u_b^n\}$ of partial background images available in each frame from (2), where

$$u_b^n = u^n - u_f^n. \tag{8}$$

(3.3.4) As described, each frame in the sequence of background images, $\{u_b^n\}$, gives a partial description of the background. However, as the foreground object moves in the scene, it reveals background from behind and covers background as it moves forward. The full background may be computed by running a long-time average of the background sequence, $\{u_b^n\}$,

$$\hat{u}_b^n = \frac{1}{n}\sum_{i=1}^{n} u_b^i, \tag{9}$$

where $\hat{u}_b^n$ is the average background after n frames.

(3.3.5) This method is successful when the number of frames is large so that the background information lost by an object moving in the scene is averaged out, in the sense that there are many more frames in the video that include the actual background in each location.

(3.3.6) There are limitations to this method, such as if an object moves into the scene, or is moved within the scene, but then remains still in the scene, *e.g.*, a person walking on a platform and stopping to wait for a train, or a chair is moved to another location. The algorithm gradually recognises the object by its outline, with the actual background appearing within this outline. In essence, it is like a ghost in the scene. It eventually becomes part of the background. This can be seen in Figure 1.

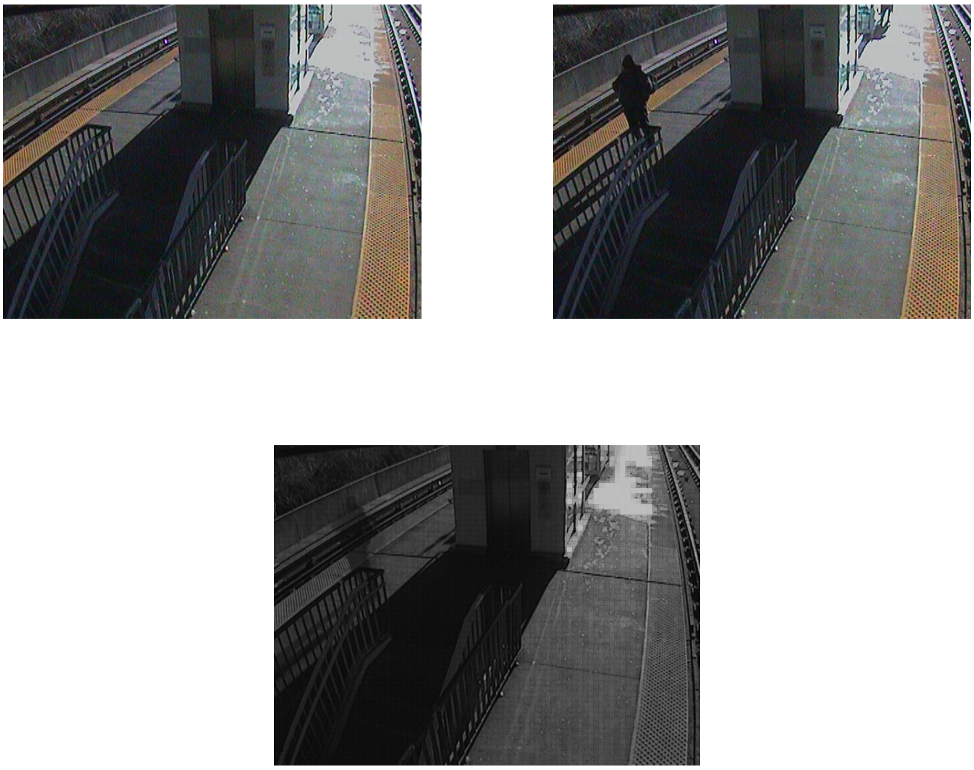

Figure 1: An empty scene (top left) shows the full background image of a train platform. After some time, a person enters the scene, walks around and eventually stands still on the platform (top right). The algorithm of computing the partial background average (9) recognises the person. However, when the person remains static for a period of time, their outline become part of the background (bottom, in greyscale).

(3.3.7) A solution to this problem may be to keep a running average of the background, but also more local (temporally) averages of the background, for comparison. These more local averages are less affected by the change in daylight and movement of shadows. Further comparison, by first-order changes between the local and long-time averages may give insight to how the background has changed over time.

3.3.2 Second Approach

(3.3.8) The second approach takes into account the appearance of new objects, or movement of existent objects, that remain static in the scene for a significant period to time.

(3.3.9) From (1) and (2), we have

$$\eta^{n+1} = (u^{n+1} - u^n) - (u_b^{n+1} - u_b^n). \tag{10}$$

For this reason, it is reasonable to take η^{n+1} to be the difference of the change in current and previous images, and the change in the mean of current and previous backgrounds. We then have

$$\eta^{n+1} = (u^{n+1} - u^n) - (\overline{u_b^{n+1}} - \overline{u_b^n}), \tag{11}$$

where the background mean is updated as described in the following paragraphs.

(3.3.10) At time $n+1$, we want to update the background, $\mathbf{u}_b^n$, using the new observation, $\mathbf{u}^{n+1}$. To use this method we need an initial background $\mathbf{u}_b^0$ which we will update to capture the variation of the background. Treating the greyscale images as integer matrices, the equation to update the background can be written as:

$$\mathbf{u}_b^{n+1} = \frac{c_n \mathbf{u}_b^n + \mathbf{u}^{n+1}}{c_n + 1} \circ (\mathbf{I} - \boldsymbol{\alpha}_{n+1}) + \mathbf{u}_b^n \circ \boldsymbol{\alpha}_{n+1} \tag{12}$$

where $\circ$ denotes the Hadamard product and $\mathbf{u}_b^n$ and $\mathbf{u}^n$ are the matrix representations of the pixel values of the background and whole images respectively. The matrix $\boldsymbol{\alpha}_{n+1}$ is defined as $\boldsymbol{\alpha}_{n+1}(i,j) = f\left(|\mathbf{u}^{n+1} - \mathbf{u}_b^n|(i,j)\right)$ where $f(x)$ is a non-decreasing function that satisfies $f(0) = 0$ and $f(x) = 1 \ \forall x \geq \sigma$. A non-zero value reflects a significant change in pixel value at that point in the image, suggesting an object is moving. The constants, $\{c_n\}$, are parameters of the algorithm.

(3.3.11) The main goal behind this algorithm is to take advantage of the good properties of averaging the values in the background but removing what we call ghosts. Ghosts are dark blurry silhouettes that appear when a moving object stands still for enough time. This is because when averaging the value of the pixels to get the background, the moving object introduces a steep variation of the value of those pixels and distorts the average. The idea is to take the weighted average of the background and the new image only for the pixels where the change is smaller than a threshold, σ, determined by $\boldsymbol{\alpha}_{n+1}$. This should allow us the capture the changes in illumination but remove the ghosts. This method can be easily extended to colour images using three-dimensional arrays. Working in colour images usually provides better results as we compare colours in a three-dimensional space instead of their projection to a one-dimensional space.

3.4 Examples

(3.4.1) We applied the method of Bayesian inference to a dataset provided by Aralia for the purposes of the study group. We used 581 frames of size 640×480, and a time-independent background mean was used, defined to be a frame in the dataset, and the covariance of the background was set to be a diagonal matrix for simplicity, where each component variance computed as the variance of the background mean. This corresponds to considering the background as white noise.

(3.4.2) The prior mean is the zero vector, with prior precision matrix $L = 0.01I$, where I is the appropriate identity matrix, and L^{-1} is the covariance. This represents a white noise prior with component-wise variance of 100. We updated the mean and precision using the update formula given in (5) and (6). Although the dataset given were frames of RGB, for simplicity and limited computational resources, we decided to work on the greyscale frames instead.

(3.4.3) For the dataset 11817BWI_139, we began from frame 0901 and used frame 0900 as the background mean. Figures 2–6 show scaled image plots at particular time steps.

(3.4.4) At observation $n = 151$, we observe a man coming onto the platform holding a suitcase, and this is segmented as the foreground by the algorithm. The man then sets down his suitcase and therefore it remains stationary for long periods of time. In the following frames, at $n = 251$, 351 and 451, the bag remains stationary but does not fade into the background, as the knowledge that the bag is part of the foreground is preserved by the algorithm. At $n = 451$, we observe the shadow of a person coming into the scene. In the image, only his shadow is visible. Such an observation questions the necessity of shadow detection and removal procedures as it may remove significant information in a scene.

3.5 Future work

(3.5.1) We have considered a simple case of image segmentation in this section,

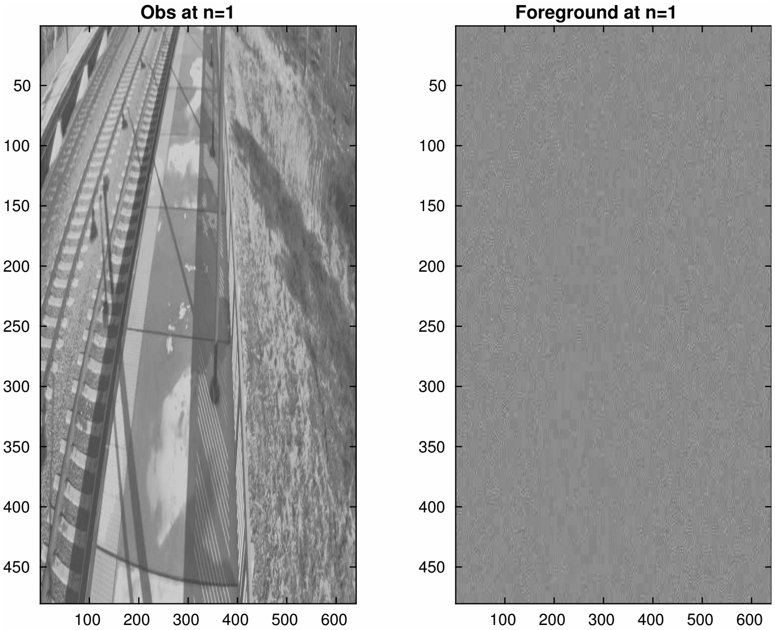

Figure 2: Observed image and foreground after 1 observation.

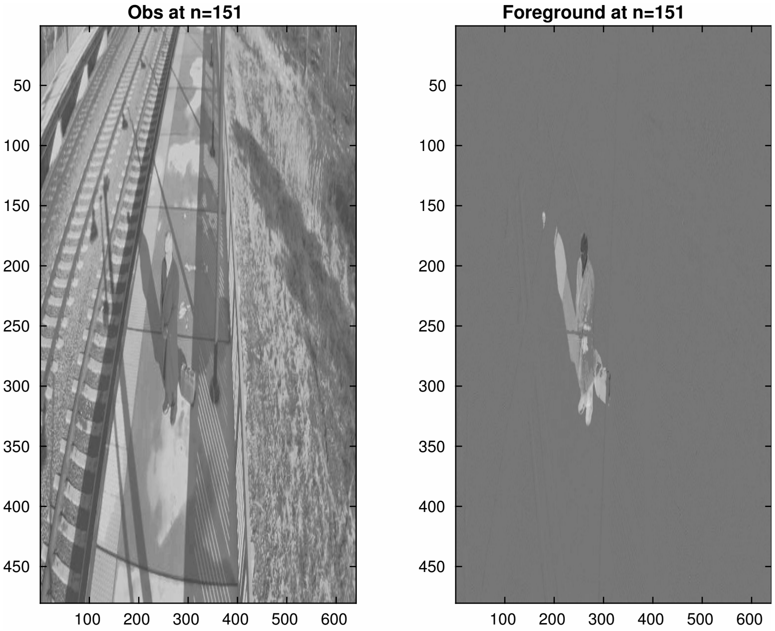

Figure 3: Observed image and foreground after 151 observations.

and defined it as the separation of the foreground from the background. One possible extension of this approach would be to rerun the simulations on the dataset using a time-dependent background mean and covariance, where the mean and covariance will be updated with time. At the moment, numerical simulations suggests that the approach detects shadows of stationary objects in the background which move through the day as the position of the Sun changes. In this regard, Section 4 may be useful

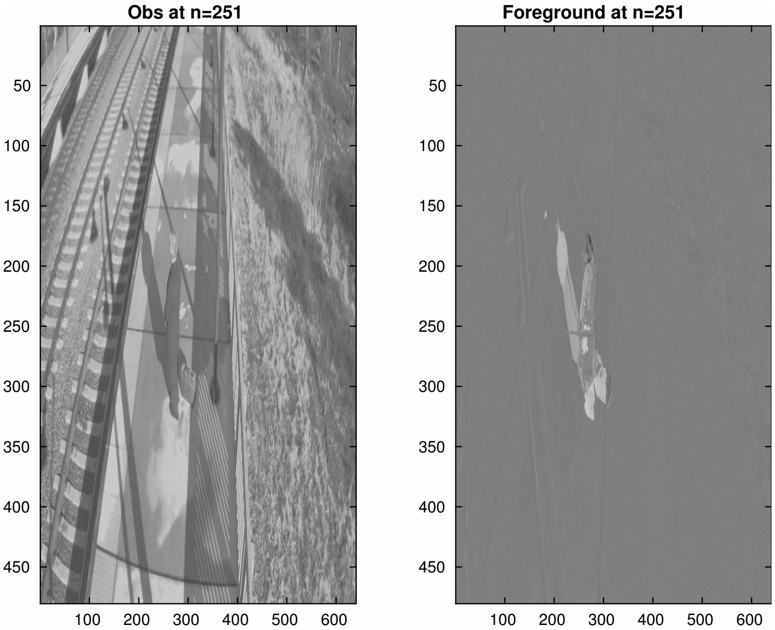

Figure 4: Observed image and foreground after 251 observations.

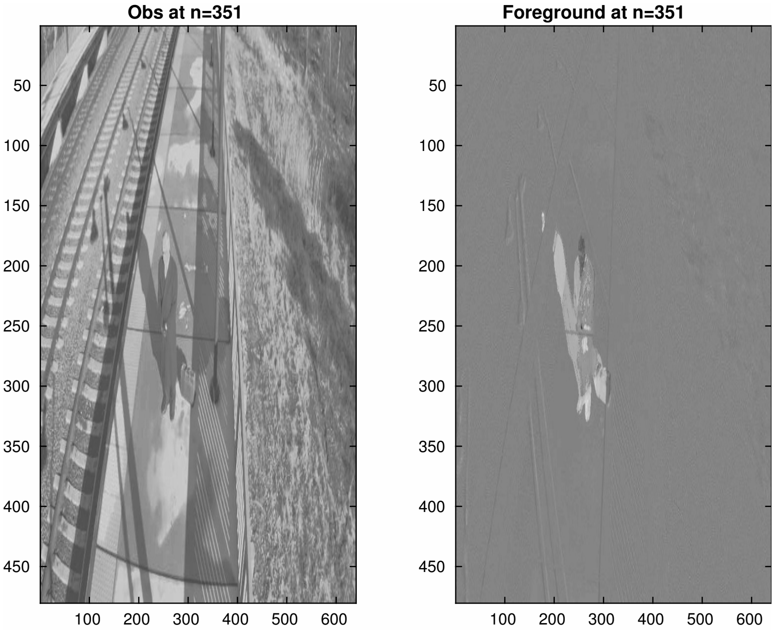

Figure 5: Observed image and foreground after 351 observations.

to eliminate such artefacts to improve background and foreground separation. Nevertheless, it is questionable if the shadow removal techniques should be applied to remove all shadows in a scene, in particular shadows of actively moving objects in the foreground, as it is believed that such shadows may content significant information within a scene. It is suggested shadow removal techniques should be embedded into the image segmentation algorithm described here to segment the background, foreground

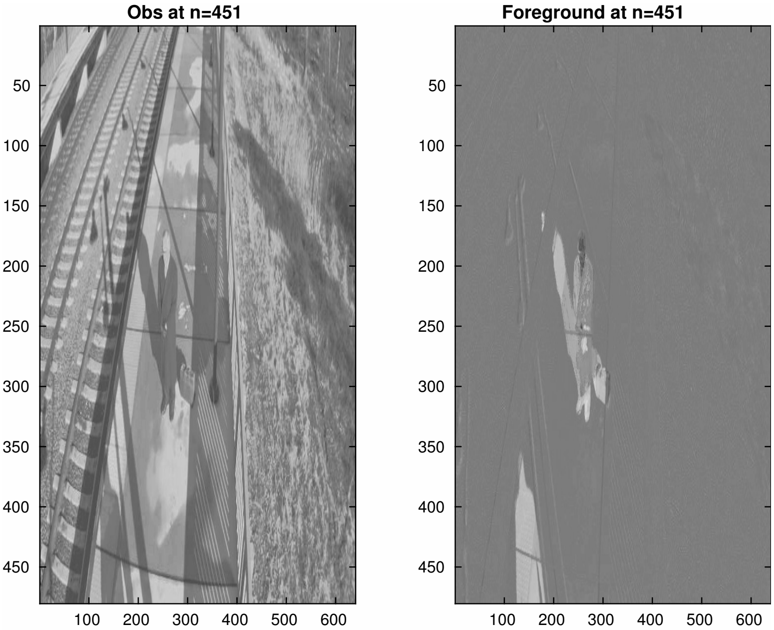

Figure 6: Observed image and foreground after 451 observations.

and shadow instead of merely the background and foreground. This will enable us to keep track of the shadows, will should be more beneficial in terms of extracting scene content from images.

(3.5.2) Finally, more research could be done to examine the Bayesian approach to image segmentation and scene content, as it provides a systematic framework to integrate the prior knowledge of a particular scene with current observations.

4 Shadow Detection and Removal

4.1 Introduction

(4.1.1) Shadows have been a big issue in image processing over the past few decades, in particular due to the undesirable problems they bring to image segmentation algorithms. In surveillance systems at a train station, for instance, or on a moving overground, the presence of shadows may jeopardize an effective detection of immobile luggage that may constitute a potential threat.

(4.1.2) Our aim is to detect and remove shadows from a single 2-D image. Without 3-D perception, we can only rely on the colours to detect shadowed regions. A shadow can be defined as a part of the image that is not directly illuminated by a light source due to an obstructing object. Based on the intensity, the shadows can be classified as soft (retaining the texture of the background surface) and hard (retaining almost no texture). From a single 2-D image it is difficult to distinguish between hard shadows and dark objects. Therefore, we will focus on soft shadows only.

(4.1.3) In the following, we briefly discuss two methods proposed in the literature ([11], [12]). Subsequently, our approach "Shadow removal using the Orthogonal Plane" is presented, followed by some suggestions and improvements.

4.2 Shadow detection using the $\mathrm{YC_bC_r}$ colour space

(4.2.1) We implemented a method for shadow detection based on the one proposed in [11], which uses the luminance Y as follows:

- convert the image from the RGB colour space to the $\mathrm{YC_bC_r}$ colour space, where $\mathrm{C_b}$ and $\mathrm{C_r}$ denote the blue-difference and red-difference chroma components;
- compute the average μ and the standard deviation σ at Y channel;
- the pixels with luminance below $\mu - \sigma$ are classified as shadow points.

(4.2.2) This method produces a binary matrix with the classification of the shadow (white) and non-shadow regions (black) which can be seen in Figure 8.

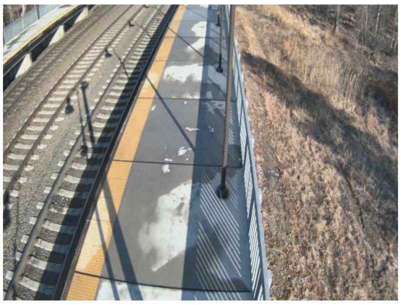

Figure 7: Original image of a train station platform.

(4.2.3) The detection works reasonably well, however it appears to fail to detect the shadow in the snowy regions (Figure 8).

4.3 Shadow removal using the LMS colour space

(4.3.1) We studied a paper on visual difference prediction from psychologists [12], suggesting a special transform of the pixels of the image based on the way the visual cortex perceives colour:

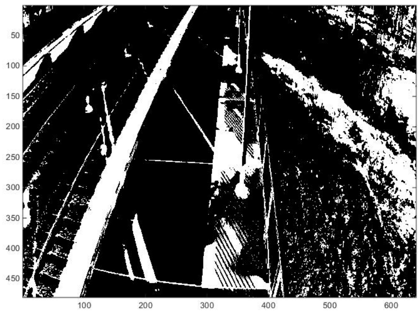

Figure 8: Binary shadow image.

- convert the image from the RGB colour space to the LMS colour space;
- compose a greyscale image with pixels equal the following values: $\frac{L-M}{L+M}$.

(4.3.2) The LMS color space represents the response of the three types of cones of the human eye, named after their responsivity (sensitivity) at long (L), medium (M) and short (S) wavelengths.

(4.3.3) It is believed that this transformation makes shadows less visible, helping the human being to distinguish them from real objects. This approach is promising for the first sample provided (Figure 10). However, this method does not seem to work well in general as we can see in Figure 11, although it is not clear why this is so.

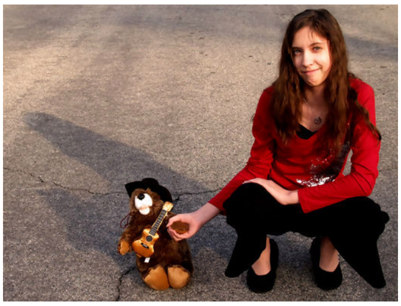

Figure 9: Original image of a girl.

4.4 Shadow removal using the Orthogonal Plane

(4.4.1) The observation that the previous transformation works apparently quite well for some images but not so well for others suggests the use of a different transformation on each image. We then tried to compute the transformation that minimizes the presence of shadows as follows:

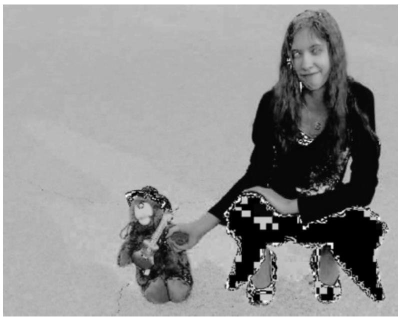

Figure 10: Image obtained from Figure 9 by applying the transformation described in [12].

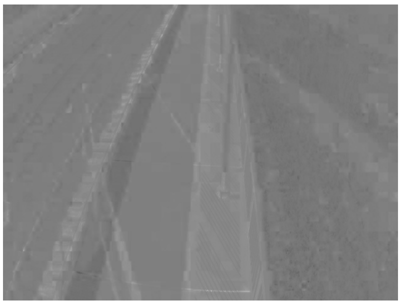

Figure 11: Image obtained from Figure 7 by applying the transformation described in [12].

- Pick two pixels in the RGB colour space from the image (this process may be automated, see Section 4.5): pixel $\mathbf{p}_1$ of the shadowed region and pixel $\mathbf{p}_2$ of the non-shadowed region of the same surface;
- Compute the vector $\mathbf{v} := \mathbf{p}_2 - \mathbf{p}_1$ and the plane Π orthogonal to $\mathbf{v}$;
- Project all pixels of the image onto Π.

(4.4.2) The idea is that the shadowed and unshadowed pixels will project to the same point on a plane and therefore they will look the same in the resulting image.

(4.4.3) However, we encounter some problems. Various pixels of the shadowed and unshadowed regions vary in colour. It is not guaranteed that other shadowed and unshadowed pixels, different from $\mathbf{p}_1$ and $\mathbf{p}_2$, will still project to the same point. However we hope that their projections will still be very close to each other and the visibility of the shadow will be very low. This problem may be minimized by subdividing the image and applying this method to each part.

(4.4.4) Another problem is the light variation within the image and, consequently, the existence of different types of shadows. The vector $\mathbf{v}$ will depend greatly on the source of light and significantly on the reflection properties of the particular surface from where $\mathbf{p}_1$ and $\mathbf{p}_2$ were choosen, consequently,

v will be optimal for one shadow but may not work for other shadows. This problem may be overcome by adjusting the vector **v** depending on the mean luminance of the image (or some other global parameter of it). We did not have time to investigate this correction.

(4.4.5) Finally, for some images the projection becomes too blurred. In that case the resulting image becomes useless for segmentation algorithm. In that case some other algorithm is needed to remove the shade.

(4.4.6) We applied the procedure to both images from Figures 7 and 9. This produced the results depicted in figures 12 and 13, respectively. As we can see, the shadow is indeed much less visible as expected, however it does not disappear completely, especially near the boundary with the non-shadow regions (see figure 12). As in [11], we may smooth the boundary by applying a Gaussian mask. This may increase its applicability to image segmentation algorithms. On the other hand, contrarily to the image in Figure 11 produced by the previous method, the image in Figure 12 is much more clear.

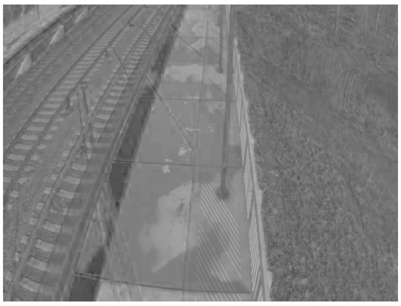

Figure 12: Image obtained from Figure 7 by applying the method of the Orthogonal Plane.

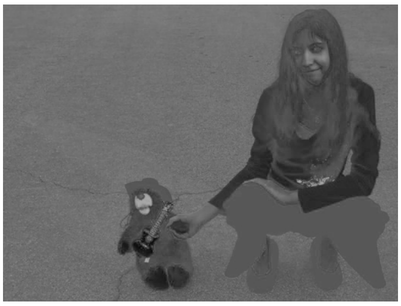

Figure 13: Image obtained from Figure 9 by applying the method of the Orthogonal Plane.

4.5 Improvements and future work

(4.5.1) In order to automate the first step of picking two pixels in Section 4.4, we may use the shadow detection algorithm described in Section 4.2 as follows: two adjacent points of the binary matrix with values 1 and 0, respectively, are selected. Those points correspond to two pixels on the same surface with and without shadows, respectively.

(4.5.2) Finally, combining the ideas discussed before we get the following procedure:

- Shadow detection using the $\mathrm{YC_bC_r}$ colour space (computing the binary matrix);
- Subdivide the image in parts and compute a “good” orthogonal plane for each part by using the binary matrix;
- Shadow removal using the Orthogonal Plane;
- Apply a Gaussian mask to the shadow’s boundary.

(4.5.3) There was no time to implement this last method nor the corrections and improvements discussed above, whereby we leave them for future work. Nevertheless, since the idea behind the ”Shadow removal using the Orthogonal Plane” is to find a transformation that makes the shadowed and unshadowed points to look the same we expect it to work reasonably well in general. This procedure should now be more rigorously compared to other methods presented in the literature, in particular with the one proposed in ”Describing Reflectances for Colour Segmentation Robust to Shadows, Highlights and Textures” [13], which seems to be efficient as well as quite successful on dealing with shadows.

5 Shape Recognition

5.1 Deep Learning

(5.1.1) An approach for shape recognition is to classify segmented objects by determining their *shape signature* and then comparing it against an existing set of known shape functions. To determine the shape signature of some closed area Ω, we assume that Ω is star-shaped with respect to its centre of mass $\boldsymbol{O}$. Then, we can parametrize the boundary $\partial\Omega$ with respect to $\boldsymbol{O}$ using some function $r(\theta), \theta \in [0, 2\pi)$, and so we call $r(\theta)$ the shape signature of Ω.

(5.1.2) For a given set of parametrizations $\{s_1, \ldots, s_n\}$ of shapes, we can reformulate the problem of recognizing a shape with given signature f as the minimization problem

$$\min_{s\in\{s_1,\ldots,s_n\}} \min_{\theta\in[0,2\pi)} \left\| \frac{f}{s} - \frac{1}{2\pi} \int_0^{2\pi} \frac{f(\theta+\phi)}{s(\phi)} \mathrm{d}\phi \right\| . \tag{13}$$

Note that the shape signature is unique, if the given area satisfies the conditions outlined above. Also, the minimization problem is independent of the scaling of the signature. Recognizing a shape by its signature is a valuable theoretical concept but is not suitable for real-world applications, as the shape signature heavily depends on the underlying image segmentation and comparing a given signature against a set of signatures is computationally expensive. Furthermore, shapes in real-world images are usually projections of three-dimensional objects, thus they depend on the angle of the projection of the object. To make use of the shape signature in this instance, a catalogue of all signatures for every projection angle would be necessary, this is in general not feasible.

(5.1.3) Motivated by the famous problem "Can one hear the shape of a drum?", we also thought about characterising shapes by solving a linear, time-dependent scalar partial differential equation (PDE) with the shape being the domain and zero Dirichlet boundary conditions and then using the average value over time as the signature. As an example, we solved the heat equation

$$u_t - \Delta u = 0, \tag{14}$$

subject to

$$u(\mathbf{x}, 0) = img(\mathbf{x}) \quad \text{and} \quad u = 0 \text{ on } \partial\Omega, \tag{15}$$

where $img(x)$ is the $\{0,1\}$-valued function arising from the image segmentation, and used the average temperature $\bar{u}(t)$ as the signature of the shape Ω. This ansatz has the advantage that it is independent of the 2-D rotation of the shape and is more stable with regard to noise than the shape signature, it also does not impose any conditions on the shape of the area. However, the signature obtained by solving the heat equation is no longer unique and at the same time numerically expensive. Nevertheless, the general idea, combined with some refined PDE, might be of interest for offline classification problems.

6 Background separation

6.1 Motivation

(6.1.1) Given a surveillance video, we want to be able to separate out the static background from the dynamic foreground, both of which may be of interest. Separating the background from the foreground may provide useful context in object classification, since very different objects exist in the background (chairs, signs, rail) and foreground (people, bags).

(6.1.2) Whilst the background is almost constant, the fact that it is expected to change slightly throughout the day—e.g., due to the sun—means that we cannot simply look for a video frame of an empty station and use this as the background. The reason for this is that we are obtaining the foreground by subtracting the background from the scene, which would pick out regions of changed lighting as foreground elements. In addition, we would like our method to be as widely applicable as possible, and do not want to make the assumption that an empty scene is ever observed.

6.2 Inpainting Method

(6.2.1) The intention of this approach is to generate a set of baseline images that could be used to represent the background of a particular scene and to represent this background in a reduced format by means of a Principal Component Analysis (PCA). By comparing a scene's principal components (background and background+foreground), we hypothesize on the possibility of quickly detecting any aberrations of a scene, such as the presence of anomalous stationary objects or moving objects (people). In some ways the initial stage of the analysis, which we have termed "Inpainting", overlaps with the method of the next section, but in other ways is distinct. During the week of the study group we only succeeded to complete the first element of the program: that of developing an in-principle method of generating a set of baseline images. The second goal, of developing an algorithm that would extract for comparison a set of underlying principal components of the background image set, was only was drafted but not tested.

(6.2.2) With regard to the overall aim of the proposal, there are some of difficult aspects that need to be considered.

1. Firstly, there is the problem of achieving the initial goal of producing a set of clean background images. In an operational setting it is not possible to assume that one can identify a time instance when the scene in question is devoid of extraneous objects.

2. Secondly, it is not possible to assume that one knows a priori how a scene is supposed to appear. That is, any devised method must work independent of knowledge of stationary object features. Indeed, there exist the possibility that elements of the background scene (location, form and structure of public seating or litter bins, or signage, etc.) could be altered during filming. The possibility of such sudden scene changes need to be considered and a means of modifying the background set of images and new principal components.

3. Finally, for outdoor scenes one needs to consider the natural daytime variation resulting from the movement of the sun (the principal light source), such as the associated movement of shadows

of stationary objects. Similarly, there is a possibility that between dusk and dawn, new light sources, i.e., the onset of artificial lighting becomes significant and eventually dominant, which will introduce sudden changes to the scene.

(6.2.3) Such complications require consideration in any analysis but particularly in the implementation of the method proposed in this section. The natural variation of a scene due to sunlight changes, for instance, can be accommodated by the procedure outlined below. For surveillance cameras capturing an outdoor public transport facility, the aspect of lighting condition and shadow position changes following the relative movement of the sun's position can be taken into account by accumulating a finite set of background images that is continually being updated with time. The premise for this is that natural lighting variations and stationary object shadow movements occur on a much longer timescale than variations due to embarking and disembarking commuters or arrival and departure of transport vehicles. Thus, with a continually updated set of background images one can produce an evolving set of representative principal components, that reflects the long term time scale variation in lighting and shadow position. Sudden changes in background object setting due to human intervention or sudden light source changes may require more sophisticated approaches.

6.2.1 Inpainting: Algorithm and Implementation

(6.2.4) During the week of the Study Group, we did not attempt to produce a fully automated way of identifying an initial foreground-free image. Instead, we generated a set of background images based on the assumption that at least one image is available. The problem of producing this image autonomously remains to be solved. Thus, the underlying assumption for the method proposed in this section is that at least one clean background image is available. Let this image be denoted by $\mathbf{I}_0$.

(6.2.5) The fundamental goal here was to develop a procedure that would generate a set of training images that could be used to fully characterize the background of a video scene. As mentioned above, one fundamental difficulty that needed to be overcome was that of establishing a portfolio comprising a sufficient number of images containing only background information. Thus, the need was to capture only images devoid of any transient objects (people, trains, etc., moving in or out of the scene) that would otherwise interfere with the background scene characterization. As also mentioned, the method devised needs to be mindful of any natural scene evolution such as long term lighting variations and shadow movement.

(6.2.6) The approach taken during the study group can be summarized in the following algorithm:

- Load a set of $n < N$ images, $\{\mathbf{I}_\alpha\}_{\alpha=1}^n$, from a video stream, where N is the number of frames. In a real application, this would performed in blocks, stepwise during operation.
- Identify at least one image, call it $\mathbf{I}_0$, which contains only background data. (This was assumed during the study group, but needs a preprocessing stage in an actual application.)
    - Set mean image $\bar{\mathbf{I}} = \mathbf{I}_0$.
    - Assign $\mathbf{I}_0$ to the set of background images, B.
- Compare sequentially image $\mathbf{I}_\alpha$ for $\alpha = 1, \ldots, n$ with image $\bar{\mathbf{I}}$: is $\Delta_\alpha = \|\mathbf{I}_\alpha - \bar{\mathbf{I}}\| < \epsilon$? Here, $\|\cdot\|$ is a given matrix norm and $\epsilon > 0$ is a threshold value.

Yes:
- Assign $\mathbf{I}_\alpha$ to B.
- Update $\bar{\mathbf{I}} = \frac{1}{m}\sum_{\beta=0}^m \mathbf{I}_\beta$, where $m = \text{size}(B)$.

No:
- If $\Delta_\alpha(i,j) \neq 0$, set $\mathbf{I}_\alpha(i,j) = \bar{\mathbf{I}}(i,j)$ for pixel (i,j).
- Assign revised $\mathbf{I}_\alpha$ to B.
- Update $\bar{\mathbf{I}} = \frac{1}{m}\sum_{\beta=0}^m \mathbf{I}_\beta$.

(6.2.7) The above algorithm is applied to each new block of m images in order to keep $\bar{\mathbf{I}}$ in sync with the long term variation of lighting conditions.

(6.2.8) An example of this process is shown in Figure 14 which depicts a typical original image, difference image and an updated background image (left, middle and right panels, respectively). Note that for simplicity in the implementation during the study group, we used the preceding, updated image $\mathbf{I}_{\alpha-1}$ as $\bar{\mathbf{I}}$ in the above steps rather than the true mean.

6.2.2 PCA of Background

(6.2.1) The aim of this exercise is to produce a succinct descriptive measure of the background to a scene. Using this measure, we should base any deviations in scene content. During the study group, only a skeleton code was developed. This was, however, not tested nor refined for any specific application. With regard to applications, there are several directions that could be considered. These are listed at the end of this section for future consideration.

(6.2.2) The method of principal component analysis (PCA), also called the discrete Karhunen-Loève transform, is essentially a statistical version of the eigenvalue-eigenvector representation of a discretized linear operator. Given a set of observations, the PCA attempts to represent the variables by a set of linearly uncorrelated "principal components". The representation

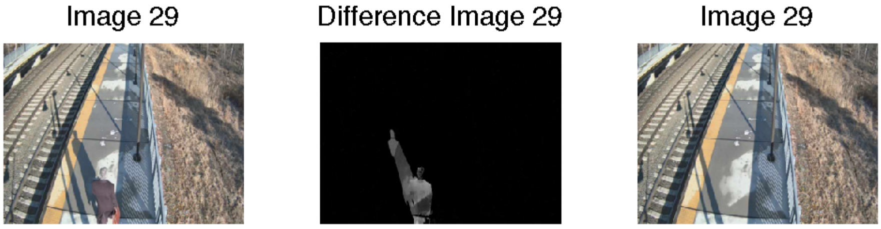

Figure 14: Example of the results of the Inpainting algorithm. Left panel shows a still from the original image sequence $\mathbf{I}_\alpha$, the middle panel shows the difference image Δ_α, which highlights the image of a man walking along a railway platform and the right panel shows the updated background with man and shadow deleted, $\bar{\mathbf{I}}$.

is such that the most significant component is the first one, subsequent components appear in order of decreasing importance.

(6.2.3) Consider a matrix of observations of an array of sensors, $\mathbf{X}$, where the n rows are sensor observations, $\mathbf{x}_{(i)}$, and the p columns represent the sensors. A PCA attempts to map the row vectors into new vectors which are the principal component scores $\mathbf{t}_{(i)}$:

$$\mathbf{t}_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$$

where the weights $\mathbf{w}_{(k)}$ are determined through the Rayleigh quotient operation:

$$\mathbf{w}_{(1)} = \arg\max \frac{\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}$$

with subsequent weights defined by first following a Gram-Schmidt orthogonalisationlike procedure defined via the ”orthogonalised” matrix

$$\widehat{\mathbf{X}}_k = \mathbf{X} - \sum_{m=1}^{k-1} \mathbf{X}\mathbf{w}_{(m)}\mathbf{w}_{(m)}^T$$

and then reapplying the Rayleigh quotient operation to $\widehat{\mathbf{X}}$.

(6.2.4) What is effectively achieved through this process is a matrix equation

$$\mathbf{T} = \mathbf{X}\mathbf{W}$$

where $\mathbf{W}$ is a $p \times p$ matrix of eigenvectors of $\mathbf{X}^T\mathbf{X}$.

(6.2.5) For the ultimate goal of a principal component analysis of our background, one may consider three possible directions. We outline these below. Although no option was properly explored during the study group, these options nevertheless offer opportunities to be followed up by the industry partner.

1. For the matrix $\mathbf{X}$, one could take the sequence of background stills determined through the previous process (Section 6.2), varying over a time period commensurate with the long term variation of lighting. Thus, the repeat observations are represented by the images in the set and the pixel content of each image (repeated across images) adopts the role of the sensor information. With this application the derived principal components would represent the variation of the scene with time.
2. As a variant of the previous option, one partitions the image into sections (both vertical partitions and rectangular block partitions were discussed). One then applies a PCA to each partition of the set of images. Again, the subset of principal components for each partition would describe the variation of that partition with time. The rationale for partitioning the image into sections is to achieve more efficient comparison with subsequent background+foreground images wherein sudden scene changes occur. The expectation is that not all of the image will be altered with the appearance of a foreground object. Consequently, only one or more affected partitions would need further scene analysis.
3. Considering the mean image, $\overline{\mathbf{X}} = (1/N)\sum_i \mathbf{X}_i$, from the training set, $\{\mathbf{X}_i\}$, of background images in Section 6.2 as the matrix $\mathbf{X}$, the principal components derived would represent, in concise form, the dominant characteristics of the background. This concise data would then form the basis of comparison with all new images, in order to rapidly identify a variation in the image. Given that the training set and hence the mean image would be continually updated as mentioned in Section 6.2,

the principal components would similarly evolve in time and provide a current descriptive summary of the background scene as lighting and shadows gradually change during the course of a day.

4. As a variation of the last item, PCA analyses could instead be applied to partitions of the mean image, $\overline{\mathbf{X}}$. The system of time-evolving, representative principal components would then be utilized in comparisons with corresponding partitions of new images. As in point 2 above, this would offer a more efficient program of further scene analysis.

6.2.3 Future Work

(6.2.1) Two aspects need immediate addressing. The first is to establish a means of generating at least one clean scene image as initial input into the training set of background images. We have worked on the assumption that such a scene exists, but a first step would be to create this image. It is possible that a variation of the scheme outlined in Section 6.2 could be employed for this purpose. The second aspect relates to a qualification of the intended application of the PCA method and its subsequent implementation as applied to the training set of background images.

(6.2.2) In this regard, there is the subsequent, additional ambition, to analyze the principal components so as to identify and classify scene objects for the dual purpose of (a) recognising and accommodating when an existing background-related object (such as a public bench) has been moved to a different location in the scene, and (b) identifying and recognising when new objects have been introduced into the scene (such as a left baggage). A Gaussian Mixture Model (GMM) approach applied to the principal components could be considered for this purpose.

6.3 Low-rank plus sparse matrix decomposition

6.3.1 Proposed approach: RPCA

(6.3.1) Our approach is to use Robust Principal Component Analysis (RPCA). Given a sequence of greyscale images, we stack each frame into a vector of pixel intensity values, and concatenate each of these vectors into a matrix $X \in \mathbb{R}^{m \times n}$. RPCA looks to perform the matrix decomposition

$$X = L + S \tag{16}$$

where L is low-rank and S is sparse, which correspond to the background and foreground, respectively. The intuition behind this approach is that objects in the foreground only take up a small region, leaving most of the scene unchanged, and thus foreground vectors (columns of S) are sparse. The background is allowed to change over time, but only in a small number of ways, such as the shadows sweeping across the scene throughout the day. This corresponds to the allowed backgrounds (columns of L) spanning a low dimensional subspace of possible images, and thus the matrix of backgrounds is low-rank.

(6.3.2) We use existing MATLAB code `inexact_alm_rpca` [14] to perform the decomposition (16). This code implements the inexact Augmented Lagrange Multiplier method, which is convenient for our decomposition. Indeed, this method is experimentally shown to be the fastest among other commonly used approaches to compute (16). Moreover, an approximated decomposition is fine when dealing with images. In Figure 15 we provide an example of such decomposition.

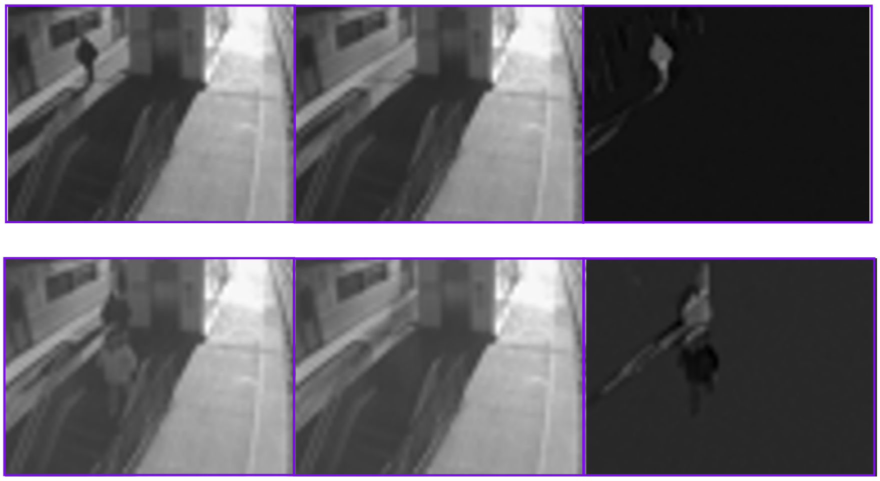

Figure 15: Decomposition by RPCA. We pick two different snapshots of the orginal video, and we display the corresponding decompositions row-wise. In each row: the leftmost frame displays a snapshot of the original video (rearranged column of X); the middle frame displays a snapshot of the background (rearranged column of L); the rightmost frame displays a snapshot of the foreground (rearranged column of S).

(6.3.3) RPCA is mathematically formulated as the following convex optimization problem:

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1, \quad \text{subject to} \quad X = L + S, \tag{17}$$

(6.3.4) where $\|\cdot\|_*$ denotes the nuclear norm of a matrix (i.e., the sum of its singular values), $\|\cdot\|_1$ denotes the sum of the absolute values of matrix entries, and λ is a positive weighting parameter. In our experiments, we take

$$\lambda = (\max(m, n))^{-\frac{1}{2}}\,. \tag{18}$$

(6.3.5) After the decomposition has been performed, it becomes easy to track the foreground. Moving objects (e.g., human, trains), can be detected by looking at the non-zero elements of S. Smoothing over each frame with a 2-D Gaussian filter (in order to remove noise) and employing edge-detection (e.g., the Watershed Algorithm [8]) allows foreground objects to be highlighted in the original video.

6.3.2 Link to PCA

(6.3.6) RPCA is related to Principal Component Analysis (PCA) except with different assumptions on the structure of the noise. It is common to perform PCA on some data matrix and say the top few principal components correspond to signal, whilst the remainder correspond to noise. Thus we are performing the matrix decomposition

$$X = L + N \tag{19}$$

where L is the 'signal' part of our data and is low-rank due to being a linear combination of a small number of principal components. N is a dense matrix of low magnitude since it is made up of only the 'small' directions of variation in the data. Comparing to RPCA, PCA assumes X to be a combination of a background with dense, low-magnitude corruption, whilst RPCA assumes the corruption to be sparse and of arbitrary magnitude. The RPCA model better matches surveillance video, since what we actually observe is an almost constant background with spatially localised foreground elements which cause large magnitude, sparse perturbations to the background.

6.3.3 Drawbacks

(6.3.7) Performing RPCA is computationally expensive, since it requires the computation of the SVD of the data matrix. For the test data made available by Aralia Systems Ltd, the dimension of the data matrix is approximately 300000 x 5000. We managed to run preliminary tests after reducing the size of each frame image, and discarding every other video frame.

(6.3.8) When some objects are static, i.e., when they become still for some time, they are incorporated into the background. For instance, this is the case for left luggage, or people standing still (see Figure 16). In order to avoid such a shortcoming, some 'control' procedures can be implemented. For instance, one can try to detect standing people as they move from being part of the foreground to being incorporated in the background. This may be done by evaluating differences between consecutive frames. Alternatively, we expect that using a much longer video would fix this issue. For instance, in the two-minute video sample provided, the algorithm classifies a static bag as part of the background after $\simeq$ 30 seconds. Given a video over a longer period (possibly at a reduced frame rate) such objects would remain part of the foreground whilst the background captures slower timescale changes such as the weather.

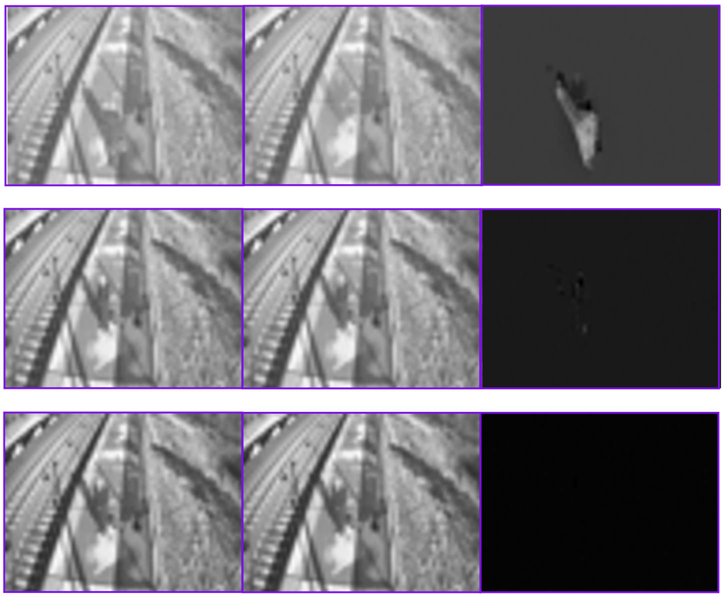

Figure 16: Snapshots displaying a potential failure of the RPCA approach. The layout is the same as Figure 15, i.e., in each row we display different snapshots of the original video and, from the left, the corresponding decompositions. From the top, we display a still person detected as foreground at first, but then progressively incorporated into the background.

6.3.4 Future work

(6.3.9) One can use machine learning techniques [15], such as convolutional neural nets, in order to identify the moving objects once they have been separated from the background. One approach may be to use pre-trained models from the Caffe framework [16], saving the expensive data collection and training process. However, for tracking people across video frames, a lighter weight approach may be more appropriate.

(6.3.10) The method of Convolutional Neural Network (CNN) is one of the methods of Deep Learning. An example of its application was in 2012 when Andrew Ng, along with Google, constructed an experiment by building a neural network with 16,000 processors and learned for three days from images in Youtube [2]. This experiment demonstrated the usefulness of this approach in the image classification. The work [2], [3], [5], [6], [9] demonstrated that this method can successfully be used for classification of static images. Further, [4] and [9] show the high utility CNN in recognition and classification cameras. It is therefore strongly recommended to use different Deep Learning algorithms to solve the problem. Unfortunately, due to limited computational resources at the time, this was not implemented. However, using the basic algorithms for image classification were at 80% - half of the learning completed (with each new epoch of learning the error was reduced by about 2-3%). The idea was to use Deep Learning methods for segmentation, then each of the segments pass by CNN or DBN or another algorithm that has been classified.

(6.3.11) Computationally more efficient methods can be devised by avoiding to compute the SVD of the data matrix. For instance, Krylov methods may be employed to approximate the SVD [10]. However, this approach is not immediate and requires a deep theoretical study.

(6.3.12) It should also be underlined that the RPCA procedure is a post-processing procedure performed on the video. An ultimate goal would be to devise an online procedure, which is able to process the video as soon as a new frame is added. Adopting an approach based on the computation of the SVD of the data matrix, this may be achieved by using special formulas for the SVD update when one column is added to the original matrix.

(6.3.13) The RPCA decomposition of X into L and S is theoretically exact. However, the video frames are corrupted by noise which we do not want to incorporate into either the background or foreground. Thus it would be preferable to instead perform the decomposition

$$X = L + S + N, \tag{20}$$

where L and S are as described above, and N is a dense matrix of low magnitude noise. This decomposition may be performed with Stable Principle

Component Pursuit (SPCP), and should result in a sparser S, enabling easier object identification.

7 Conclusion

(7.0.1) During this study group, we have examined the method of Bayesian inference to scene content, methods used for shadow detection and removal, shape recognition and background separation. We introduced the Bayesian method as an approach to keep track of what is being learned of the background and foreground, so that stationary objects in the foreground do not dissolve into the background. Within the Bayesian framework proposed, we devised several methods to update the background, which was integrated within the algorithm.

(7.0.2) The numerical experiments showed that this method works fairly well, although it detects movements in the shadows of stationary objects throughout the day. This problem may be mitigated by applying shadow removal techniques. Such techniques were explored in the study group, and preliminary results were encouraging, although there is much room for further explorations.

(7.0.3) In terms of updating the background was concerned, we also explored several methods for decomposing an image into its foreground and background, centered around the Principal Component Analysis. Issues surrounding changing slowly varying lighting conditions were addressed, and numerical experiments reported encouraging results.

(7.0.4) Although the ultimate goal is to apply these methods to 3-D data, most of the numerical experiments were performed on 2-D data. This is because it was important for us to first demonstrate these principles and gain a better understanding of them by working within in a simpler 2-D setting (perhaps with the exception of shadow detection and removal), before moving on to applications on more complicated data.

(7.0.5) There were several common themes in the approaches that were taken. Within the sections of image segmentation and background separation, the issue of choosing an initial background was a recurring problem. Since every scene in different settings has a different background, it is therefore not possible to pick a suitable background for any given scene of moving images. Another issue that plagues the issue of image segmentation is that of shadows. Since shadows move according to changes in the lighting conditions, it may be necessary to remove shadows of a background, but not necessarily to remove shadows of a foreground, as it may be the case that shadows of foreground objects need not be removed as they may contain significant information concerning a scene.

(7.0.6) Both the challenges of choosing a suitable starting background and moving shadows of the background may be addressed when working within a Bayesian framework, as it provides a systematic way of learning the changes in the background, illumination and foreground whilst keeping track of assumptions we have made when solving the problem. In the study group, we considered a simple case of foreground and background separation, but further extensions can be made to track shadows, sudden changes or to update the background as we step through the frames.

(7.0.7) The background is determined by averaging out the moving objects that appear in the short term in the scene. The algorithm computes first-order changes in subsequent images. A further correction to this is made for the case of the changing illumination during the day. As the intensity of light changes slowly, a threshold measure of the first-order change is introduced, above which the change is considered significant. This threshold ensures that we neglect the slow and small changes in illumination.

(7.0.8) Once the foreground and background can be identified, we may then be able to apply shape recognition techniques to perform object classification. However, due to time constraints, no significant numerical simulations were performed during the study group, and this is subject for future work.

(7.0.9) All in all, the subject of image segmentation and scene content in moving images continues to be an exciting and challenging area of research.

References

(7.0.10)

[1] Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems.* Springer-Verlag, 2005

[2] Andrew Y. Ng, Jeff Dean, Greg S. Corrado, Kai Chen, Matthieu Devin, Rajat Monga, Marc'Aurelio Ranzato. *Building High-level Features Using Large Scale Unsupervised Learning.*

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet classification with deep convolutional neural networks.* In Proc. NIPS , pages 10971105, Lake Tahoe, Nevada, USA, 2012

[4] C. Farabet, C. Couprie, L. Najman, Y. LeCun. *Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers.* USA, 2012

[5] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors.*

[6] R. Girshick, J. Donahue, T. Darrell, J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5).*

[7] B. Huval, A. Coates, A. Ng. *Deep learning for class-generic object detection.* Hosted at `http://perception.csl.illinois.edu/matrix-rank/sample_code.html`.

[8] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, Vol. 38 (1994), pp. 113–125.

[9] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici. *Beyond Short Snippets: Deep Networks for Video Classification.* 2015

[10] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* SIAM, 2011.

[11] K. Deb, A. H. Suny. *Shadow Detection and Removal Based on YCbCr colour Space.* Smart Computing Review, vol. 4, no. 1, February 2014

[12] D. Tolhurst, C. Ripamonti, A. Parraga, P. Lovell, T. Troscianko. *A multiresolution colour model for visual difference prediction.* Association for Computing Machinery, Inc., 2005

[13] E. Vazquez, R. Baldrich, J. van de Weijer, M. Vanrell. *Describing Reflectances for Colour Segmentation Robust to Shadows, Highlights and Textures.* Journal of Latex Class Files, vol. 6, no. 1, January 2007

[14] L. Zhouchen, L. Risheng, and S. Zhixun. *Linearized Alternating Direction Method with Adaptive Penalty for Low Rank Representation.* NIPS, 2011.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer-Verlag New York, Inc., 2006.

[16] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor. *Caffe: Convolutional Architecture for Fast Feature Embedding* arXiv preprint arXiv:1408.5093, 2014.