BACKGROUND IMAGE SUBTRACTION USING MULTI-VIEW GEOMETRY FOR WIDE BASELINE IMAGE

A Master Thesis Project

Submitted by

Rajat Anantharam

(3197093)

Supervisors

dr. Remco Veltkamp, dr. Nico van der Aa

Game and Media Technology

Department of information and Computing Sciences

Utrecht University – The Netherlands

Abstract

Background subtraction has found its application for tracking and 3D reconstruction. The challenge remains in identifying a foreground object from a given set of 2D images. Such a framework needs to take into account shadows, occlusion, intensity variations and movements in the background. This project proposes one such background subtraction method which makes use of multi-view geometry. While such methods exist till date for stereo images, little research has been done for wide-baseline images. A novel approach to performing background subtraction across such wide-baseline images is proposed. For the purposes of background subtraction a dense pixel-to-pixel correspondence map is generated across three wide-baseline images. Using this correspondence data, a background subtraction method, which is invariant to shadows and intensity variation is proposed. The method is tested over wide-baseline example set with single and multiple-persons as foreground objects. A thorough evaluation of the results is done using epipolar geometry. Promising results are obtained even at 20 frames per second. The obtained results are compared with the results using the "Kernel Density Estimation" method and the conventional frame differencing to test its quality. Several extensions to the approach are proposed to further improve the quality of the foreground images including an extension to N-view images.

Acknowledgements

My sincere thanks to the entire team at Utrecht University working on this project, Nico van der Aa, Robby Tan, Xinghan Luo and Remco Veltkamp. I would like to thank Nico van der Aa for his invaluable suggestions and feedback on the project and continuous support throughout the phase of the project. Without his continued persistence and guidance I would not have managed to finish this project in the stipulated time-frame. Sincere thanks to Remco Veltkamp for providing an opportunity to work in such a challenging topic. Thanks to Robby Tan for his feedback on concepts of computer vision which were used throughout the progress of the project. Very special thanks to Xinghan for providing the calibration data and the input dataset.

Contents

1. Introduction

Background subtraction is a very well-known area of research in the field of computer vision. It has extensive use for the purpose of tracking, posture and gesture recognition. Simply stated, background subtraction is the process of classifying each pixel of a given image as a pixel belonging to a foreground object or to the background scene. However, an image is merely a set of pixels with typically one intensity value for grayscale images and three intensity values for color images. To make the classification, a background model is used, which contains information of the pixel intensities belonging to the background scene. At each time step, the current frame is compared to this background model. If the current pixel value is significantly different from the background model, the pixel is classified as a foreground pixel. If there is no significant difference, the pixel is labeled as a background pixel.

There are many algorithms for the purpose of background subtraction such as frame differencing, single Gaussian, mixture of Gaussians and kernel density estimate. All of these algorithms analyze one frame pixel-wise at a given time. The simplest form of a background subtraction model is frame differencing, which requires an image with no foreground images present in the view. In the ideal case, the absolute difference between this image and the current image gives the location of the foreground pixels. Pixel-based background subtraction methods tend to be sensitive to illumination changes, periodic motion of objects on the scene, occlusion, noise, shadow, camera jitter, etc. There is no method yet which promises to be invariant to all cases mentioned above. However, different applications require a different degree of accuracy in the results from a background subtraction method. The results of background subtraction also have a direct effect on the performance of subsequent methods such as tracking. An example is shown in Figure 1, where multiple camera views are used to construct a 3D voxel representation of a person, based on the result of the background subtraction on each view. Since the side-view of the person suffers from bad foreground detection, the 3D voxel labeling shows voxels that are wrongly labeled. In general, the scenes to be analyzed are becoming more and more complex. Hence, there is a need for a more robust background subtraction method in addition to the existing state-of-the-art methods, which perform real-time. Especially, shadows casted by objects that are not present in the background model, image noise and changes in illumination of the scene add further challenges to background subtraction.

In this thesis we restrict ourselves to the case of static backgrounds in order to handle shadows and intensity changes. The chosen approach is to use multiple camera views, because it takes depth information of the pixels into account. By using depth information in addition to the pixel intensity, the method becomes invariant to shadow and illumination changes.

Computing the depth information for each consecutive image pair is time consuming. In (1) a multi-view background subtraction method is presented, which has a time consuming initialization, but performs the actual background subtraction for each new pair of frames real-time. The initialization step involves finding a dense pixel-to-pixel correspondence map, which relates the pixel location in one view to the pixel location in the other view. The idea behind this is that, when we know the pixel-to-pixel relation, we also know the 3D point corresponding to the pixel locations. Since background subtraction requires all pixels in the image, this correspondence map must be dense. Because the background is assumed to be static, these correspondences will only be distorted when a foreground object enters the scene. This is also the reason why the dense pixel-to-pixel correspondence map has to be computed only once.

This multi-view background subtraction method was defined for a small baseline setting of the camera views (cameras separated by small amount) and the initialization of the method requires manual input. The main contribution of our work is that we extend this approach to a wide-baseline setting and investigate methods to obtain the dense pixel-to-pixel correspondence mapping automatically.

For a given pair of views, it is possible to have certain regions which are occluded. However, background subtraction will only be possible for regions which are not occluded in both views. With a higher number of views, the overlapping region decreases the chance that a region will be occluded in all available views. However, the method discussed in this thesis refers to three views at a given time, one primary views and two auxiliary views. If this method is repeated over all possible combinations of two views among the N available views, the results are likely to be more robust. However, the principle behind the method will not change and therefore the limitation to three views is legitimate.

Chapter 2 provides a brief overview of the existing background subtraction methods and the existing wide-baseline matching algorithms.

Chapter 3 is the actual “Requirements” of the project. A detailed overview of what is provided and what needs to be achieved is discussed in this chapter. Also, the algorithm is explained that is chosen and motivated. A detailed framework is specified which explains each of the steps taken for obtaining the final background subtraction.

Chapter 4 is the evaluation of the methods which are used for the background subtraction. The various steps taken in the background subtraction are evaluated using epipolar geometry. Also a comparative study of the existing background subtraction methods is given.

Chapter 5 concludes this document with conclusions and a few notes on what the future work will be.

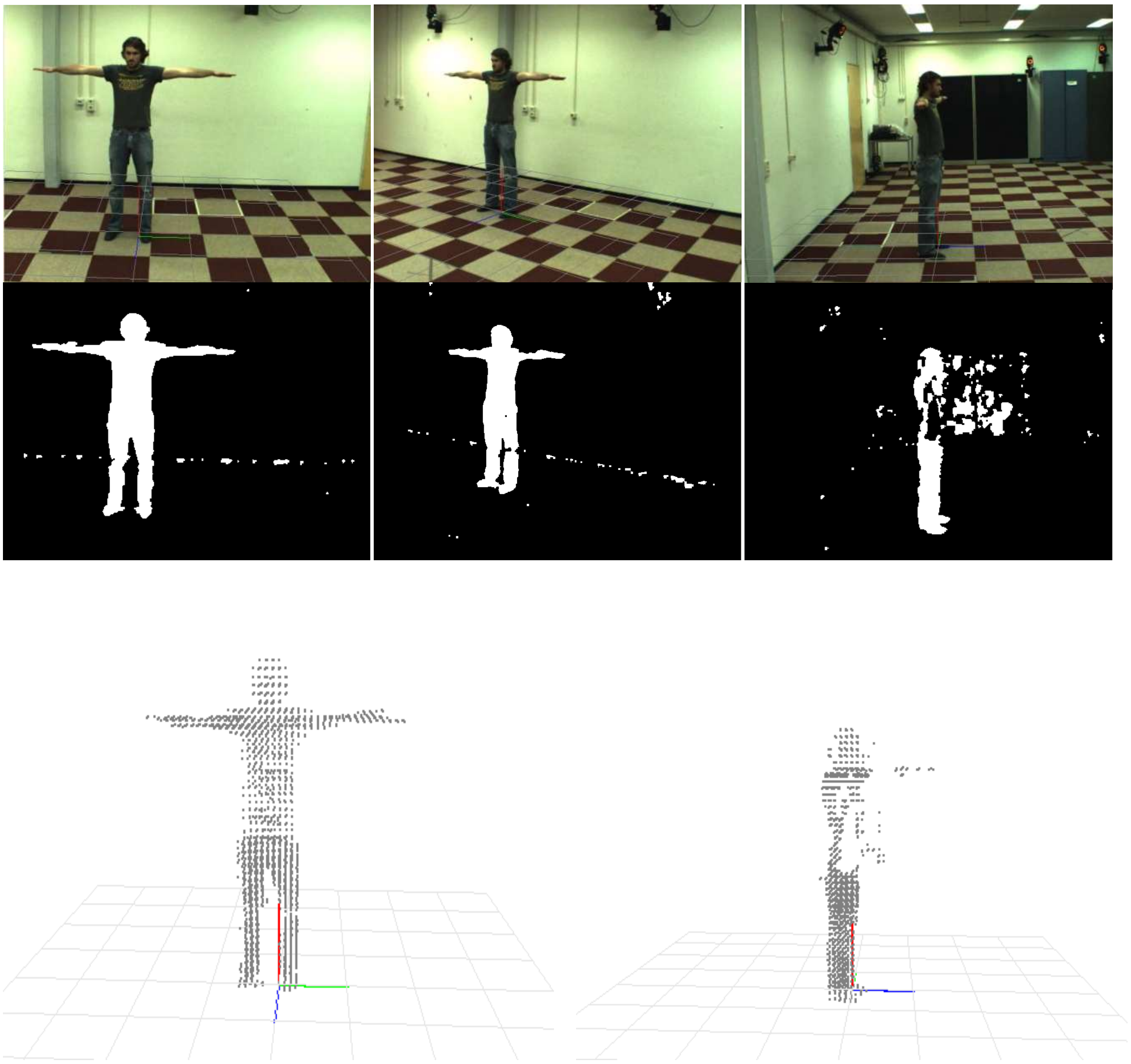


Figure 1: Illustration of a 3D voxel reconstruction using the result of background subtraction on multiple camera views. *Top*: original camera views taken by three cameras, *middle*: result of the background subtraction for each view, and *bottom*: result of the 3D voxel reconstruction.

2. Literature Survey

The main objective of the thesis project is to investigate the possibilities of extending the multi-view background subtraction method presented by (1) to a wide-baseline setting. The multi-view background subtraction algorithm requires a dense pixel-to-pixel correspondence map. Preferably, this mapping needs to be computed automatically by taking two images from two different viewpoints, where these viewpoints are far apart from each other (wide-baseline setting).

This chapter will provide a survey on the current state-of-the-art in background subtraction methods, such that we can compare the results of those methods with our method. Next, another survey is given on the existing algorithms to automatically find a dense pixel-to-pixel correspondence mapping for images in a wide-baseline setting.

2.1 Background Subtraction

Background subtraction is the process of detecting foreground pixels from the background image. Figure 2.1 shows the output of a background subtraction algorithm for an input image. The main idea behind a background subtraction model is to find the difference between the background model and the current image. The methods differ in the way they judge whether a pixel belongs to the background or not. Most background subtraction algorithms are not robust against changing lighting conditions, non-static backgrounds, camera jitter, shadows, etc. In the literature, several background subtraction methods have been developed to cope with these challenges. Adding to these challenges, most applications require a background subtraction model that can perform inreal-time. In the upcoming section we review some traditional background subtraction methods.

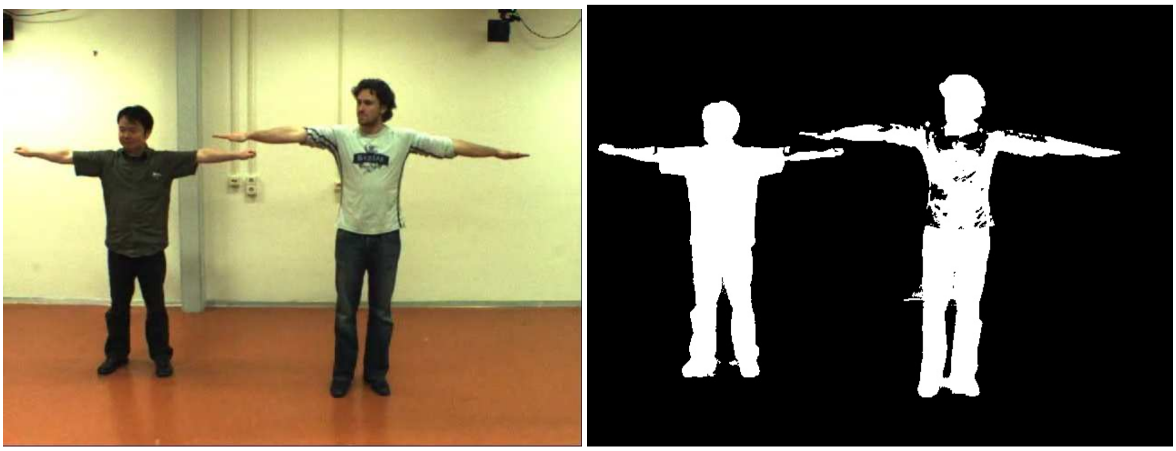


Figure 2: Example output of a background subtraction algorithm. The white pixels indicate foreground and black pixels indicate background.

All the background subtraction methods discussed below operate pixel wise and do not use multiple camera views simultaneously in their analysis, except the method based on multi-view geometry.

2.1.1 Frame Difference

In this method the background model is an image supplied by the user or simply the first image of an image sequence. This image may not show any foreground objects. For each time step the current frame is compared with the given background image. If for a certain pixel the difference in intensity is above a user-defined threshold, the pixel is appointed to be a foreground pixel. Otherwise, it will be labeled as background. Frame difference is the most fundamental background subtraction method available. It is not adaptable to any changes in the background.

2.1.2 Single Gaussian

The intensity distribution of the background at each pixel location is modeled by a single Gaussian with a mean and a variance for every color channel (2). Since it is assumed that a foreground object is only present in the image for a short while, the Gaussian is determined by the intensity of the background pixels. If a pixel intensity of the current frame does not lie within a certain threshold determined by the variance of the Gaussian, the pixel is identified as a foreground pixel. This method assumes a static background, since the intensity distribution of the background is modeled by one Gaussian only. It can handle only gradually changing backgrounds because the distribution parameters are the result of several consecutive frames. However, this method handles noise much better than the straightforward frame difference background subtraction method, since the threshold is determined by the noise level. Despite its shortcomings, the Single Gaussian background subtraction method can be used in practice for a controlled indoor environment.

2.1.3 Mixture of Gaussians

In the Mixture of Gaussians the intensity value of a color channel of a particular pixel is modeled as a mixture of Gaussians (3). By using multiple Gaussians for each pixel, it is possible to handle a periodically moving background like weaving trees. Based on the persistence and the variance of each of the Gaussians of the mixture, it will be determined which Gaussians may correspond to a background color. The method is originally developed to cope with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow-moving objects, and introducing or removing objects from the scene. Hence, it is evident that for static backgrounds the single Gaussian background subtraction model perform similar to the mixture of Gaussians. The latter will only slow down the computation due to its computational complexity.

2.1.4 Kernel Density Estimate

The kernel density estimate method uses kernels to estimate the probability density function (4). For background subtraction these kernels usually are chosen to be Gaussians. For every pixel a kernel function is defined for each of the N previous samples of the intensity value, which is centered at the data points. The main difference between the kernel density estimation method and the single and mixture of Gaussians methods is that the probability density function can take any arbitrary form. This makes the kernel density estimation method more generic.

One major issue that needs to be addressed when using the kernel density estimation technique is the choice of a suitable kernel bandwidth (scale). Theoretically, as the number of samples reaches infinity, the choice of the bandwidth is insignificant and the estimate will approach the actual density. Practically, since only a finite number of samples are used since the computation must be performed real-time, the choice of a suitable bandwidth is essential. A too small bandwidth will lead to a ragged density estimate, while a too wide bandwidth will lead to an over-smoothed density estimate.

As stated, this model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the density function of the pixel intensity distribution. Therefore, the model is able to estimate the probability of any newly observed intensity value. This gives the model some distinct set of features to handle situation where the background

- is cluttered;
- is not completely static, but contains small motions due to moving tree branches and bushes.

This indicates the fact that the model is able to adapt to changes in the scene background, which is one of the main benefits of this approach. The choice of a Gaussian as the kernel function for the computation of the probability density function ensures a fast implementation.

For similar reasons as for the mixture of Gaussians, the kernel density estimate takes too much time to execute without significant improvements in the accuracy of the results in the case the background is static.

2.1.5 Multi-view background subtraction

This background subtraction method takes the depth information of the pixels into consideration. Instead of relying on photometric properties, the information of the scene the cameras look at makes it possible to have a background model invariant to shadows and illumination changes. If the method assumes that the background is static, the background subtraction can be performed real-time, since the pixel-to-pixel correspondences between the camera views will only be distorted when a foreground object is present in the scene.

The difficulty of this method lies in finding this dense pixel-to-pixel correspondence between two given camera views. There are two possible scenarios for the camera setup:

1. The cameras capturing the images only have a small amount of translation and no rotation. At the same time the camera centers only have a small amount of displacement. This case is referred to as *stereo cameras* or *stereo images*.
2. The cameras capturing the images have a large amount of translation and possible a rotation. Also, the cameras are placed at much wider angles, thereby giving two consecutive frames which may have very large occluded regions. This case is referred to as the *wide-baseline setup*.

For the purpose of background subtraction, the online computation of depth and the reconstruction of the 3D model of space at each step can be avoided. If the background is static, stereo disparity between a primary and an auxiliary camera view of the background scene can be determined by specifying the pixel-to-pixel transformation from one image of the background scene to the other. Ideally, this model is only violated by an object that does not belong to any of the background surfaces. For each pixel, the intensity of the corresponding pixel in the other image is checked. If there is a difference in intensity values above a specified threshold, the pixel is marked as foreground. This method is invariant to changes in illumination and casted shadows.

Table 1 provides a summary of all background subtraction methods discussed above. For the features discussed in this table, the multi-view background subtraction method looks most promising.

Method	**Invariant to lighting changes**	**Invariant to noise**	**Invariant to non static background**	**Invariant to shadow**
Frame Differencing	No	No	No	No
Single Gaussian	Yes	Yes	No	No
Mixture of Gaussians	Yes	Yes	Yes	No
Kernel Density Estimation	Yes	Yes	Yes	No
Multi-view Background Subtraction	Yes	Yes	Yes	Yes

Table 1: Feature comparison between background subtraction methods.

2.2 Dense Wide-Baseline Matching

In the initialization step of the multi-view background subtraction method, a dense pixel-to-pixel correspondence map must be computed to relate the pixels of the two input images. In (1) this initial pixel-to-pixel mapping is found manually. The following section discusses some methods to automatically construct a dense pixel-to-pixel correspondence map by using a sparse correspondence map.

Remark: the methods described below assume that this sparse correspondence map is well-distributed across the image. This implies that the images must have textures over the entire image as will be explained in the next chapter. Therefore, generic test scenarios are limited. However, this is one of the findings of the investigation and implementation efforts of this thesis project. Therefore, this survey is given in order to understand the difficulties of generating a dense pixel-to-pixel correspondence map.

All of the methods discussed below, start with a sparse set of pixel-to-pixel correspondences, which can be obtained e.g. by using SIFT (this will be discussed in the next chapter). A sparse set implies that there are some pixel locations in the main image that are linked to pixels in the auxiliary image, but these are only a small subset of the total number of pixel locations.

2.2.1 Adaptive second order intensity moment based method

This algorithm starts from the sparse set of pixel-to-pixel matches to produce a quasi-dense disparity map (not completely dense), which contains a large number of pixel-to-pixel correspondences (5). The main step of the algorithm is the "*match propagation step*" where the quasi-dense correspondence map is generated. Initially, a seed set is defined consisting of the sparse set of pixel-to-pixel correspondences. At each propagation step, small image patches are extracted around the current seed point in both images and the new candidate matches are scored according to the so-called zero mean normalized cross-correlation (ZNCC) (6). The ZNCC is a scoring scheme which does the initial matching of the sparse set of correspondences. These input matches are taken into the seed set and a spatial neighborhood for each of them is found. The pixel with the best ZNCC score is removed from the seed, and new candidate matches are searched from the spatial neighborhood. The process is repeated until a quasi-dense correspondence is obtained.

This method does not generate a dense pixel-to-pixel correspondence map, but only increases the number of point correspondences.

2.2.2 Partial Differential Equation based method

This method uses the concept of inverse depth (7). It uses this approach to compute the 3D coordinates for the sparse set of pixel-to-pixel correspondences. Using the camera calibration parameters, a 3D point is projected to the image plane. A cost function is then defined for a given camera, which takes into account the similarity of the intensity values and the depth value of the matching pixels (extendable to multiple cameras). The value of this function will be close to 1 if the right match is found and close to 0 otherwise. Partial differential equations are used to minimize this cost function.

This method is able to generate a dense pixel-to-pixel correspondence map. However, it requires the knowledge of the camera parameters. This method provides direct extensions for multiple camera views and works well for wide-baseline cases.

2.2.3 Probabilistic approach

Algorithms that find the dense pixel-to-pixel correspondence mapping often have implicit parameters. The probabilistic approach takes those implicit parameters as explicit prior to computing the results.

This method makes a dense depth reconstruction from a small set of wide-baseline images (8). However, as in the case of most wide-baseline images, the matching points across two images need not have the same color values due to non-Lambertian effects or discretization errors. What makes this approach different from the previously discussed approaches is that it takes these inconsistencies into account.

This method regards images as noisy measurements of an underlying “true” image function. The depth estimation is done by including the existing matching coordinates and the camera calibration data. This step is the same as described in the previous method. Extensions are also proposed to handle occlusion in multi-view cases. This method requires the explicit modeling of the light sources and surface properties.

2.2.4 Approach based on Epipolar Geometry

Epipolar geometry states that for a given pixel in one image, its matching pixel lies on the epipolar line in the other image (9). Once the sparse pixel-to-pixel correspondences are available, the pairs of matched image regions are accurately aligned. From these aligned patches, candidate points are chosen for matching. Then the selected points are aligned for the two image regions and the new correspondences are projected back to the new image. To come up with a more dense correspondence, an image region is extracted around each feature at the common maximum resolution. To have a dense correspondence map, it is preferred to align image regions as opposed to pixels. In this approach such image regions are obtained by extracting corner locations. These sub regions are increased progressively by starting from a small region and applying the same operator over the enlarged regions.

Using epipolar geometry over a wide-baseline setting makes this approach more robust. One of the main highlights of this method over the above mentioned methods is that it does not require the knowledge of the camera parameters. Despite the fact that aligned regions are used for the matching, this method still does not ensure a dense pixel-to-pixel correspondence.

2.2.5 Approach based on a quality function

To compute a correspondence between two images in a wide-baseline setting, there are two factors to be taken into consideration. One of them is the depth information of the pixels in the image and the other factor is the intensity value of the pixels. In this method these two parameters are combined to formulate a so-called *quality function,* which is used to compute the dense pixel-to-pixel correspondence map (10). From the initial sparse set of pixel-to-pixel correspondences, a seed set is defined. This is the set of pixel coordinates, which are fed as input for the match propagation step. Each of these matching pixels is taken into the seed set and measured over a quality function. The measure is compared to a threshold value. A good quality match is reported if the value is close to the threshold and the match is removed from the seed. Else the iteration continues with the neighbors of the match. Next, the same procedure is performed on the eight neighboring pixels to the current pixel.

This method ensures that there is a dense pixel-to-pixel correspondence map for two wide–baseline images. It requires the camera calibration data. The quality function makes use of epipolar geometry to make this method robust for different types of data sets.

To conclude this survey, it should be stated that if the requirement of a well-distributed sparse set of pixel-to-pixel correspondences is satisfied, there are more than one method to find the dense pixel-to-pixel correspondence map, namely the partial differential equation based method, the probabilistic approach and the quality function approach.

3. Requirements

This section discusses the requirements of the thesis project. The use of multi-view geometry for background subtraction is motivated and the main assumptions made by the project are given.

The main objective of the thesis project is to perform real-time background subtraction using two or more camera views. The idea behind this method is based on the method proposed by (1). The use of multi-view geometry for background subtraction has a few advantages over traditional single view background subtraction methods. The most important one is its invariance to illumination changes and shadows due to the computation of the depth information of each pixel on the image.

One of the basic assumptions made in this method is that the background is static. If the background is static, stereo disparity between primary and auxiliary camera views of the background scene is also static fully specifying the pixel-to-pixel transformation from one image of the background scene to another. Ideally, this model is only violated by an object that does not belong to the background. Multi-view geometry assumes the presence of multiple cameras. However, in our description of the method, we will assume the presence of only two cameras. A generalization to N camera views is possible in a straightforward way. Another assumption that is made in this thesis is that the surface reflectance properties are Lambertian, which means that light falling on such a surface is scattered such that the apparent brightness of the surface to an observer is identical regardless of the observer's angle of view. Figure 3 illustrates the idea of Lambertian reflectance exhibited on a surface.

As a consequence of the static background assumption, the multi-view background subtraction method proposed in (1) has mainly two phases. The first is the initialization phase, where the dense pixel-to-pixel correspondence map is generated to link the pixel locations in the primary image to the pixel locations in the auxiliary image belonging to the same background scene point in the real world. If the background would not be static, this correspondence map has to be generated at each time step again. The next phase is that this correspondence map is used in the actual multi-view background subtraction step.

The main contributions of this thesis project to the implementation of the multi-view background subtraction method is that we extend this method to a wide-baseline setting and investigate the possibility of automatically computing the dense pixel-to-pixel correspondence during the initialization phase.

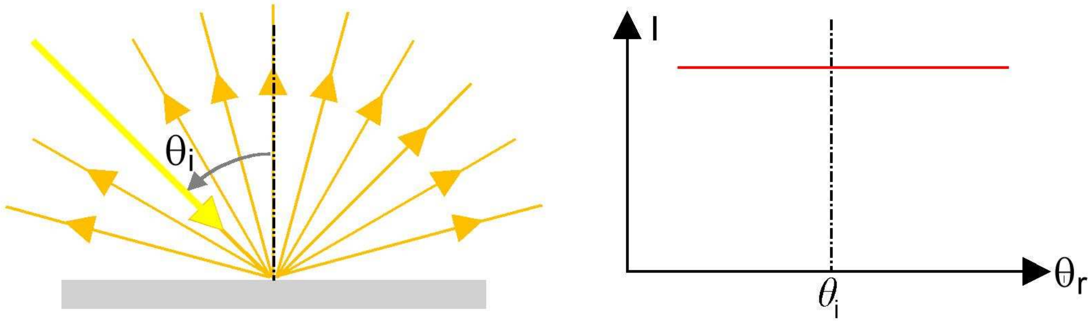


Figure 3: The incident beam of light is completely scattered, the reflected luminance is constant for all angles of inclination.

The prime challenge of this project is the generation of the dense pixel-to-pixel correspondence map from an input set of sparse pixel-to-pixel correspondences. In the case of stereo images, there is usually a small amount of translation across the two reference images and almost no rotation. Hence, obtaining a dense pixel-to-pixel correspondence between the two views is simple because there is not much change in the intensity information. However, in the case of a wide-baseline setup, inconsistency in the intensity information occurs even among matching pixels. Also, there are significant regions on both reference images which are occluded.

When dealing with a primary and an auxiliary image it is not possible to directly obtain a pixel-to-pixel correspondence. However, there are feature extraction algorithms available for extracting significant features of similar properties across two images. The complete initialization step of the multi-view background subtraction method consists of four steps, namely

1. Feature extraction to find a sparse set of points in each image
2. Feature matching to relate the points in one image to points in the other image
3. Find a dense pixel-to-pixel correspondence map from the sparse correspondence map
4. Use epipolar geometry for the evaluation of the accuracy of the matches

The four steps belonging to the initialization step of the method are performed only once. After finding the dense pixel-to-pixel correspondence map, the actual background subtraction can be performed for each incoming image pair at a new time step. The background subtraction step uses the outcome of these four steps for each new pair of images.

In the first step of the initialization phase, the feature extraction is done using SIFT (Scale Invariant Feature Descriptor) (11). SIFT transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. As an alternative, we also investigated MSER (Maximally Stable Extremal Regions) (12). The MSER on the other hand extracts features by grouping pixels with similar intensity values.

The amount and distribution of the feature points obtained by either SIFT or MSER depends on how well the background is textured. For a homogeneous colored plane, both methods will not give any point. Hence, we will use another technique using the homography matrix between planes as an alternative.

To compute the dense pixel-to-pixel correspondence map given the initial sparse pixel-to-pixel correspondences, we consider only one method from the literature survey of the previous chapter. The method uses a match quality function, which takes into consideration the light invariant properties of a pixel and the depth information in each pixel. Iteratively the depth and intensity values of each eight neighbors of a matching pixel are measured by using a quality function. The final value determines the quality of the match for a pair of pixels across two views. Epipolar geometry is used to validate this process.

After the initialization phase, we have the actual background subtraction phase. The main steps in the background subtraction method are as follows:

For each pixel in the primary image:

1. Use the dense pixel-to-pixel correspondence map to find the pixel in the auxiliary image, which corresponds to the current pixel.
2. If the two pixels have the same color and luminosity, label the primary image pixel as background. If the pixels have different color or luminosity, then the pixel in the primary image either belongs to a foreground object or to an "occlusion shadow": a region of the primary image which is not seen in the auxiliary camera view due to the presence of the actual object.
3. If multiple cameras are available, verify the potential object pixels by warping to each of the other auxiliary images and looking for background matches.

Section 3.1 discusses the method used for obtaining the initial key features in the image, such as SIFT and MSER. Section 3.2 gives the details of the matching algorithm used to match the initial sparse pixel-to-pixel correspondences. Section 3.3 explains the creation of the dense correspondence map from the sparse map. Section 3.4 explains the fundamentals of epipolar geometry and also details how it is used for the evaluation of the dense correspondence map in our application. The final section, Section 3.5 explains the actual background subtraction step that happens in the application.

3.1 Key point Detection

This is the first step in our algorithm prior to the computation of the dense pixel-to-pixel correspondence map. The background subtraction needs to be invariant to affine transformations across images. Hence, it is necessary to capture key features in the input images, which are unique and necessarily invariant to affine transformations, illumination, etc. Two such methods are discussed in detail below.

3.1.1 SIFT (Scale Invariant Feature Descriptor)

The SIFT features are key features which are distinct for a given input image (11). In this approach the input image is transformed into a large collection of local feature vectors. These feature vectors are local features, which identify features on the image that are unique for the given image. These features are invariant to image translation, scaling, and rotation and partially invariant to illumination changes. This means that the feature extracted from a given image would remain distinctive across another image which is a scaled/rotated/translated version of the previous image. This invariance property is also associated with invariance to changes in lighting. The SIFT features are extracted over the following stages (or steps):

1. Looking for scale space invariant regions

The SIFT features, which are to be extracted from the image, are required to be invariant to image transformations such as rotation, translation and scaling. Hence, it is necessary to initially identify these regions in the image. In (13) it has been shown that under some rather general assumptions on scale invariance, the Gaussian kernel and its derivatives are the only possible smoothing kernels for scale space analysis. The scale space derivatives at any specified scale can be computed either by differentiating the scale space directly or by convolving the original image with the Gaussian derivative kernels. These derivatives are also referred to as Gaussian derivatives. Figure 4 shows the Gaussian kernels and its derivatives up to order three in 2D images.

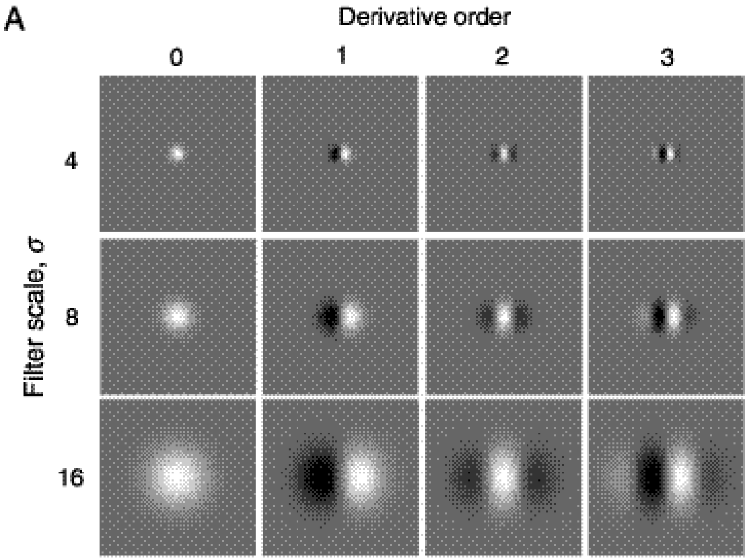


Figure 4: Receptive fields (RFs) of Gaussian derivative spatial filters up to order 3, at several scales. The sign (polarity) of RF has been inverted for orders 2 and 3.

The Gaussian kernels in the 1-D and 2-D cases are given by

$$G_{1\mathrm{D}}(\mathrm{x};\ \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{\mathrm{x}^2}{2\sigma^2}}, \qquad G_{2\mathrm{D}}(\mathrm{x},\mathrm{y};\ \sigma) = \frac{1}{2\pi\sigma^2}\mathrm{e}^{-\frac{\mathrm{x}^2+\mathrm{y}^2}{2\sigma^2}},$$

Where, σ represents the width of the Gaussian kernel.

To achieve affine invariance and a high level of efficiency, key locations are selected at maxima and minima of a difference of Gaussian function applied in scale space. An example of this is illustrated in Figure 5. This is computed by building an image pyramid with re-sampling between each level. Furthermore, it locates key points at regions and scales of high variation, making these locations particularly stable for characterizing the image. Maxima and minima of this scale-space function are determined by comparing each pixel in the pyramid to its neighbors. First, a pixel is compared to its 8 neighbors at the same level of the pyramid. If it is a maxima or minima at this level, then the closest pixel location is calculated at the next lowest level of the pyramid, taking account of the 1.5 times re-sampling. If the pixel remains higher (or lower) than this closest pixel and its 8 neighbors, then the test is repeated for the level above. An illustration of the Gaussian pyramid with neighboring images separated by a constant scaling factor is shown in Figure 6.

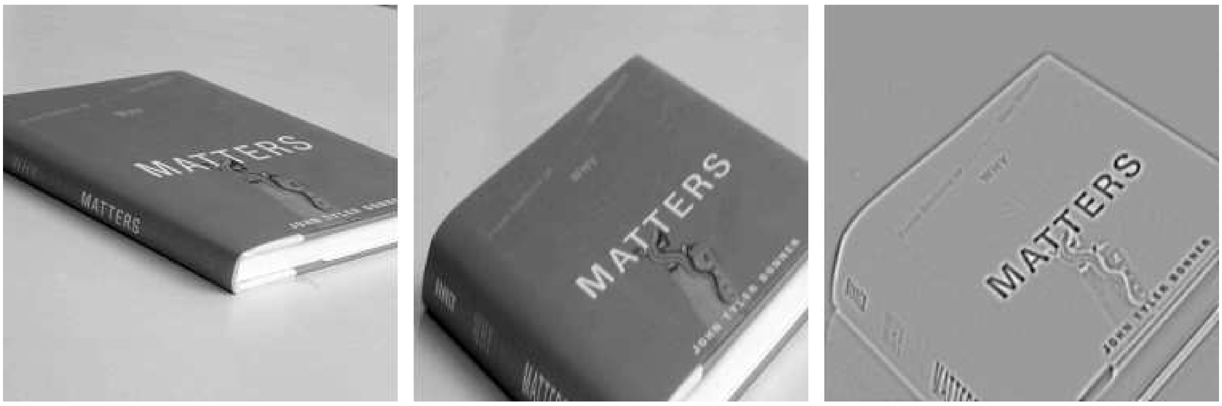


Figure 5: Affine normalization by shape adaption in affine scale-space (left) image of a book cover with an oblique view (middle) result of affine normalization (right) an example of computing differential geometric descriptors.

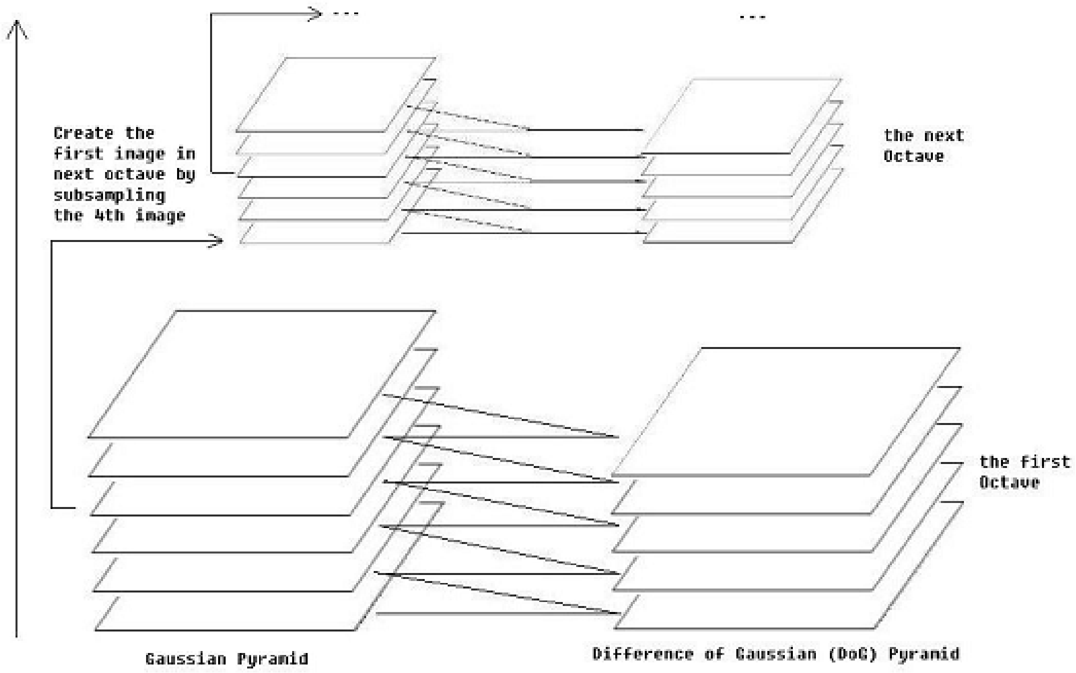

Create the first image in next octave by subsampling the 4th image
the next Octave
the first Octave
Gaussian Pyramid
Difference of Gaussian (DoG) Pyramid
...
...

Figure 6: Bottom: On the left is the Gaussian pyramid, with the neighboring images separated by a constant scaling factor. These are subtracted to give the DoG pyramid on the right. Top: The Gaussian with σ twice that of the original is sub-sampled and used to construct the next octave.

The test is repeated for the level above. Since most pixels will be eliminated within a few comparisons, the cost of this detection is small and much lower than that of building the pyramid indexing these regions to identify candidate object models.

2. Testing for the stability of the SIFT keys

To characterize the image at each key location, the smoothed image A at each level of the pyramid is processed to extract image gradients and orientations. At each pixel $A_{i,j}$ the image gradient magnitude $M_{i,j}$ is computed using pixel differences:

$$M_{i,j} = \sqrt{\left(A_{i,j} - A_{i+1,j}\right)^2 + \left(A_{i,j} - A_{i,j+1}\right)^2}$$

Robustness to illumination change is enhanced by setting the gradient magnitude as a threshold at a value of 0.1 times the maximum possible gradient value. This reduces the effect of a change in illumination direction for a surface with 3D relief, as an illumination change may result in large changes to gradient magnitude but is likely to have less influence on gradient orientation.

3. Local image description using the SIFT keys

The previous step provides us with a SIFT feature descriptor which is also known as the SIFT key. Each SIFT key has a location, scale and orientation information. However, it is desirable to use these SIFT keys to come up with an image description such that this representation is robust against small shifts in local geometry. This can be obtained by using the same pre-computed gradients and orientations for each level of the pyramid that were used for orientation selection.

Figure 7illustrates the initial features that are extracted from the input images. The primary and auxiliary images are taken from two different viewpoints which are significantly far apart. As can be seen from the images, each key feature has a location and orientation information. Another important notice to be made is that there are not many features detected on the walls of the images (mainly on regions where there is limited texture information).

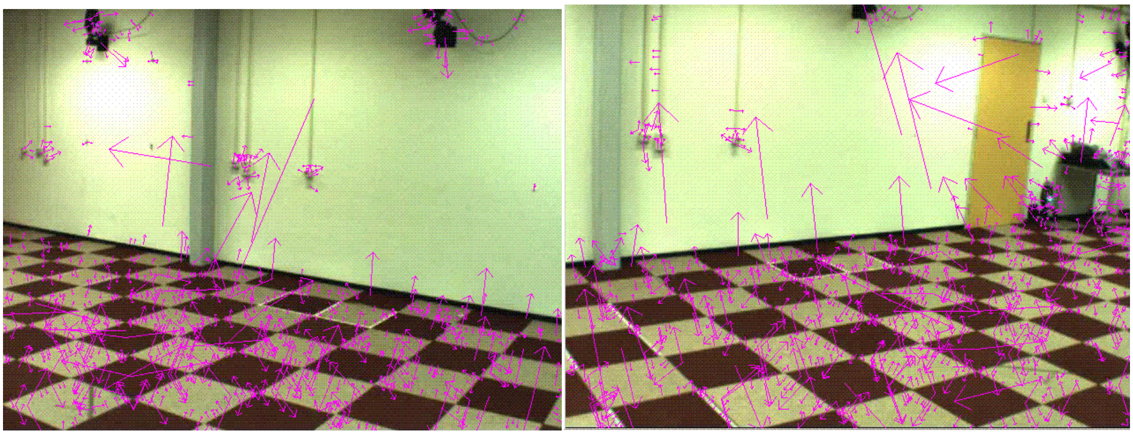


Figure 7: SIFT key features extraction.

The SIFT features are given as output to the proposed algorithm as stated above. Using these features, the matching is done using the matching algorithm proposed in (11). The detailed explanation of the same is provided in Section 3.2 of this chapter.

3.1.2 MSER (Maximally Stable Extremal Regions)

The MSERs are regions formed by groups of pixels having similar intensity values. This concept of extremal regions was introduced in (12). The concept of maximally stable extremal regions can be explained informally as follows: One of the main phases in MSER extraction from a given image is the enumeration phase. In this phase, first, pixels are sorted by intensity. After sorting, pixels are placed in the image (either in decreasing or increasing order) and the list of connected components and their areas is maintained. The process produces a data structure storing the area of each connected component as a function of intensity. A merge of two components is viewed as termination of existence of the smaller component and an insertion of all pixels of the smaller component into the larger one. Finally, intensity levels that are local minima of the rate of change of the area function are selected as thresholds producing maximally stable extremal regions. In the output, each MSER is represented by the position of a local intensity minimum (or maximum) and a threshold.

Figure 8 shows the extremal regions on a test image using MSER algorithm.

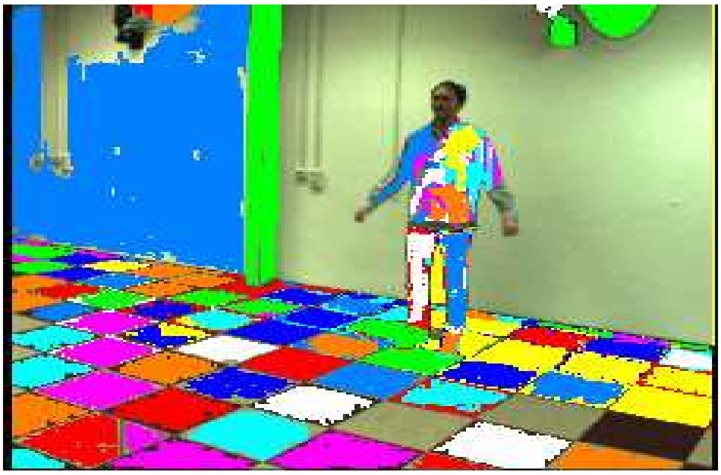


Figure 8: MSER features on the test image. The various colored regions indicate regions on the image which have similar intensity values.

The set of extremal regions are closed under:

1. Continuous (and thus projective) transformation of image coordinates.
2. Monotonic transformation of image intensities.

There are primarily two regions which are defined while computing the MSER. The "Distinguished Region" (DR) is detected which mainly comprises of the MSERs computed on the intensity image (MSER+) and computed on the inverted image (MSER-). The other kind, namely the maximal region (MR) is of arbitrary size which may be associated with each DR, if the construction is affine-covariant. Smaller MRs are both more likely to satisfy the planarity condition and not to cross a discontinuity in depth or orientation. On the other hand, small regions are less discriminative, i.e. they are much less likely to be unique. Increasing the size of a measurement region carries the risk of including parts of the background that are completely different in the two images considered. Clearly, the optimal size of a MR depends on the scene content and it is different for each DR.

For the purposes of background subtraction it is initially required to have features extracted. However, the feature extraction algorithms need to be suitable for the cases that are investigated. Hence, it is important to take into account the requirements set by the project and see if the above mentioned methods meet those requirements. The following table illustrates a comparison between the two methods based upon the requirements set by the research project,

Method	Translation Invariance	Rotation Invariance	Scaling Invariance	Invariance to Changing Illumination	Occlusion Handling	Shadow Detection
SIFT	Yes	Yes	Yes	Partly	No	No
MSER	Yes	Yes	Yes	Yes	No	No

Table 2: Comparison between SIFT and MSER.

From the Table 2 it is evident that the choice of methods is not obvious. Since the background subtraction model needs to take into account changing lighting conditions, MSER seems like a better choice than SIFT. However, apart from that difference, the two feature extraction methods produce similar results. Hence, both SIFT and MSER can be chosen for the initial feature matching prior to background subtraction.

In the following section the matching algorithms for SIFT is explained in detail. Based upon these, initial matches (also known as the putative correspondences) are registered. These putative correspondences are further used for the computation of the fundamental matrix. The fundamental matrix is in turn used to get the right correspondences between the sparse set of feature points.

3.2 Feature Matching

Feature extraction methods are aimed at obtaining scale and possibly affine invariant descriptors for query images. This is vital since the features, which are extracted, need to be unique for a given image and also easily distinguishable. SIFT and MSER enables the extraction of such features from a given image. If the feature extraction algorithm is applied to a pair of images taken from two different viewpoints, then it is also possible to match those features. In the following sections, the matching algorithm used in SIFT is discussed. Our project also makes use of this matching algorithm.

3.2.1 SIFT Feature Matching

To match the SIFT features across two images, (11) uses a modification of the k-d tree algorithm called the Best-Bin-First (BBF) search method that can identify the nearest neighbors with high probability using only a limited amount of computation. The BBF algorithm uses a modified search ordering for the k-d tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This search order requires the use of a heap (data structure) based priority queue for efficient determination of the search order. The best candidate match for each key point is found by identifying its nearest neighbor in the database of key points from training images. The nearest neighbors are defined as the key points with minimum Euclidean distance from the given descriptor vector. The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. To further improve the efficiency of the BBF algorithm, the SIFT key samples generated at the larger scale are given twice the weight of those at the smaller scale. This means that the larger scale is in effect able to filter the most likely neighbors for checking at the smaller scale. This also improves recognition performance by giving more weight to the least-noisy scale.

Figure 9 shows the initial matches which are obtained across the SIFT features that are computed across the primary and auxiliary views.

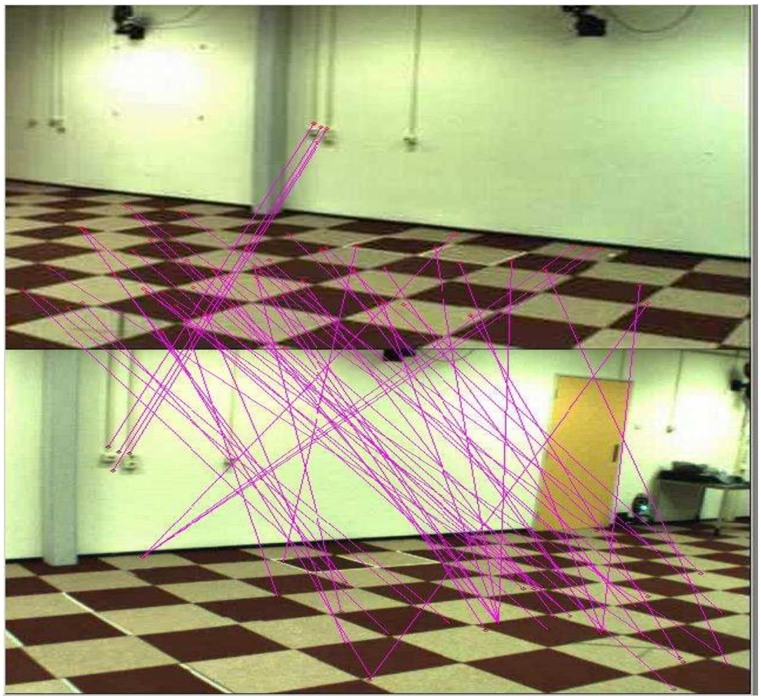


Figure 9: SIFT matches across two views.

3.3 Dense Correspondence Map

To generate a dense pixel-to-pixel correspondence map for two images, it is necessary to have as many features extracted as possible, which are visible in both images. These features (key points) need to be well distributed across the entire image.

Finding the dense pixel-to-pixel correspondences using the sparse set of correspondences as described in the previous section, we have to:

- Compute depth information
- Compute intensity value
- Quality function proposed in (14)

3.3.1. The Dense Point Correspondence Algorithm using SIFT

There are two aspects in computing a correspondence map. The depth information of the pixels in the image needs to be preserved and the intensity value of the pixel needs to be taken into consideration. It is possible that when an image is viewed from two different angles, then two corresponding pixels may have different intensity values. These two aspects are taken into account in (15) for the computation of the dense pixel-to-pixel correspondence map.

In the first step we detect the key features on the image using SIFT. Then we use the matching algorithm proposed by (11) to have the first set of sparse correspondences.

The following steps are taken to generate the dense pixel-to-pixel correspondence map from the initial sparse correspondence map:

- Match Propagation
    - Defining the best pixel matching quality function
        - Clustering based light invariant photo consistency constraint
        - Depth smoothness constraint
- Mismatch Rectification and scene reconstruction

The objective of the match propagation step is to assign for every 2D pixel $\mathbf{x}$ of the input image, a corresponding depth value $D(\mathbf{x})$ and a matching quality value $Q(\mathbf{x})$. The value of $D(\mathbf{x})$ is computed using the IBM algorithm and the value of $Q(\mathbf{x})$ is computed using the following expression:

$$Q(\mathbf{x}) = \max_{d \in [d_{min}(\mathrm{v}), d_{max}(\mathrm{v})]} \{\lambda Q_{photo}(\mathbf{x}, d) + (1 - \lambda) Q_{depth}(\mathbf{x}, d)\}$$

with

Q_{photo}: Pixel matching quality function based on the clustering-based light invariant photo-consistency constraint which deals with occlusions, light changes or image noise between images. This is a quality function that measures how invariant a given pixel is to lighting changes.

Q_{depth} : Pixel matching quality function based on the data driven depth smoothness constraint which handles the depth discontinuities in the scene.

λ: Weighing factor which is a constant.

$[d_{min}(\mathbf{v}), d_{max}(\mathbf{v})]$: Depth searching range.

While generating the dense correspondence map, each pair of the initial matching points is taken as input seed points. Then the quality value of each of the eight neighbors around each of these seed points is computed and compared against a threshold. This step is done iteratively until all the pixels in the image are covered Figure 10 shows one such match propagation round.

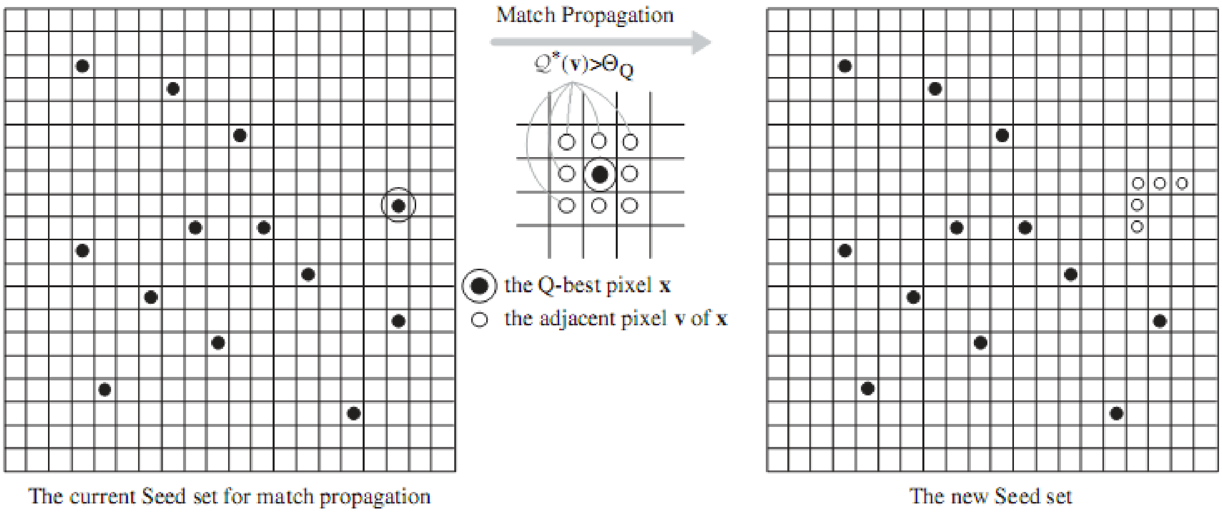

Match Propagation
$Q^*(v)>\Theta_Q$
● the Q-best pixel **x**
○ the adjacent pixel **v** of **x**
The current Seed set for match propagation
The new Seed set

Figure 10: Example match propagation round.

At the end of this match propagation round, this method makes it possible to generate a dense correspondence mapping. The whole process as is evident starts from a sparse correspondence and at the end produces a dense pixel-to-pixel correspondence, which is desired for the background subtraction phase.

3.3.2 Dense Correspondence Map using Homography matrix

Since the test data set in our experiments were videos captured in an indoor setting with a very limited amount of texture, the use of SIFT for the key point feature extraction proved insufficient. The use of SIFT features led to very few feature points along with walls of the image (see Figure 11 and hence it was necessary to have another approach to obtain the dense pixel-to-pixel correspondence. In Figure 11, we see that the SIFT features are only detected in regions on the wall that have some texture. These small patches with a large density of feature points are shown with the red circles. It is also evident that most of the region on the wall does not have any key point features.

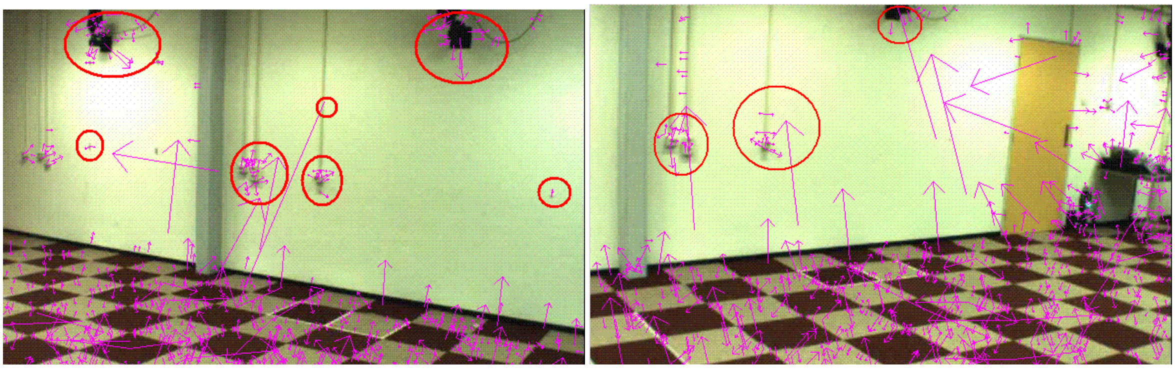


Figure 11: SIFT features extracted from the two input views as shown by the red circles.

An alternative approach to find matching pixels across two images is the use of homography transformations across the two image planes. Before we explain how a homography matrix can be used for the computation of the dense pixel-to-pixel correspondence map, we define the concept of the homography matrix.

In computer vision, we define planar homography as a projective mapping from one plane to another. Thus, the mapping of points on a two-dimensional planar surface to the image of our camera is an example of planar homography. It is possible to express this mapping in terms of matrix multiplication if we use homogeneous coordinates to express both the viewed point Q and the point q on the image to which Q is mapped. If we define

$$Q' = [X\ Y\ 1]^T$$

$$q' = [x\ y\ 1\ ]^T$$

we can express the action of the homography simply as:

$$q' = sHQ'$$

where H is a 3x3 matrix and Q' and q' are 3x1 homogeneous representation of coordinates.

Here, we have introduced the parameter s, which is an arbitrary scale factor (intended to make explicit that the homography is defined only up to that factor). It is conventionally factored out of H, and we will stick with that convention here. The most important observation is that H has two parts: the physical transformation, which essentially locates the object plane we are viewing; and the projection, which introduces the camera intrinsic matrix. This is shown in the Figure 12.

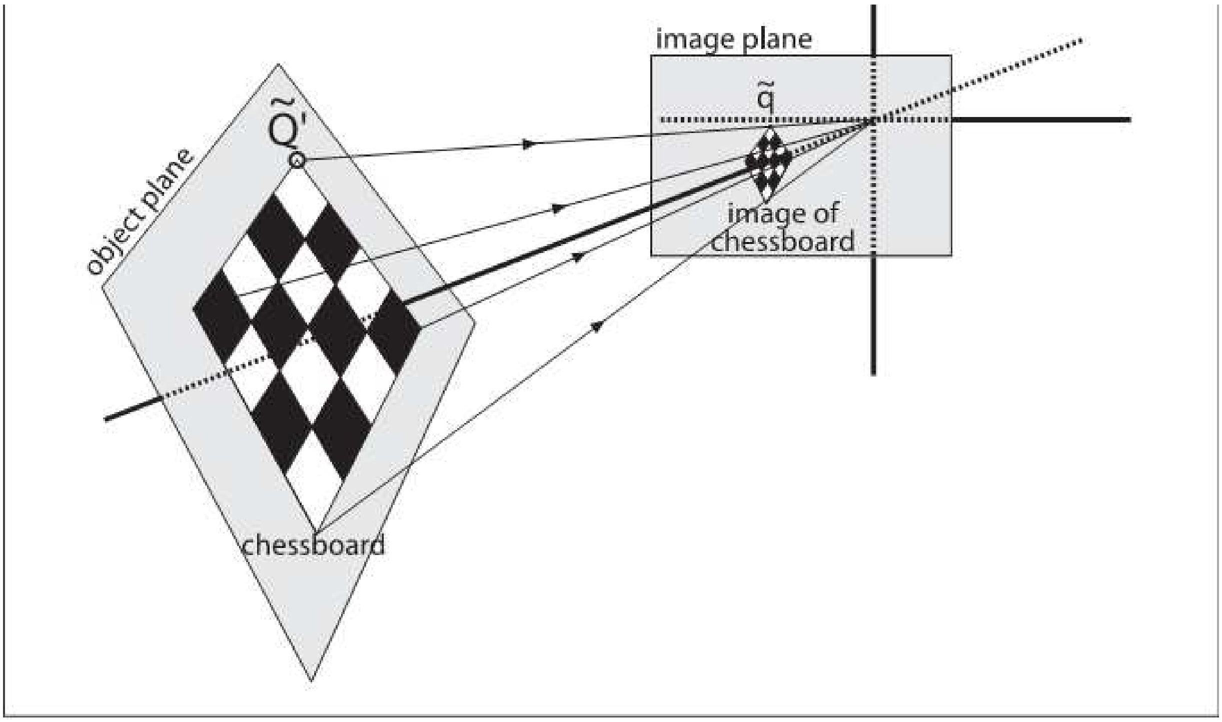


Figure 12: View of planar object as described by homography.

The homography matrix H relates the positions of the points on a source image plane to the points on the destination image plane (usually the imager plane) by the following simple equations:

$$P_{dst} = HP_{src}, \quad P_{src} = H^{-1}P_{dst}$$

$$P_{dst} = \begin{bmatrix} x_{dst} \\ y_{dst} \\ 1 \end{bmatrix}, \quad P_{src} = \begin{bmatrix} x_{src} \\ y_{src} \\ 1 \end{bmatrix}.$$

By defining this homography matrix across the two images, we can obtain a dense pixel-to-pixel correspondence across the image planes.

Figure 13shows the homography regions, which we obtained across two background views. The colored regions indicate the regions in the two images for which pixel-to-pixel correspondence is defined.

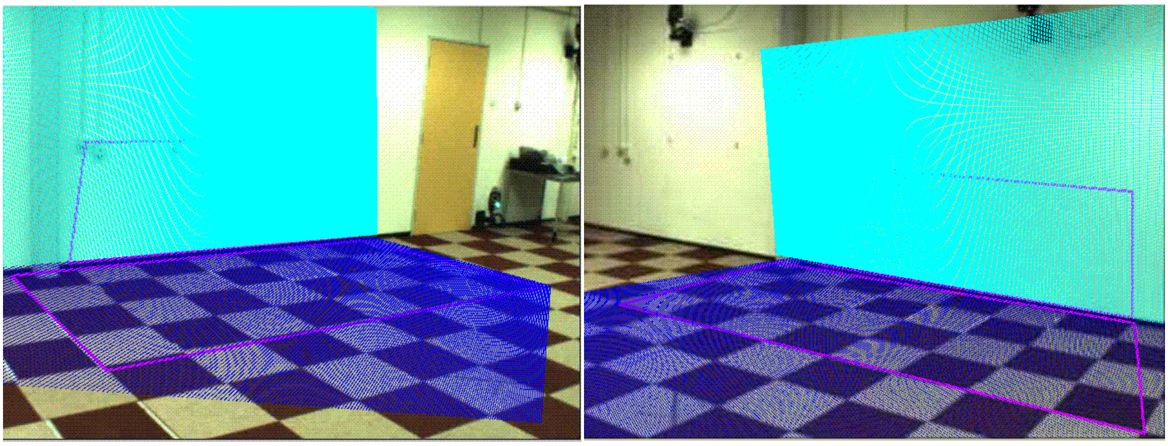


Figure 13: Dense point correspondence using homography. The blue regions in the two images indicate the correspondence regions on the floor across the two views. The regions on the wall are represented by cyan color.

3.4 Epipolar Geometry and Fundamental Matrix

The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the set of planes having the baseline as axis. By convention, the baseline is the line joining the camera centers. This geometry is usually motivated by considering the search for corresponding points in stereo matching. One of the prominent properties of epipolar geometry is the way it relates the coordinates in one image with respect to the other despite the fact that these two images are obtained by cameras which are placed significantly apart (wide-baseline). The construction of the fundamental matrix enables the computation of the epipolar lines and therefore establishes a relation for all the points in the image. This relation can be used as a verification process for the pixel-pixel correspondences which are obtained.

Figure 14 illustrates the point correspondence problem and the motivation for epipolar geometry for solving this point correspondence. Figure 14(a) shows the two camera centers indicated by C and C'. The plane formed by the baseline and the real world point is known as the epipolar plane. The challenge however arises in automatic computation of **x**' (the point on the real world object as seen by camera C') given the point **x**. Figure 14(b) introduces the concept of the epipole and the epipolar line.

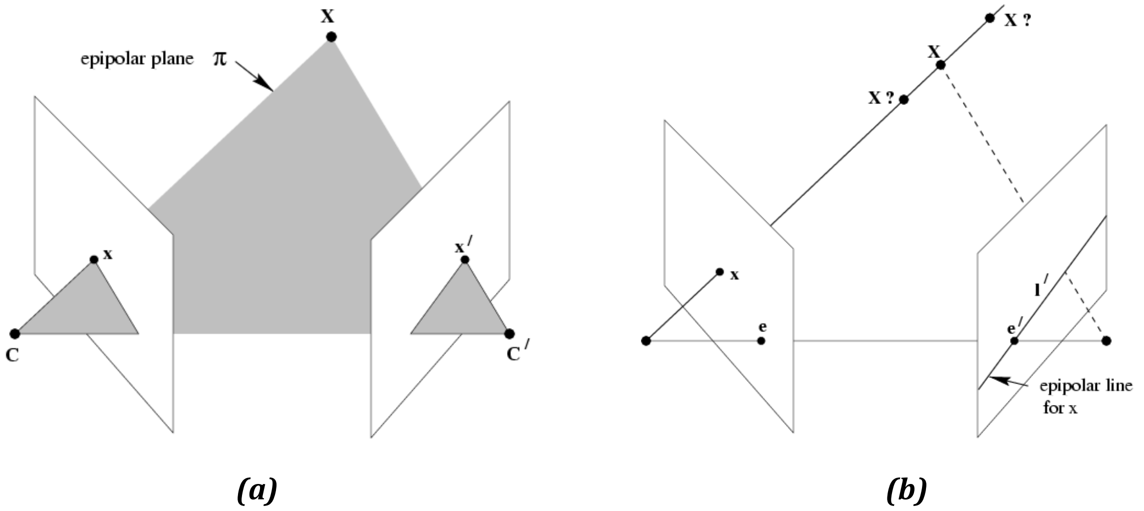


Figure 14: Illustration of epipolar geometry. (a) epipolar plane, (b) epipolar line and the epipole.

The definitions of the geometric entities in epipolar geometry are as follows:

Epipole: The point of intersection of the line joining the camera centers with the image plane. Equivalently, the epipole is the image in one view of the camera center of the other view.

Epipolar plane: The plane formed by the baseline and the 3D world point.

Epipolar line: The intersection of the epipolar plane with the image plane. All epipolar lines intersect at the epipole. An epipolar line intersects both image planes in epipolar lines, and defines the correspondence between the lines.

Fundamental Matrix: The epipolar geometry as defined above forms the basis for the formulation of the fundamental matrix. The fundamental matrix is the algebraic representation of the epipolar geometry. It is a 3x3 matrix of rank 2. If a point in the world **X** is imaged as **x** in first view and **x'** in the second, then the image points satisfy the relation $\mathbf{x}'^{T}\mathbf{F}\mathbf{x} = 0$. The geometric derivation of this fundamental matrix is explained in (16).

Given the value of **x,** what can we say about the point **x'**? The plane π is determined by the baseline and the ray defined by **x**. From above we know that the ray corresponding to the unknown point **x'** lies in π, hence the point **x'** lies on the line l', which is the intersection of of the epipolar plane and the image plane corresponding to camera C'. This line l' is the image in the second view of the line through **x** and **X**. This line is known as the epipolar line corresponding to **x**. In terms of stereo correspondence algorithm, the benefit is that the search for the point corresponding to **x** need not cover the entire image plane but can be restricted to the line l'.

To filter the outliers, epipolar geometry primarily defines a matching criterion for two points in two images. This criterion states that for any pixel in the primary image, its matching coordinate in the auxiliary must lie on the epipolar line. Hence, the overall process requires the construction of epipolar lines for each pixel in the primary image in the auxiliary image.

Note: The use of epipolar lines is used as a verification tool in the thesis project and not a tool to generate the dense correspondence

Figure 15shows the epipolar lines which are constructed over the auxiliary image for all the coordinates in the primary image. The points in the primary image hence have corresponding epipolar lines on the auxiliary image.

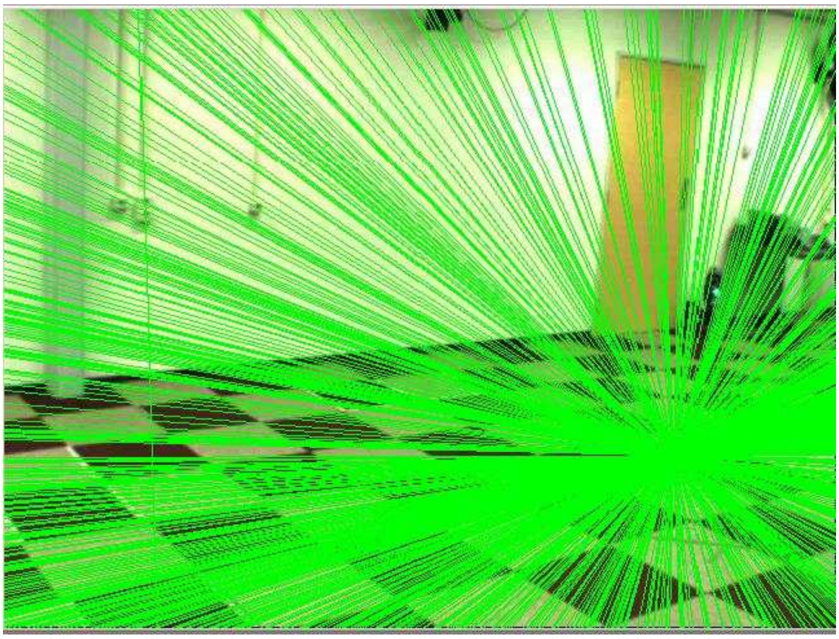


Figure 15: Epipolar line for each point of primary image.

3.4.1 RANSAC

For the computation of the fundamental matrix it is initially required to supply a set of matching points. It is possible that some of these matches are inaccurate meaning they can form outliers. It is necessary to have a robust estimation using the Random Sample Consensus Algorithm (RANSAC) (17). The idea of using RANSAC for removing outliers from a set of possible matches is illustrated below.

Two points are initially chosen randomly; these points define a line. The support for this line is measured by the number of points that lie within a distance threshold. This random selection is repeated a number of times and the line with the most support is deemed the robust fit. The points which are within this threshold distance are the inliers. The outliers do not contribute to the support value of the line.

OpenCV has a direct implementation of RANSAC for the computation of the fundamental matrix. It is required that there are at least eight matching points as the initial matches. The fundamental matrix also enables the construction of the epipolar lines our test images. These epipolar lines are the ideal tools for the necessary validation of the matches in the dense pixel-to-pixel correspondence map.

3.5 Background Subtraction

The next stage of the project involves the actual background subtraction. Our method uses the idea introduced by (1).

3.5.1 Method

The background subtraction method classifies image points belonging to a known static surface, rather than to a group of neighboring points of similar texture. Unfortunately, the direct implementation of this idea requires executing dense stereo algorithms in real-time, which is possible, but requires either massive computational power or specialized hardware.

For the purposes of background subtraction, we can avoid the online computation of depth and the reconstruction of the 3D model of space at each step. If the background is static, stereo disparity between a primary and an auxiliary camera view of the background scene is also static fully specifying the pixel-to-pixel transformation from one image of the background scene to another. Ideally, this model is only violated by an object that does not belong to any of the background surfaces.

The basic background disparity verification algorithm can be described in the following table.

For each pixel in the primary image:

1. Use the disparity warp map to find the pixel in the auxiliary image, which corresponds to the current pixel.
2. If the two pixels have the same color and luminosity, label the primary image pixel as background;
3. If the pixels have different color or luminosity, then
   the pixel in the primary image either belongs to a foreground object,
   or to an "occlusion shadow": a region of the primary image which is not seen in
   the auxiliary camera view due to the presence of the actual object;

4. If multiple cameras are available, verify the potential object pixels by warping to each of the other auxiliary images and looking for background matches.

Because the basis of comparison is the background disparity warp between two images taken at the same time, illumination or, to a certain degree, reflectance can vary without significantly affecting the performance. It should be noted that all that is required for the algorithm to work is a disparity map between the primary image and each of the auxiliary images.

To isolate the foreground image from the background, a binary masking function $f(r)$ for construction of the background image is used. This masking function $f(r)$ determines the complexity and accuracy of the algorithm.

In general, $f(r)$ is a Boolean function which takes a value of 1 for all primary image locations, **r**, which belong to the set of foreground pixels F,

$$f(\mathbf{r}) = \begin{cases} 1, & \forall_{\mathbf{r} \in \mathcal{F}} \\ 0, & \forall_{\mathbf{r} \in \mathcal{F}} \end{cases}$$

Then, the subtracted image is formed simply by applying the mask to the primary camera view:

$$S(r) = f(r)I(r)$$

$S(r)$ is the resulting image with the background pixels removed. In its simplest form the foreground pixel set, F is just a set of pixels not satisfying the disparity constraint. In this case, the function $f(r)$ is determined as follows:

$$f(\mathbf{r}) = \begin{cases} 0, & if\ I(r) = I'(r') \\ 1, & \text{otherwise} \end{cases}$$

In reality we never get measurements good enough to comply with this sort of screening, so we relax this constraint to compensate for possible errors and formulate the comparison to accept a value within some tolerance rangeε:

$$f(\mathbf{r}) = \begin{cases} 0, & if\ |I(r) - I'(r')| < \varepsilon \\ 1, & \text{otherwise} \end{cases}$$

This value of ε is given as a user input such that the results are directly evident on the basis of user input.

It is however unlikely that even the exact pixel-to-pixel matches have the same pixel intensity. Hence, we compute a correction term, such that the pixel intensity difference is normalized before it is compared to the threshold to decide if it is a foreground or background pixel.

As a result of the background subtraction step, the output is a black image with white silhouettes indicating the foreground objects.

3.5.2 Handling Occlusion Shadows

An occlusion shadow is the region of pixels in the primary camera view which is not seen in the auxiliary camera view. If we were to imagine the cameras as light sources themselves, then the occlusion shadow is the same shape as the regular shadow of the foreground object. This can happen in case if the view is restricted to 2-views. Figure 16 shows the occlusion shadows which were observed in our test data.

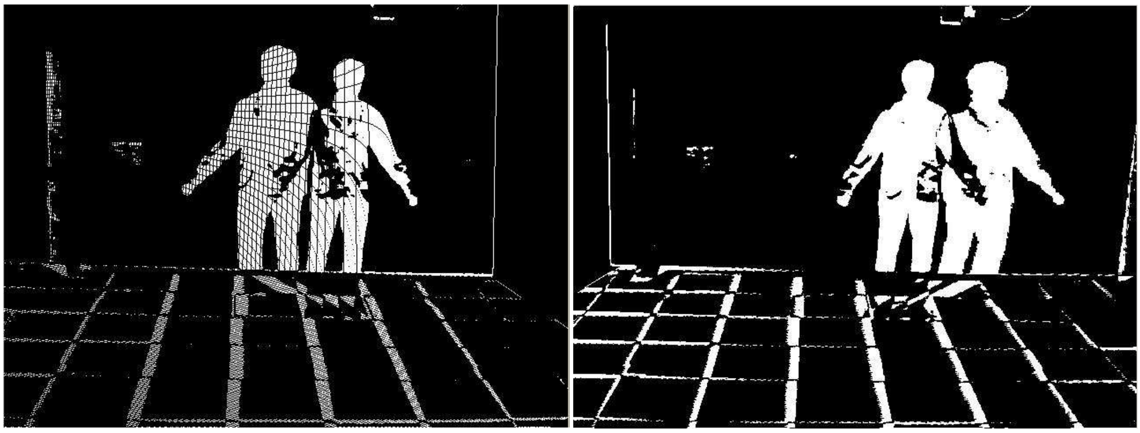


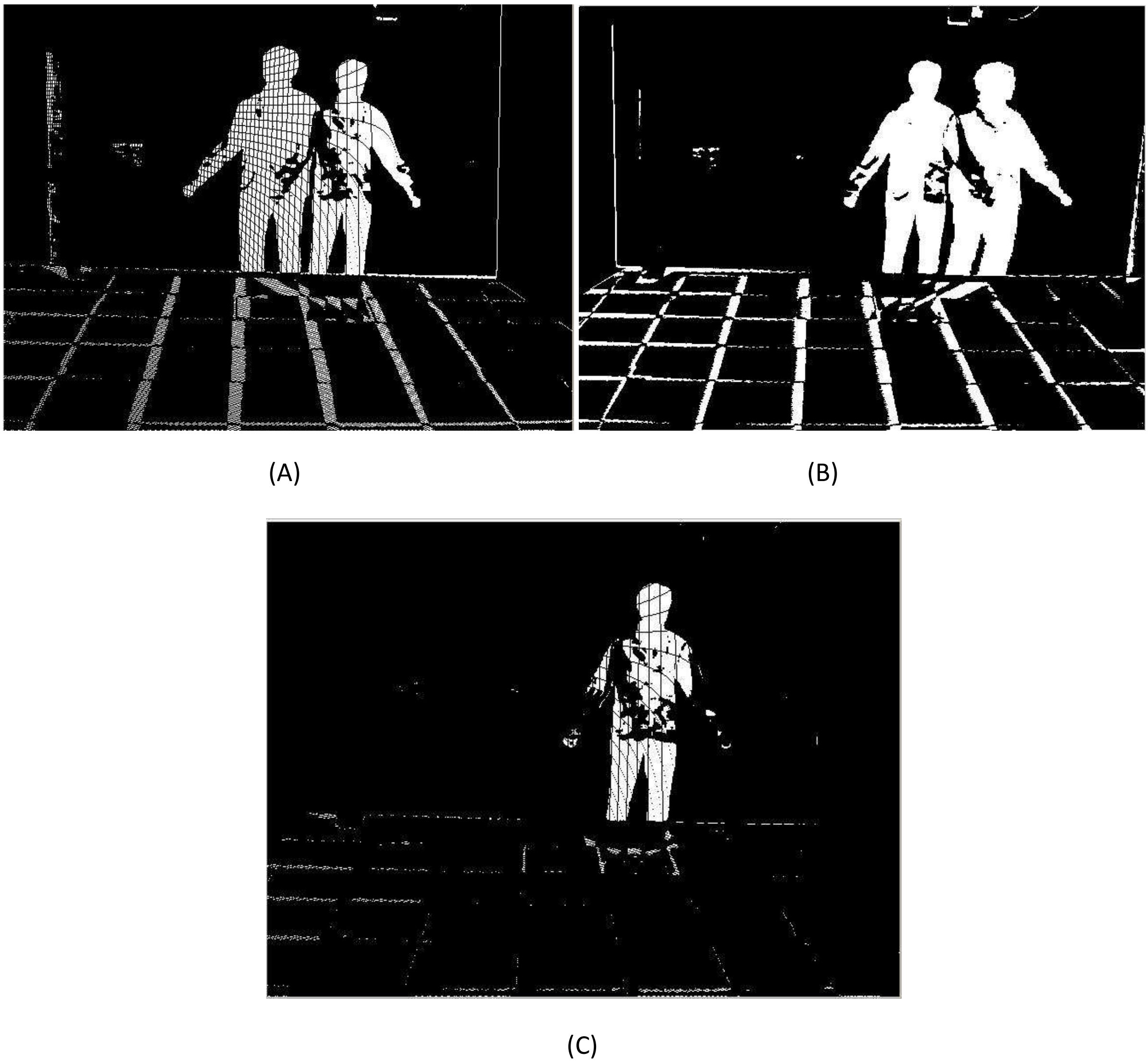
Figure 16: Occlusion shadows observed in the two views of the camera.

One way of handling this, is to have a third camera view and cross verifying each pixel across three camera views. By this approach only the "true" foreground pixel will violate the background disparity in all camera views, while the occlusion shadows will be marked as background. This is the final step of our background subtraction. What we obtain is the foreground object separated from the background without any occlusion shadows.

A function $f(r)$ is defined such that it takes a value of 1 when the pixel at location r belongs to the foreground in every camera pair. This is illustrated as follows:

$$f(r) = \begin{cases} 1, & \forall r \in F = \cap F_l \\ 0, & \forall r \neq F \end{cases}$$

Figure 17 illustrates the idea of occlusion shadow removal.


(A)

(B)

(C)

Figure 17: Occlusion shadow removal in background subtraction step. (A): Shows the occlusion shadow due to the “left-front” view in the front image (B) Shows the occlusion shadow due to the “right-front” view in the front image (C) Shows the occlusion shadow removed from both the views using the AND operator.

4. Evaluation

In this chapter we evaluate the various steps in the multi-view background subtraction method. A qualitative evaluation of each step is done and a comparison is drawn with existing results. The algorithms are tested on a data set, which is described in Section 4.1, the actual evaluation starts with evaluating the steps of the initialization phase, namely

- SIFT
  The density of the number of features is evaluated along with a few observations about the location of the key point features.

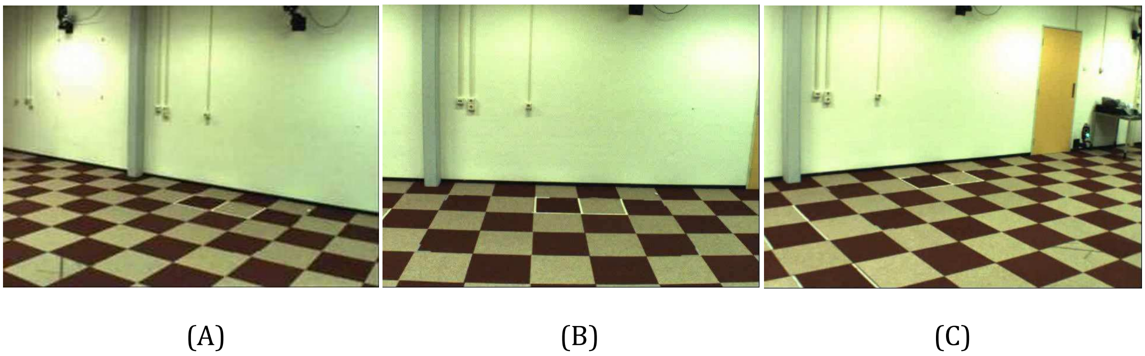
- Matching
  The quality of the matching is evaluated and the parameter to increase or decrease the number of matches is tested.

- Homography
  The accuracy of the pixel-to-pixel correspondences obtained through the homography computation is evaluated using epipolar geometry and manual hand labeling.

The results of the actual background subtraction will be shown and discussed in Section 4.4. The results are compared with the results obtained from kernel density estimate background subtraction. Finally, the run-time complexities are given.

4.1 Description of the data set

For the purposes of background subtraction we chose the Utrecht University MoCap database. This is a set of movies shot with single and multiple persons which are shot from wide baseline cameras. There are five camera views in total and their positions are left, right, middle, left-middle and right-middle with respect to the person's face. We make use of the left-front, right-front and the front views for the purpose of evaluating the background subtraction method. The images have resolution 640x480 and are recorded with 25 fps. The setting is a controlled indoor environment. This data set also has calibration information available of the cameras. This database satisfies the requirements of this thesis set-up and is therefore used in the evaluation. A multi-view example of the background scene of this data base is given in Figure 18.



(A) (B) (C)

Figure 18: Example dataset background images. (A) left-front view (B) front view (C) right-front view

4.2 Evaluation of the sparse pixel-to-pixel correspondence map

We used SIFT to find the initial sparse pixel-to-pixel correspondences between two views. For the test images, we obtained 729 features on the left view and 1013 features on the right view. For computing the matches from the initial SIFT features, we use a threshold parameter. This is known as the nearest neighborhood distance threshold. By reducing the number we have a higher number of matches. Figure 19 shows 128 matches obtained using the threshold value of 0.5.

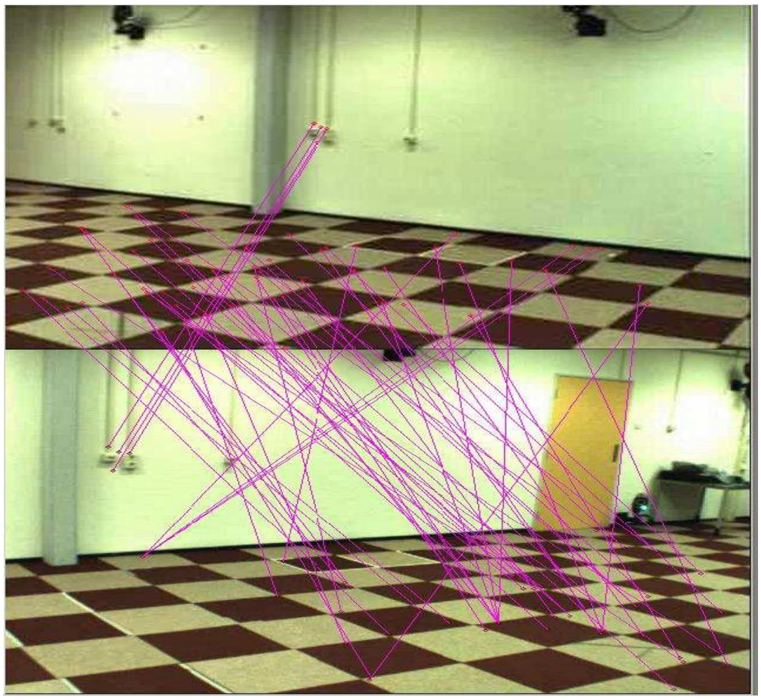


Figure 19: SIFT features obtained with a threshold value of 1.5

Figure 20 shows the SIFT features we retrieved from the left and right views. The SIFT feature point is the starting point (dot) of the SIFT feature vector (arrow).

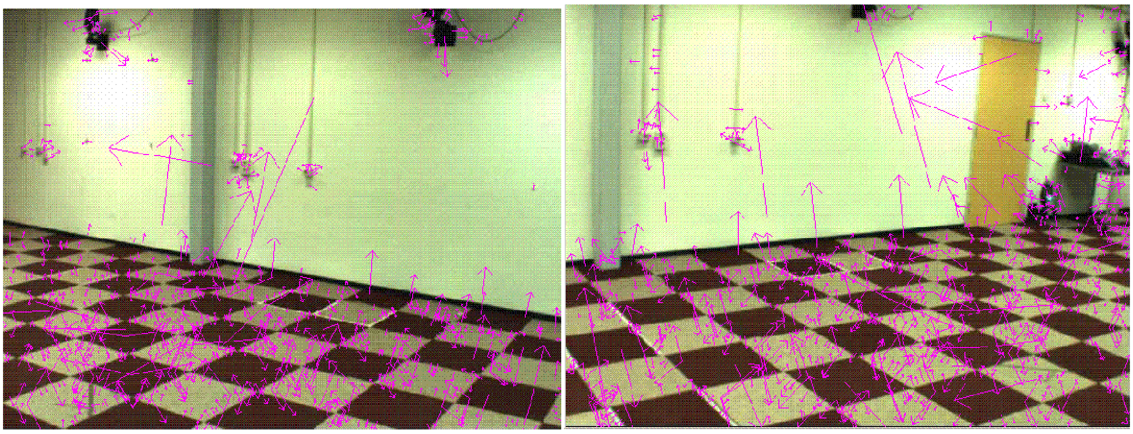


Figure 20: SIFT features retrieved from the two views. The origins of the arrows indicate the location of the point features and the direction vector is represented by the direction of the arrow itself.

As can be seen from the above results, we have detected almost no feature point on the wall, but the density of the SIFT points on the floor is quite high. Some key observations were made from the SIFT features which were extracted,

Observation 1: The SIFT features are mainly seen on regions on the image which have texture.

Observation 2: There is a very low density of features seen on the wall of the image as opposed to the floor.

4.3 Evaluation of the dense pixel-to-pixel correspondence map

The background scene of our data set consists of a wall and a floor. Therefore, to generate a dense pixel-to-pixel correspondence map of the background images, we computed two homographies between the pair of images: one for the floor and one for the wall. To define the homographies, it was necessary to hand label four pixel-to-pixel correspondence points on each of the planes for both views. Figure 21 shows the regions of which we took the corner points to compute the homography matrix.

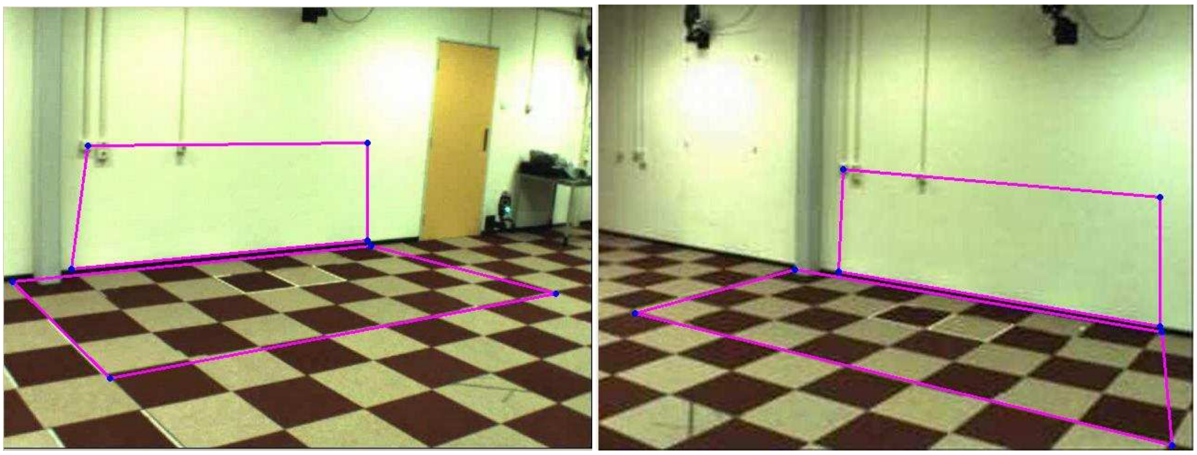


Figure 21: The corner points of the regions indicated by the colored lines are the points selected in the image pair for computing the homography. These points are manually chosen for each plane.

The homography matrix defines a relation between a point q_{floor} belonging to the floor on image 1 and Q_{floor} belonging to the floor image 2 simply as:

$$q_{floor} = HQ_{floor}$$

A similar expression holds when we relate between points on the wall as,

$$q_{wall} = HQ_{wall}$$

To make sure, we only have the pixels that are visible in both images, we back project the computed correspondences to the original image. Thus,

$$Q_{floor} = H^{-1}q_{floor}$$

Figure 22 shows the result of matching the pixels belonging to the wall and the floor that are visible in both images by using the computed homography of the wall plane and the floor plane. As can be seen from the output, only the regions on both the planes which are visible to both the views, the correspondences are defined.

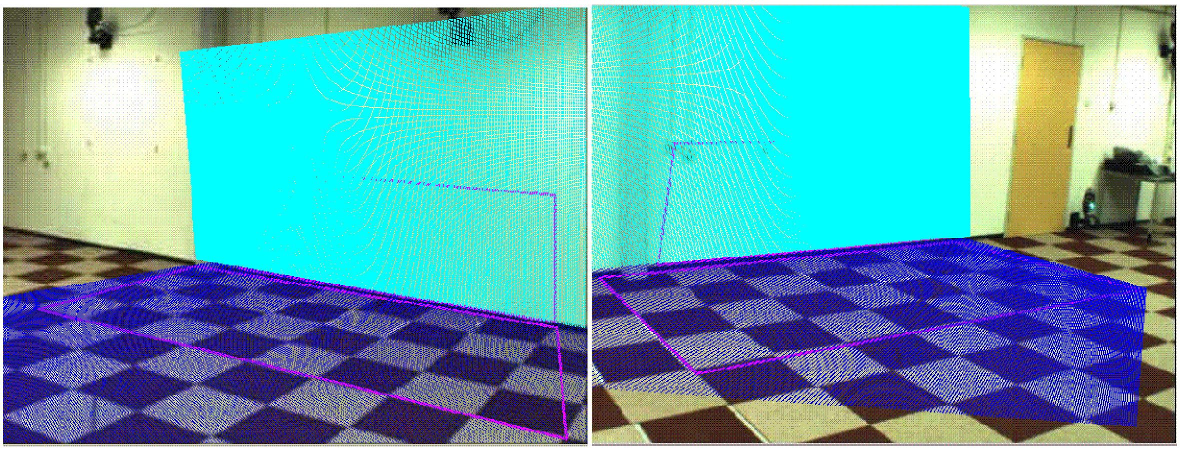


Figure 22: The blue region indicates the points that have a pixel-to-pixel correspondence for the floor on both views. The cyan region corresponds to pixel-to-pixel correspondences on the wall.

There are mainly two ways of evaluating the accuracy of the dense correspondence map. The first way is by taking a set of random test points on the left image and plotting their matching points on the right image. We term it as "evaluation using manual point selection". The second approach is to use epipolar geometry for validation.

4.3.1 Evaluation using manual point selection

In this approach we selected a set of random pixels from the left image and plotted their correspondences from the right image. Figure 23 shows the matching coordinates that we obtained across the two views.

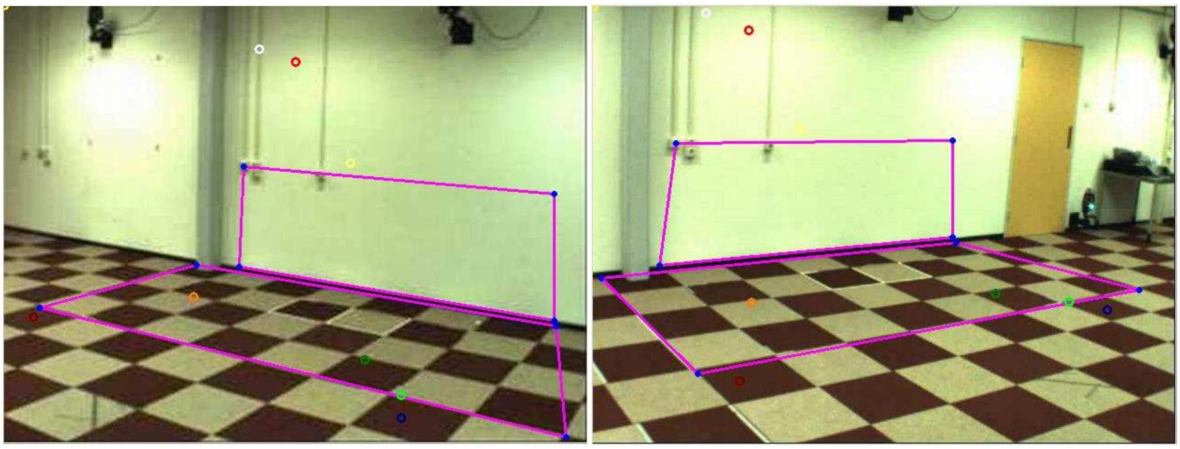


Figure 23: Manual point selection and matching across two views. The matching coordinates have the same color in both the views.

As can be seen from the results, the matches are quite accurate across both the views. The next step is evaluation using epipolar geometry.

4.3.1 Evaluation using epipolar geometry

The concept of fundamental matrix has been discussed in detail in Section 3.4. Simply stated, for each point on one image, the fundamental matrix defines a corresponding line in the other image, the epipolar line.

To verify the accuracy of the dense correspondence map, we take a set of sample "test" coordinates on the left image and display them. For each of these "test" coordinates we display their corresponding epipolar lines on the right image. Then we validate the results by observing if the epipolar lines intersect with the matching coordinates on the right image.

To compute the epipolar line for a given pixel in one view, it is necessary to compute the fundamental matrix across the two views. From our test dataset we were able to obtain three fundamental matrices.

1. Fundamental matrix obtained from internal camera parameters.
2. Fundamental matrix obtained from the SIFT matching feature points.
3. Fundamental matrix obtained from the matching points from the homography computation itself.

We evaluate each of these above mentioned fundamental matrices over the test image points which we choose randomly.

From our experiment we could deduce that the results obtained were fairly accurate with all the test coordinates and their matches were seen to be lying on the computed epipolar line at all times. Figure 24 shows the two views with the randomly selected "test" coordinates on the left view and the "match" coordinates on the right view. One can see that the epipolar lines coincide with the "match" coordinates on almost all cases.

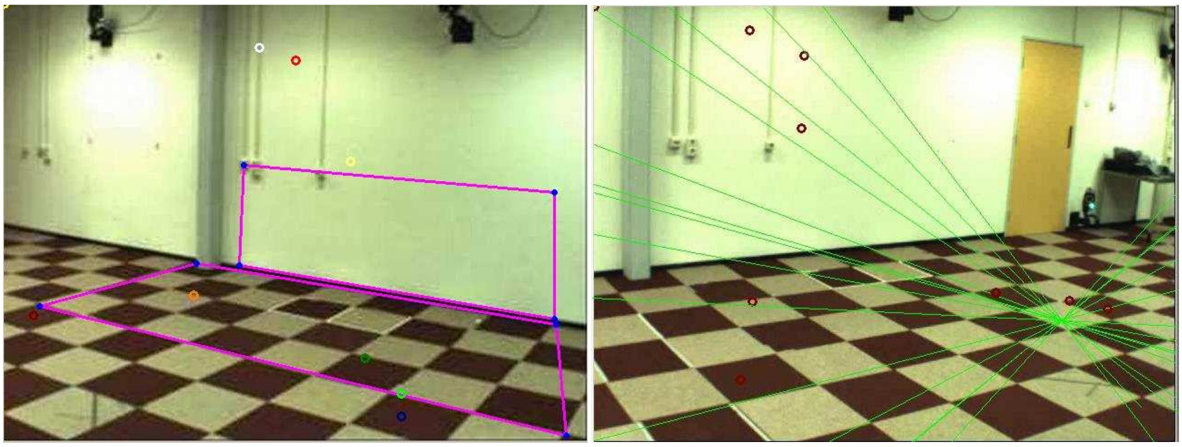


Figure 24: Epipolar line representation of the test coordinates from left image on the right image. The fundamental matrix used for computation was obtained using SIFT key feature point matches.

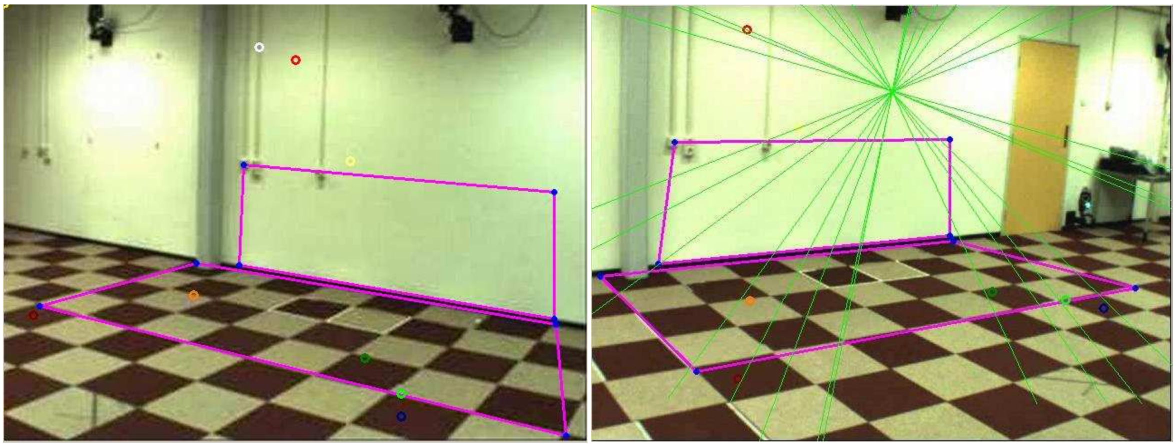


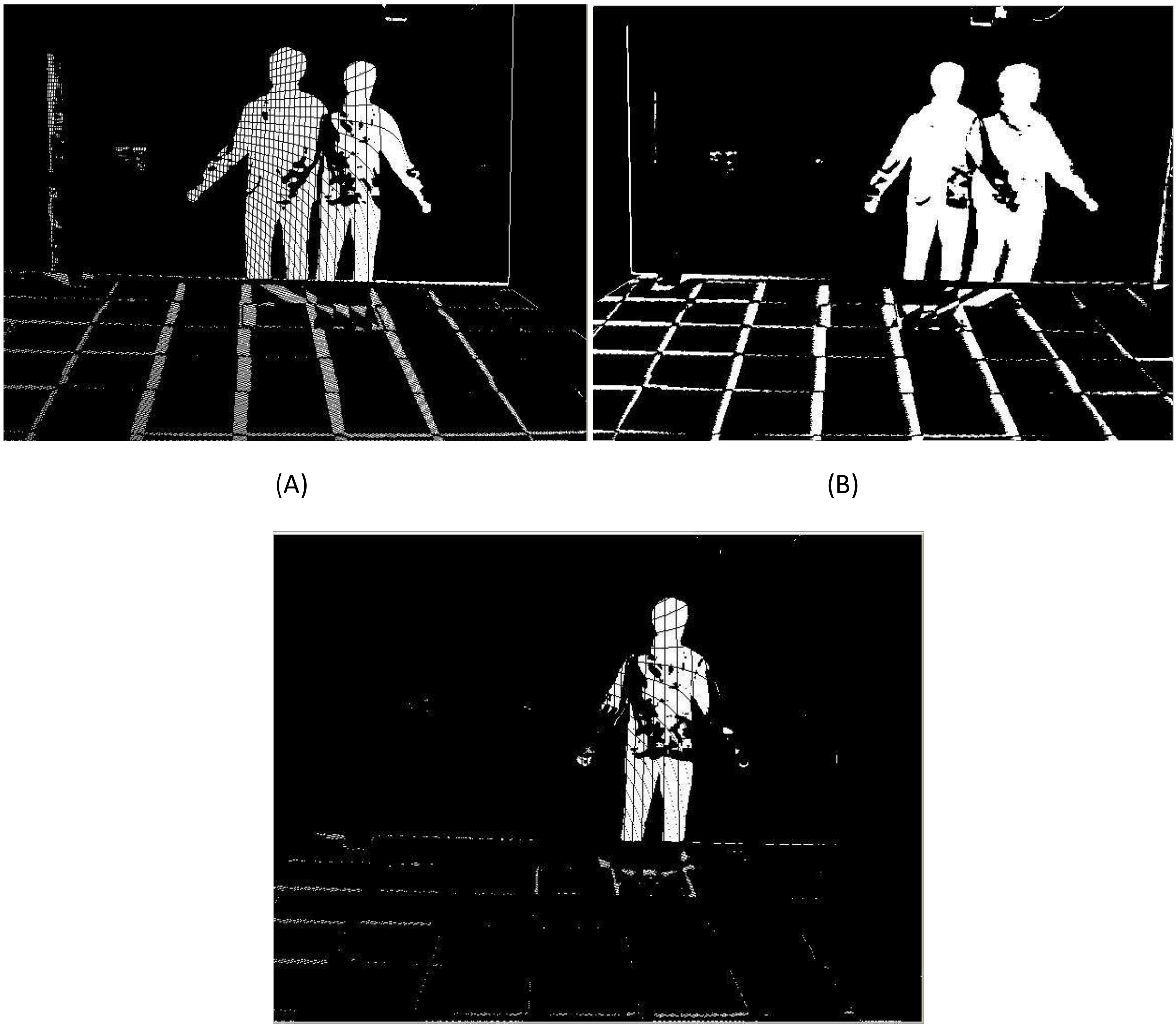
Figure 25: Epipolar line representation of the test coordinates from left image on the right image. The fundamental matrix used for computation was obtained from the matches from the homography matrix.

As can be seen from the above image, the validation using epipolar geometry provides good results. Most of the matching points on the right image either coincide or lie close to the epipolar lines corresponding to their matches on the left image.

4.4 Evaluation of Background Subtraction

In this section the multi-view background subtraction results are shown for a single person as well as a multiple person cases. A comparison is done with “frame differencing” and “kernel density estimation” method for the same input set.

Figure 26 shows the background subtraction results which we obtained by using multi-view geometry. The top views are the result of using the dense pixel-to-pixel correspondence map to compare intensity values of pixels in one view corresponding to the pixel in the other view. The right top view is the result of comparing the “front-left” view with the “front” view, which is taken to be the primary view. The left top view is the result of comparing the “front-right” view with the primary view. The bottom figure is the result when the AND operator is applied to both images, giving us the background subtraction result for the primary view.



(A)

(B)

Figure 26: The above two figures indicate the foreground objects obtained with their silhouettes due to each of the views on the front view. The figure below is obtained after removing the occlusion shadow from both the views.

From the above output we can make several observations. Each of them are listed below.

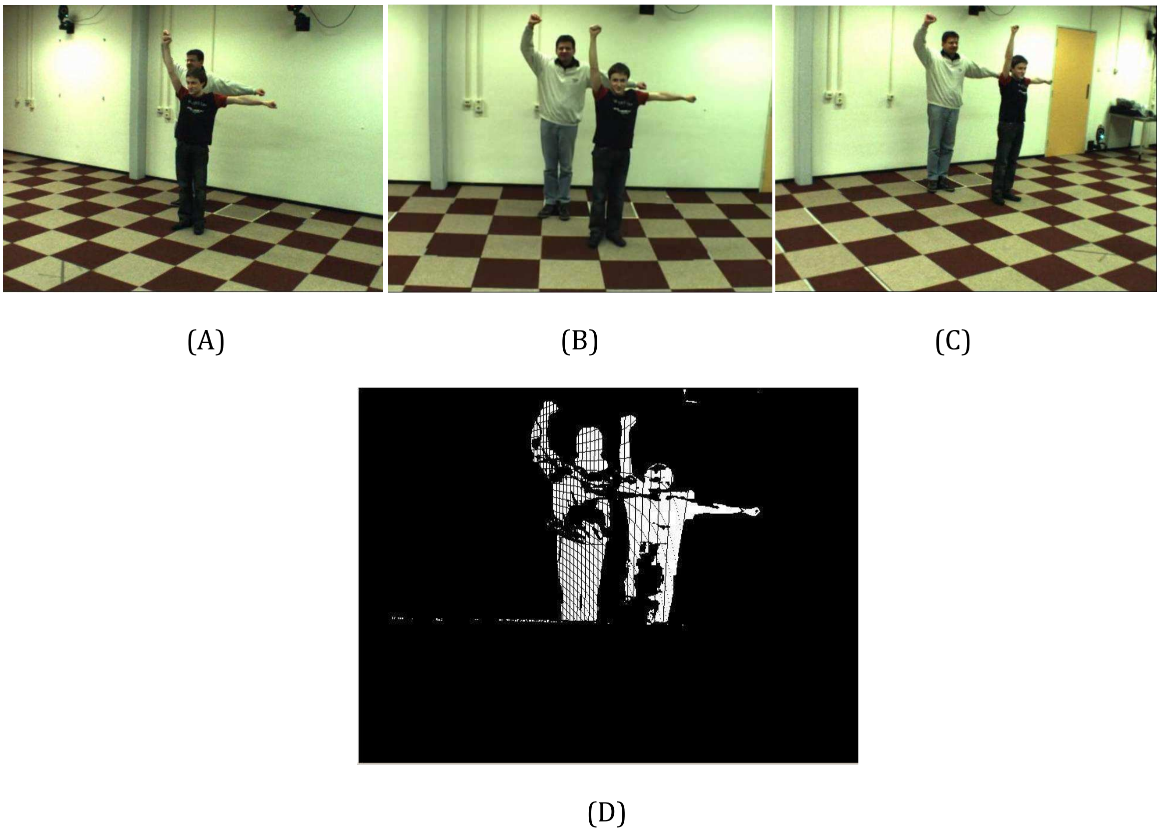
Observation 1: The final output as can be seen is invariant to shadow.

Observation 2: Since the AND operator was used to achieve the final result, there are regions on the actual foreground which are removed. This happens because the silhouettes on both the views remove parts of the actual foreground when they occlude. This however can be improved by optimizing the code.

Observation 3: There is some amount noise that is visible in the initial foreground image (such as the gridlines). This happens due to noise that is added by the homography matrix that is computed.

A more accurate homography matrix can always be computed by taking a higher number of matching point samples.

We conducted another experiment with the same setup but this time with multiple persons. Figure 27 shows the input set and the results of the background subtraction.



(A) (B) (C)

(D)

Figure 27: (A), (B) and (C) Show the input images in the left, front and right views respectively. (D) Foreground object obtained from the input views

Observation 1: Despite regions that are clearly overlapping in the foreground objects in the left view, the results are not affected. This is since the same regions are "not" occluded in the right and front view.

Observation 2: The test foreground object in this case has clothing which is quite similar to the color of the background wall itself. Yet, it does not affect the overall quality of the output.

4.4.1 Comparison with Kernel Density Estimate

Figure 28 shows the output that was obtained with the default settings using the kernel density estimation background subtraction method. This output was obtained for a multiple person case from the Utrecht University MoCap dataset.

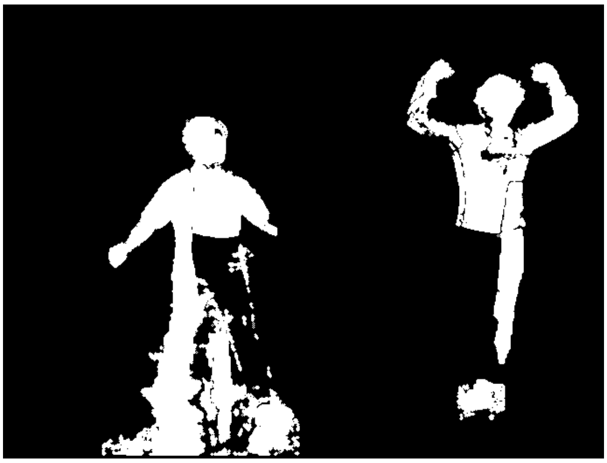


Figure 28: Output foreground object obtained using the kernel density estimation method.

Observation 1: Since the background is primarily static, almost no part of the background is registered as a foreground object.

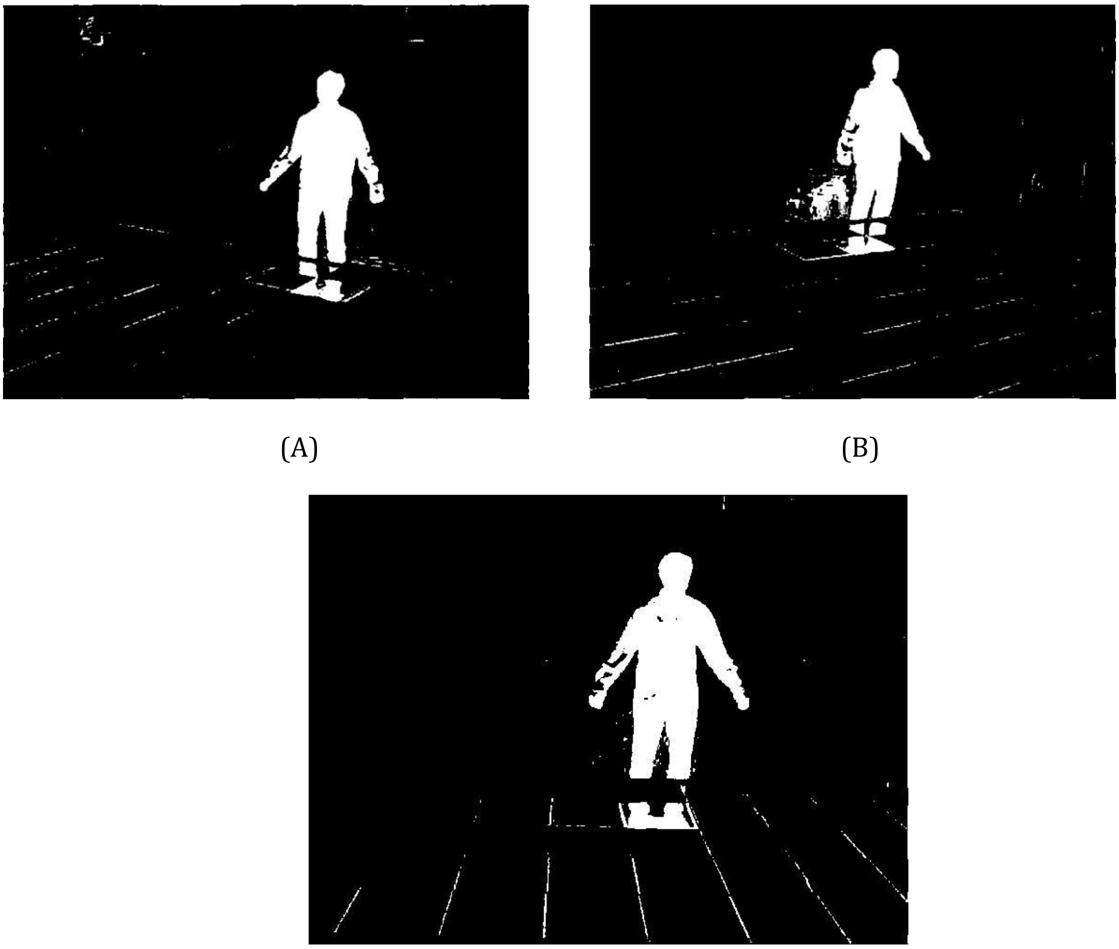
Observation 2: This result was obtained by training the first 40 frames of the movie file. A better and larger training set can provide better foreground image.

Observation 3: A large part of the actual foreground is not detected in the final output image. This can be due to the parameter settings of the method which can be optimized for better results.

Observation 4: There is a significantly large section of the foreground that is not visible. This region of the foreground however is the region that has the same color as the background (the color of the shirt worn by the person).

4.4.2 Comparison with Frame Differencing

Figure 28 shows the results we obtaining by using frame differencing for background subtraction for the three input views. Due to the simplicity of the model, the results obtained are poor. Also it is evident that the method is not invariant to shadows since the shadows are also part of the foreground.



(A)

(B)

(C)

Figure 29: (A) Frame differencing result obtained on the left view (B) Frame differencing result obtained on the right view (C) Frame differencing result obtained on the front view.

Observation 1: As can be seen from the output the frame differencing is not invariant to shadows.

4.4.2. Time Complexities

The entire algorithm was tested over an Intel® Atom™ CPU with a 1.66 GHZ processor and an onboard RAM of 1.99 GB. We used C++ and the language for implementation and OpenCV as the library for computational and visual purposes.

The initial size of the input images is 640 × 480 pixels. With this setting the background subtraction happens at the rate of 0.285 frames per second. However for the purpose of a real-time background subtraction, the images were down sampled to a resolution of 320 x 240 pixels. Using this resolution each frame was processed in 0.068 seconds meaning an approximate 20 frames per second.

5. Conclusion and Future Work

5.1 Conclusions

A background subtraction method is investigated, which uses multiple camera views to use information of the world scene. Compared to traditional background subtraction methods, this method do not rely on a background model created in the past, and therefore it is invariant to illumination changes in the scene.

This method assumes a static background such that the foreground detection can be made real-time. The basic idea of the method is that pixel-to-pixel correspondences between the multiple views will only change if a foreground object appears. Hence, this pixel-to-pixel correspondence map has to be initialized only once, and can be used in the foreground detection.

The background subtraction method starts with the generation of a sparse correspondence map. To generate these initial matches we initially chose SIFT for the key point descriptor. We obtained a good number of matches across both the views. The Utrecht University MoCap dataset has regions on the images which do not have much texture information. In the experiments, it was observed that SIFT has downfalls, since the key points detected are not uniformly distributed on the test images. This was a motivation to use homographies for the two image planes in the two views.

The homography matrix was computed by choosing an initial set of points as matches and a dense pixel-to-pixel correspondence was obtained from this set. A thorough evaluation of the accuracy of this method was done by using epipolar geometry. The results reflected that the obtained matches were accurate. A set of points were randomly chosen on the left image and the matches were plotted on the right image to verify the accuracy of the correspondence.

With this dense correspondence map, the background subtraction step was done for the 3 views. Two foreground images with one of them between the “left-front” and the “front” and the other between the “right-front” and the “front” views were obtained. The occlusion shadows were removed by using the AND operator. A final foreground image is retrieved after this AND operator with promising results. The foreground image obtained is invariant to shadows and intensity changes. A similar experiment was done with multiple persons on the input views, and nice results were obtained even in this case. Even with overlapping regions in one view, the foreground image remained unaffected. This was since the corresponding pixels in the other view were not overlapping.

Using a pixel-to-pixel correspondence was required to help with the speed of execution of the background subtraction step. This was verified by measuring the time complexities. It was found that the background subtraction happens in real time at 20 frames per second making it useful for all practical purposes.

5.2 Future Work

In the traditional background subtraction methods several extensions to the frame difference background subtraction method have been provided such as the single Gaussian, mixture of Gaussians and the kernel density estimate method, to handle noise, update the background model and remove the need of manually setting thresholds. A similar extension can be thought of in our implementation of the multi-view background subtraction algorithm. If we were to train the sample sets over a period, then clearly it is possible to reduce the influence of noise in the creation of the dense pixel-to-pixel correspondence map.

The background subtraction method makes use of a dense pixel-to-pixel correspondence map across the different views in the input dataset. Since SIFT had its downfalls with our dataset we made use of the homography computation. There are many more approaches to compute the dense correspondence provided the regions have sufficient amount of texture information. Using a quality function which takes into account intensity and depth information of the pixel to measure the quality of matches is a novel approach.

Our method makes use of an intensity threshold value for background subtraction. A pixel on the primary view is considered as part of the foreground if it exceeds this threshold when compared with its corresponding pixel on the auxiliary view. At this stage of implementation, this threshold value is manually specified. It is possible to estimate a more accurate threshold value with a probabilistic approach.

There are mainly four points chosen on each of the image plane pairs to define the homography. Having a higher number of matching points will improve the accuracy of the homography and hence the quality of the output mapping.

Our method uses three camera views for the multi-view background subtraction to remove the occlusion shadow. With a higher number of views, the quality of the result can be improved. Hence a direct extension of this method includes N views.

References

1. *Fast lighting independent background subtraction.* **Y. Ivanov, A. Bobick, and J. Liu.** 2000. International Conference on Computer Vision.

2. *Pfinder: Real Time Tracking of Human Body.* **C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland.** 1997. IEEE Transactions on Pattern Analysis and Machine Intelligence.

3. *Adaptive background mixture models for real-time tracking.* **Grimson, Chris Stauer and W. Eric L.** 1999. IEEE Conference on Computer Vision and Pattern Recognition (CVPR1999).

4. *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance.* **A. Elgammal, R. Duraiswame, D. Harwood, and L.S. Davis.** 2002. Proceedings of the IEEE.

5. *Quasi-Dense Wide Baseline Matching Using Match Propagation.* **Brandt, Juho Kannala and Sami S.** 2007. IEEE Conference on Computer Vision and Pattern Recognition (CVPR2007).

6. *Match propagation for image based modeling and rendering.* **Quan, M. Lhuillier and L.** 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence.

7. *Dense Matching of Multiple Widebaseline views.* **Christoph Strecha, Tinne Tuytellers, Luc Van Gool.** 2003. Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV'03).

8. *Wide-baseline Stereo from Multiple Views: a Probabilistic Account.* **Christoph Strecha, Rik Fransens, Luc Van Gool.** 2004. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04) .

9. *Dense Correspondence Extraction in Difficult Uncalibrated Scenarios.* **Ruan Lakemond, Clinton Fookes, Sridha Sridharan.** Image and Video Laboratory, Queensland University of Technology : s.n.

10. **J. Yao, W.K. Cham.** *Feature matching and scene reconstruction from a set of unordered widely separated views.* 2005.

11. *Object recognition from local scale-invariant features.* **Lowe, David G.** 1999. International Conference on Computer Vision.

12. *Robust Wide Baseline Stereo from Maximally Stable Extremal Regions.* **J. Matas, O. Chum, M. Urban, T Pajdla.** 2007. British Machine Vision Conference.

13. *Scale-space theory: A basic tool for analysing structures at different scales.* **Lindeberg, Tony.** 1994. Journal of Applied Statistics.

14. *3D modeling and rendering from multiple wide-baseline images by Match Propagation.* **Jian Yao, Wai-Kuen Cham.** 2006. Signal Processing: Image Communication 21.

15. **Richard Hartley, Andrew Zisserman.** *Multiple View Geometry in Computer Vision.*

16. **M. A. Fischler, R. C. Bolles.** *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.* s.l. : ACM, 1981.