VIDEO-BASED CHANGE DETECTION

A Dissertation Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Hakan Haberdar

August 2013

VIDEO-BASED CHANGE DETECTION

Hakan Haberdar

APPROVED:

Dr. Shishir K. Shah, Chairman
Dept. of Computer Science

Dr. Edgar Gabriel
Dept. of Computer Science

Dr. Ricardo Vilalta
Dept. of Computer Science

Dr. George Zouridakis
College of Technology

Jayan Eledath
SRI International

Dean, College of Natural Sciences and Mathematics

Acknowledgments

First of all, I thank God for giving me the chance to have this experience. My special thanks go to my advisor Dr. Shishir K. Shah for his guidance, his insightful comments, and his support throughout this endeavor. I do believe in that Dr. Shah is a great Ph.D. advisor. I would like to express my appreciation to my dissertation committee for their valuable suggestions. Throughout my graduate studies, there have been some special people who gave their support which I will never forget, Dr. Garbey, Dr. Hilford, and Dr. Paris. I also want to thank to the staff at the department for always doing the best that could be done, Yvette, Liz, Jackie, and Anh.

I also extend special thanks to all my friends for their help, moral support, and, above all, their friendship, Saber Feki, Apurva Gala, Charu Hans, Serhat Okyay, Mehmet Ali Ozbay, Gokhan Ozer, Ahmet Sonmez, Khai Tran, Tayfun Tuna, Ilyas Uyanik, Xuqing Wu, Xu Yan, and Erol Yeniaras.

Last but not least, I am grateful to my parents Melahat and Abdulaziz, my sister Meltem, my brother Fatih, and my parents-in-law, Umran and Hasan for their love and encouragement. Nothing would be possible without the love of life, my wife, my Rubabe. I wish we had our beloved daughter Zehra earlier, so we could just look at her face and fingers and feel the joy whenever either I or Rubabe had a tough time during our Ph.D. studies.

VIDEO-BASED CHANGE DETECTION

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Hakan Haberdar

August 2013

Abstract

Change detection from video recordings is critical in many applications related to surveillance, medical diagnosis, remote sensing, condition assessment, motion segmentation, and advanced driver assistance systems. The main goal of the change detection using videos is to identify the set of pixels that are significantly different between spatially aligned images that are temporally separated.

This is an extremely challenging problem because of a variety of factors, including changes in the illumination over time, appearance or disappearance of objects in the scene, and the need for temporal synchronization of the videos. Moreover, when a mobile video acquisition platform is used, a change in scale of the observed scene along with rotation and translation changes between image pairs is introduced. Thereby, the imaging geometry cannot be modeled by ordinary transform constraints because of the varying field-of-view.

Over the years, many standard image processing techniques have been leveraged to realize a solution to the problem of change detection. Each potential approach attempts to exploit properties of the image, the application domain, or a combination. The relevance of the kind of changes to be detected is application-specific, but the underlying algorithms need to detect all changes as the first step, which can later be post-processed to discriminate between relevant and unimportant changes. It would be beneficial to have a framework that analyzes the changes between videos in an automated manner.

In this dissertation, we explore more complex imaging models for solving the change detection task and propose a complete framework that accomplishes spatiotemporal registration and change detection. To this end, we develop a set of

methods for: 1) temporal alignment of the unsynchronized videos, 2) estimation and refinement of the disparity maps using temporal consistencies, 3) segmentation of the dominant plane in the scene, 4) estimation of spatial transform for the dominant plane, and 5) detection of relevant changes in the presence of several altering background elements.

To demonstrate the feasibility of the proposed methods, we carried out extensive experiments using videos obtained from various sources and present visual and quantitative results that address: 1) temporal alignment of video pairs recorded by mobiles platforms under varying illumination and scene conditions, 2) scene depth estimation, dominant plane segmentation, and change detection between videos captured by moving sensors where the complicated geometry and parallax are present, and 3) detection of relevant changes in videos acquired by stationary cameras where the environment contains several dynamic regions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The detection of relevant changes in videos of the same scene acquired at different instances is crucial in vision tasks, especially in applications related to video surveillance [42, 76, 86, 152, 176], remote sensing [17, 27, 102, 121, 154], medical diagnosis [12, 21, 80, 165, 182], condition assessment [72, 84, 97, 114, 181], motion segmentation [29, 77, 115, 118, 142], and advanced driver-assistance systems [22, 70, 89, 101, 157]. Despite the diversity of applications, change detection methods employ many common processing steps [148] to detect application-specific regions of interest. The core problem is to identify the set of pixels that are significantly different between two images of the same scene, and these pixels comprise the region of change. The region of change may result from a combination of factors, including appearance or disappearance of objects, motion of objects relative to the background, or shape changes of objects. The key issue is that

the region of change should not contain irrelevant forms of change, such as those induced by camera motion, sensor noise, illumination variation, and nonuniform attenuation.

Our data acquisition scheme is the one used by typical vision applications where mobile or stationary sensors collect either monocular or stereo data for change detection purposes. A desired change detector should automatically correlate and compare multiple videos of the scene from different viewing angles at different times and then exploit the ability to identify changes. Despite recent progress in computer vision, there are still major challenges to be overcome in realizing a robust solution to the problem of change detection. The challenges are attributed to complex variations in the appearance of dynamic scenes, unknown calibration parameters of video acquisition platform, temporal alignment of videos, and changes in the illumination over time.

1.1 Motivation

In the last decade, the use of visual imagery for the problem of change detection has become one of the most significant and active areas in the field of computer vision because of the advances in the imaging technologies. Depending on the specific goal of the change detection application, the data acquisition unit could capture images by various imaging modalities, such as stationary video cameras [10], mobile camera platforms [37], MRI scanners [21], computed tomography scanners

[80], and remote sensing [154]. Multiple stationary cameras or mobile platforms are used to observe large environments for the surveillance. It is not possible for a single stationary camera to capture the complete area of interest because of the finite sensor resolution and structures limiting the visible areas [1]. MRI scanning is an important diagnostic tool for monitoring disease evolution in medical imaging [21]. Similarly, computed tomography scanner creates tomographic images of cross-sections of the specific areas of the body. These cross-sectional images are used for diagnostic purposes, such as the risk of cancer and heart diseases and detecting tumors [80]. In the remote sensing systems, multispectral data are processed and interpreted for various purposes, for example environmental changes, land usage monitoring, building damage assessment, and aerial traffic control [31].

The constant flow of video from a number of cameras provides data but no actionable information [71]. Processing such data by a human is tiring, expensive, and ineffective. After only 20 minutes, human attention to video monitors degenerates to an unacceptable level [74]. Therefore, a common practice in commercial stores or banks is to record the videos on tapes and use them as forensic tools, i.e., after a crime, the recorded video is used to collect evidence [99]. Especially for the scenarios where a mobile platform patrols sensitive areas, the task of the human observer is even more difficult and almost impossible because the observer is supposed to remember the condition of the area in the previous recording of the scene.

Limitations of the human observers and the amount of the data to be processed can be practically and effectively achieved by automating the monitoring process and employing human interaction only for evaluation of anomalies that are detected. Our motivation is to address the need for developing methods for the automation of the change detection task.

The core problem discussed in this work is as follows. *Given two videos of the same scene captured at different times under different illuminating conditions, we aim to develop a framework for detecting relevant changes in the scene.* Namely, the goal of this dissertation is to identify image regions that are significantly different between two videos that may be temporally and/or spatially separated.

1.2 Challenges

Despite the diversity of applications, a basic video change detection framework mainly follows a four-step processing pipeline [48, 165] to accomplish detection of anomalies (Figure 1.1).

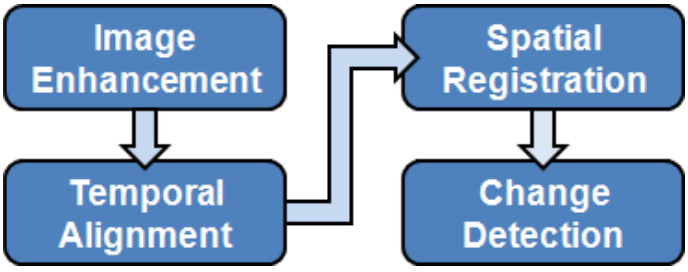


Figure 1.1: Block diagram of the common processing steps used for the change detection problem in different applications.

Apparent intensity changes and noise at a pixel resulting from data acquisition or environmental conditions are virtually never desired to be detected as real changes. Hence, a necessary preprocessing step for all change detection algorithms is accurate image enhancement. Then, establishing correspondences in time and in space between different videos of the same dynamic scene allows us to make a direct observation of relevant changes in the scene.

While the complexity of a real change detection problem solely depends on the characteristics of data sets [50], one working on a robust change detection framework should take a number of apparent challenges into account because of the nature of problem. Those challenges are introduced and investigated in the following sections.

1.2.1 Image Enhancement Challenge

A major issue with change detection in videos is to guarantee robust detection in the presence of illumination variations and noise. Illumination variations can be observed at each pixel within spatiotemporally aligned frames because of a variety of factors [148], including slightly different viewing angle of the camera, and change in the position and intensity of direct or ambient light sources. In addition, nonrigid deformations of the objects in the scene may cause intensity changes. Furthermore, in an outdoor environment, illumination not only changes slowly as daytime progresses, but may change rapidly because of changing weather conditions and passing objects (e.g., cars, airplanes, clouds, and overpasses) [25].

In this case, changes in the illumination do not have to be global. For instance, there may be shadowed regions in an outdoor scene on a partly cloudy day. These illumination variations introduce challenges on almost all of the modules of a change detection system.

1.2.2 Temporal Alignment Challenge

When the change detection problem occurs in the context of multiple videos, it is natural to exploit the temporal consistency of pixels in the same location at different times [148]. Videos of the same scene (Figure 1.2) may differ from one to another because of a mobile camera platform and scene dynamics. The primary challenge is to decide which pair of frames should be selected for the spatial registration.

This problem can be addressed by bringing two videos recorded at different times into temporal alignment. The process of establishing temporal correspondence has to deal with various imaging and scene conditions. Especially working with moving cameras increases the number of system parameters (e.g., time offset and external camera parameters) dramatically [146]. Camera parameters may change from frame to frame, and they should be determined or at least compensated for each individual frame.

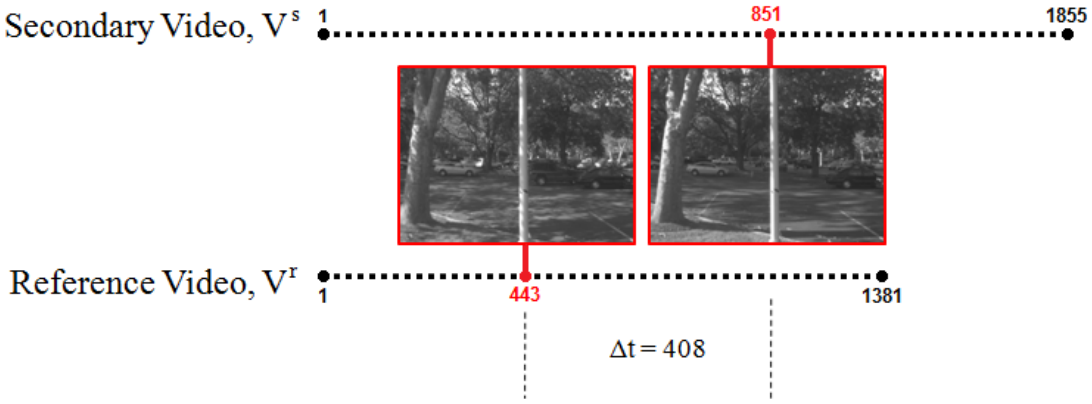


Figure 1.2: Illustration of the temporal alignment. Given two unsynchronized videos V^r and V^s of the same environment, where r denotes *reference* and s denotes *secondary*. V^s was recorded after objects of different sizes and textures were placed in the outdoor environment, and V^r was taken without the objects. Frames I^r_{443} and I^s_{851} have the most similar view. Red points indicate the synchronization points. The time offset, Δt, is 408 at the synchronization point, where Δt may changes from point-to-point.

1.2.3 Spatial Registration Challenge

The existence of temporal alignment step poses additional registration challenges compared to an ordinary image registration problem. Furthermore, complex variations in the appearance of dynamic scenes include nonrigid changes in the scene (e.g., waving tree branches) and parallax caused by change in the observation position (Figure 1.3).

Another practical issue regarding registration is the selection of feature-based or intensity-based registration algorithms. Determining an appropriate transform model relating pixel coordinates in one image to pixel coordinates in the other image is the main challenge. A variety of such parametric transformation models

(a) (b)

Figure 1.3: Example of two frames from temporally aligned video pairs acquired by a mobile platform. These two frames have the most similar view of the scene shown. Nevertheless, the moving camera platform causes a significant chance in the viewing angle between the frames. This introduces parallax among the trees in the scene.

are possible such as similarity (i.e., scaled rotation), affine, projective (i.e., homography), and quadratic. After a suitable transformation model is determined, parameters of the transformation model should be estimated.

1.3 Research Goals

In this work, we aim to establish correspondences in time and in space between two videos of the same dynamic scene acquired under different conditions and subsequently detect the regions of change between the spatiotemporally aligned frames of the two videos. In order to achieve this goal and to overcome the challenges described in Section 1.2, we propose a set of *novel methods*, each of

which addresses different subproblems of a generic video-based change detection framework.

Our first subgoal is to bring two videos taken at different times into temporal alignment. The main idea is to combine the mutual information observed from one set of matched frames and the information provided by the dynamics of the scene. This will provide us varying dynamic time offsets and corresponding frames that have the most similar view of the same scene between the two videos.

Various spatial registration algorithms have been presented in the computer vision literature. Nevertheless, each one of the existing approaches rely on tuning of different sets of parameters, which make them challenging to use for the dynamic environments. One should employ and integrate existing techniques and develop a registration module that is robust under complicated scene structure variations. Assuming the existence of a dominant plane and using only the dominant plane regions for spatial registration can overcome some difficulties related to the scene structure, but the detection of the dominant plane remains a challenging problem. There are several algorithms using the idea that all points obeying the same plane equation will be in the same layer within a frame, and those points in the layer can be employed to calculate parameters of the plane equation. Nevertheless, there are many cases where this assumption does not work because of complex geometry of the dominant plane. Hence, we need to design an alternative approach to estimate the dominant plane using a more robust scene feature, such as the scene depth.

Image change detection step is the most critical task of the proposed video-based change detection framework. We need a change detection strategy that is robust against illumination changes and a reasonable range of spatial distortion. In addition, a sophisticated change detection method should discriminate the relevant changes from the others where the background has several altering elements that may cause false alarms.

All algorithms, approaches, and techniques described in this section will be evaluated in all aspects of our central goal. For generalization, both monocular and stereo videos are considered on this work. To present the correctness and the benefits of each module, we used a video benchmark that includes recordings acquired in wide variety of conditions, such as videos recorded in real outdoor environments at night and day time, videos which include both challenging types of scene structures and changes of different sizes and textures, and videos captured by mobile platforms.

1.4 Dissertation Outline

The organization of the remainder of this dissertation is as follows. We begin in Chapter 2 by presenting a literature review of existing approaches to each of the steps that build up the entire framework. Chapter 3 describes modules for the temporal and spatial alignment procedures. We present the proposed image change detection methods in Chapter 4. In Chapter 5, we discuss experimental

results of different modules of the video-based change detection framework. Finally, the last chapter summarizes the dissertation and highlights its contributions along with presenting the future perspective of this work.

Chapter 2

Related Work

This chapter reviews and discusses performance, relative merits and limitations of existing approaches for each step in a video-based change detection framework. Several approaches for analysis of videos are provided in the literature. Although each system may require application-specific tasks, modules of a typical system can be summarized as illustrated in Figure 2.1.

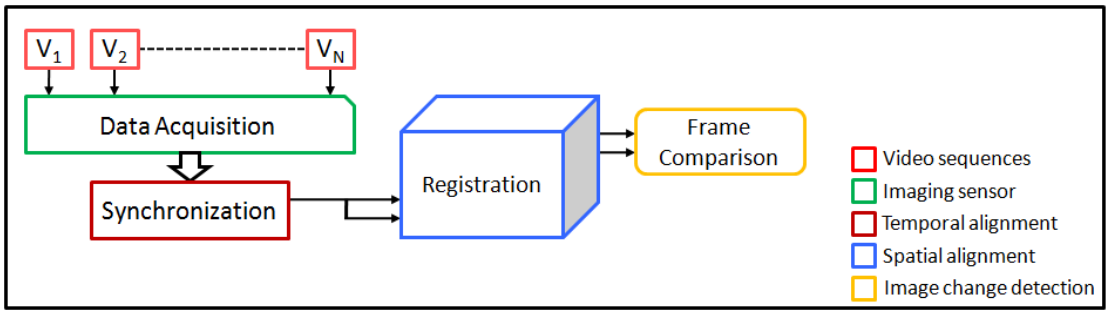


Figure 2.1: Schematic overview of a typical video-based change detection system.

In different application scenarios, different number of videos may be available

[47], while in most cases two videos are used [30, 35, 160]. Based on the limitation of the environment, stationary cameras, moving cameras [160], or rigidly attached cameras [197] can be employed for the data acquisition. Accurate temporal alignment to find corresponding frames in the videos is the first step towards robust spatial image registration. Because the complexity of the spatial registration depends solely on the data provided, each approach may require application-specific substeps. In the following sections, we present a review of previously proposed video-based change detection frameworks and discuss individual modules in these frameworks.

2.1 Video-based Change Detection Frameworks

The term video registration in literature has been interchangeably used for two different problems. The first one refers to the problem of registering frames of a single video to one of the other frames in the video. The second problem addresses the spatial alignment of two different videos of the same scene recorded at different times. In this dissertation, we focus on the task of registering multiple videos.

Various vision-based methods for detecting objects in a scene using stationary cameras are presented in the literature, while considerably fewer publications address the analysis of images acquired by mobile camera platforms. Sand and Teller [160] describe an algorithm for bringing two non-stereo videos into spatiotemporal alignment and compare them using feature point correspondences.

First, they determine a pixel matching probability using 3×3 maximum and minimum intensity filters. A motion consistency probability is calculated by starting with a corner detector for each candidate match. The product of pixel matching probability and motion consistency probability finally determines the best match. The limitation of their approach is that they require the input videos to follow spatially similar camera trajectories. Therefore, the algorithm cannot align images from substantially different viewpoints. Primdahl et al. [146] presents a method for autonomous control of moving vehicles making repeated passes through a very specific, well-defined indoor environment. They utilize measurements from a positioning system to find corresponding frames from different videos. After the frame alignment, the presence of new objects within the frame is simply detected by comparison of the gradient information. Caspi et al. [35, 36, 185] have developed a unified framework for the problem of video registration. The registration is performed by maximizing a similarity measure across all the sub-volumes of the videos. Tanjung and Lu [175] propose an automatic change detection method for multiple images of the same scene acquired by a mobile camera from different positions. Their method consists of three steps: 1) automatic image registration, 2) temporal differencing, and 3) irrelevant change removal. The main drawback of their method is that effects of illumination changes are not taken into consideration in any way. Furthermore, the experimental evaluation is very limited and does not provide enough insight about when and why the technique should be expected to yield the desired result. Chakravarty et al. [37] presents a mobile robot which is capable of moving along a route while detecting visual anomalies. They use a

monocular panoramic vision sensor to detect and track the objects. Nevertheless, they assume the nonexistence of any environmental changes and require a manually defined similar trajectory. Buchanan [28] describes a method for the detection of improvised explosive devices in videos taken over roads and terrain at different instances by unmanned aerial vehicles. Scene pixel correspondences are formed by performing image registration through the generation of images of the static scene from multiple viewpoints which have not been previously seen. After estimating the camera parameters, they create a texture-mapped three-dimensional model of the terrain and generate new views of it by direct image rendering. They report results on a few images and the performance of their system is poor when the aerial vehicle follows different paths. In this dissertation, we extend these applications to a more difficult case where a moving camera platform, which utilizes monocular or stereo sensors, follows an unknown trajectory in unknown dynamic environment under changing illumination conditions.

2.2 Temporal Alignment

In a video-matching framework, the first parameter which has to be computed before performing the spatial registration is the time offset between videos. The problem of temporal alignment has been extensively studied, and many approaches have been developed since Stein's first method [171]. Stein achieves the temporal alignment using tracking data acquired from multiple cameras assuming: the

cameras are static, and the images are related by a two-dimensional projective transformation. Caspi and Irani [34] report a direct approach to aligning two videos by finding the spatiotemporal transformation that minimizes the sum of squares difference between videos. Videos are assumed to be captured from two cameras which are mounted close to each other and move jointly. The limitation of their approach is that they align all the frames with a single image transformation and a single time offset. The proposed method fails for videos with dynamic time shifts. Tuytelaars and Gool [184] tackle the problem of video synchronization for independently moving camera platforms. They manually picks five independently moving points which must be tracked successfully to obtain the time shift. An alternative approach for the same scenario is proposed by Whitehead et al. [196]. Multiple objects are tracked through each video, and the salient points in each trajectory are located. Initial estimates of the trajectory correspondences and the time offset are established by locating points from each video which satisfies the epipolar geometry. Temporally aligned frames are then determined by locating frames where the location of the object in all views satisfies the epipolar geometry. This allows the time offset and frame rate to be recovered for all pairs of the videos. Wedge et al. [192] utilize the iterative random sample consensus method [62] to recover transformation from matched spatial features in two images. These matches are then used to compute the time offset. This approach is also limited to stationary or jointly moving cameras. Meyer et al. [125] perform temporal alignment in two steps based on motion trajectory correspondences between two or more videos recorded by unsynchronized non-stationary cameras.

They estimate time offset by analyzing the trajectories and matching their characteristic time patterns. After finding the coarse time offset by extracting salient points of trajectories and matching their time patterns, they use the estimated fundamental matrix to directly calculate the fine time offset.

2.3 Image Registration

Image registration is the process of determining the spatial transformation (e.g., translation, affine, and homography) which maps points from one image (i.e., fixed image) to the corresponding points in the second image (i.e., moving image). The problem of image to image alignment has been extensively studied in the literature, and it is a recurring challenge in computer vision. Image registration is a first step in variety of applications, such as image stitching, medical imaging, image-based modeling and rendering, structure from motion, and object recognition. Traditional methods for aligning images are often subdivided into two main categories: intensity-based methods and feature-based methods. One challenging issue regarding registration is the selection of feature-based, intensity-based, or hybrid registration algorithms.

2.3.1 Intensity-based Spatial Registration

Intensity-based image registration methods first define a metric, such as the sum-of-square differences and mutual information. The registration problem is then

solved by the minimization or maximization of a cost function derived from the metric. These methods employ intensity differences and image gradients to compute an update to the estimate of the spatial transformation. Then, the estimated transform parameters are used to warp the moving image on top of the fixed image. This iterative process continues until a stopping condition is satisfied or a fixed number of iterations is reached.

An early attempt of a widely used image registration algorithm is the optical flow method, which was developed by Lucas and Kanade [119]. Optical flow establishes a similarity metric and produces a dense pixel correspondence; however, it is not robust to occlusions. One disadvantage of the pixel-based methods is that they require high computational time (for both model construction time and prediction time). To overcome this limitation, techniques using a hierarchical coarse-to-fine approach with image pyramids have been developed. First, an image pyramid is constructed, and then a search over a small number of discrete pixels is performed at coarser levels [147]. An optimizer is employed to estimate parameters of the transformation model using a cost function. A commonly used approach is to apply gradient descent [119] on the cost function. For a complex transform model (e.g., homography), estimation of the parameters becomes complicated. Shum and Szeliski [166] propose that this can be simplified by first warping the moving image according to the current transform estimate and then comparing it against the fixed image. Intensity-based registration methods has been used extensively in medical imaging applications [94]. A major shortfall of

the intensity-based approach is that the cost function minimization procedure is quite sensitive to local minima. Accordingly, when the images to be registered have a low-overlap, this approach usually tends to fail.

2.3.2 Feature-based Spatial Registration

Feature-based registration algorithms first extracts distinctive features from the fixed and moving images. The feature points are then mapped from the moving image to the fixed image using matching methods, such as normalized cross correlation. After a rough set of matches are obtained, the closest fixed image point for each mapped point is refined. These temporary correspondences are used to estimate the geometric transformation between the images. These steps are repeated iteratively. Feature-based approached is faster and has the advantage of being more robust against scene variations.

Feature-based approaches have been used since the early days of image registration [78, 133]. They have more recently gained popularity for image stitching applications [158, 169, 210]. The most well-known feature detection methods are Canny edge detection [33] and Harris corner detection [81]. Most recent methods rely on using local descriptors that are more invariant to scale and transformations to estimate the similarity of pixels across images. These local descriptors include scale-invariant feature transform (SIFT) [117], gradient location and orientation histogram (GLOH) [128], speeded up robust features (SURF) [14], and

DAISY [179]. Salient points are not the only features that can be used for registering images. Mikolajczyk et al. [129] use affine-invariant regions to detect correspondences for wide baseline stereo matching. Matas et al. [123] detect maximally stable extremal regions to establish correspondences between a pair of images taken from different viewpoints. While this approach improves the registration accuracy in many cases, it has not been shown to fully handle the low overlap that occurs in the challenging data sets.

In this dissertation, we assume the existence of a dominant plane in the scene and subsequently employ it for the spatial registration. To segment the dominant plane in each frame, we employ the depth information of the scene. Therefore, in the following sections we briefly discuss the previous work about the disparity estimation and and dominant plane segmentation methods.

2.4 Disparity Estimation

Stereo matching is an active research area and an important computer vision problem. The goal of the stereo matching problem is to obtain an accurate depth representation of a scene from a stereo image pair. The term disparity was first used in the human vision literature in order to describe the difference in location of corresponding features seen by the left and right eyes [174]. Disparity is often treated as synonymous with inverse depth [163] in computer vision. A wide variety of computational models including area-based approaches, feature-based

methods, and energy minimization-based approaches have been proposed to solve the problem of disparity estimation of a scene from the left and right images of a stereo pair.

2.4.1 Area-based Methods

Area-based disparity estimation methods [140] employ the assumption that disparities within a neighborhood of a pixel in each image are constant; therefore, the intensity distribution within the area can be used to find the corresponding pixels in the other image using the photometric similarity and spatial consistency. The drawback of this approach is that the disparity map becomes sparse because the matching process does not consider any distinctive feature points, which are very crucial for accurate estimation of dense depth fields. Furthermore, area-based stereo methods assume that the disparities are equal for all the pixels in a matching window. This results in the blurring effect in object boundaries and causes the removal of small details.

2.4.2 Feature-based Approaches

Feature-based methods establish initial correspondences between among feature points extracted from the images. Edges, gradient peaks, line segments, and curves are usually selected as the feature points because they are the most prominent parts of the scene [193]. The main advantage of using features for the matching

stems from their robustness against photometric variations and the noise. Major drawbacks of this approach are the sparseness of the estimated disparity map and the error propagation from the errors caused by the feature mismatches.

2.4.3 Energy-based Disparity Estimation

Energy-based disparity estimation methods make use of the intensity at a single pixel coupled with energy minimization formulation for the matching. These methods suffer from three problems: 1) slow convergence, 2) computational load involved in solving partial differential equation, and 3) local minima problem. On the other hand, the energy-based disparity estimation methods tend to yield dense disparity fields with high accuracy. Early studies in this category utilize the epipolar constraint to convert the two-dimensional matching problem to a one-dimensional matching problem, which can be solved efficiently using dynamic programming [65]. Wei et al. [193] propose a stereo correspondence method by minimizing intensity and gradient errors simultaneously. Different from the conventional use of image gradients, they use the gradient in the deformed image space. In order to avoid local minima, they propose to parameterize the disparity function by hierarchical Gaussians. In the last decades, graph cuts have emerged as a powerful optimization technique for the minimization of energy functions. Very first work on the graph cuts was proposed by Roy and Cox [156]. They applied it to an N-camera stereo correspondence problem. Roy and Cox presented that the global minimum of a certain type of two-dimensional energy function

could be computed using the graph cuts. Nonetheless, their formulation does not allow the sharp discontinuities in disparity and yields poor results at the object boundaries. Boykov et al. [24] propose an alternative approach for the minimization known as the multiway cut. This approach finds a provably good local minimum and preserves the sharp discontinuities. However, its disadvantage is that it cannot guarantee to find the global minimum. Kolmogorov and Zabih [105] employ fast energy minimization algorithms based on graph cuts, which have the ability of avoiding the problems due to the local minima. This method can be applied to the cases where the intrinsic and extrinsic camera parameters are not provided. Kolmogorov and Zabih directly compute the disparity map from the gray-level image intensities without dealing with any intermediate process, such as rectification. Another popular approach to the depth analysis is to formulate the correspondence problem in the scope of segmentation. Before the matching, the image is segmented using monocular cues. Then, the correspondences between the layers is determined. These methods [18, 20] assume that the scene can be decomposed into a small set of layers with few parameters. They use an expectation-maximization algorithm to iteratively segment the image into regions of common transformation. Hong and Chen [91] first segment each image individually using a color-based mean shift algorithm. Then, they proceed the disparity estimation step. These techniques are quite successful on untextured regions and color images; however, they tend to fail on textured gray-level images because of the difficulty of the monocular segmentation.

2.5 Dominant Plane Detection

A dominant plane is defined as a planar region which occupies the largest area in the image observed by a video camera. The detection of the dominant plane is a preprocessing step to a wide variety of vision tasks, such as camera self-calibration, feature matching, image mosaicing, obstacle detection, object recognition, and scene analysis. This usually precedes the exploitation of constraints imposed by planarity.

Existing methods for the dominant plane segmentation are typically based on the extraction and matching of salient geometric features from images. Se and Bredy [164] propose a dominant plane disparity model which is formulated as a planar map linear to the image coordinates. They use a set of pixel coordinates and disparity values to generate the dominant plane disparity map. Then, they employ the iterative random sample consensus method [62] to estimate optimal parameters for the dominant plane from disparity values, which is robust to the obstacles in the scene. Lourakis et al. [116] propose a method for detecting the dominant plane in a scene using line features and a set of matched points. The proposed method searches for homographies using an iterative voting model based on point and line feature correspondences without requiring camera calibration. The disadvantage of this method is that it relies upon the availability of a set of corresponding points and lines extracted from a pair of stereo images. Yang et al. [198] describes two methods for planar segmentation based on integrating the image point coordinates in a higher dimensional real of complex plane. The

advantage of this approach is that a closed-form solution is presented; however, they are limited to two views. Ohnishi and Imiya [138] tackle the problem of segmenting the dominant plane using optical flow from a sequence of images captured by a moving robot for the purpose of navigation. They describe that the points on the dominant plane between consecutive images could be tracked using an affine transformation. After the affine transform parameters are computed, they are used to estimate the dominant plane motion between the images. The difference between the estimated planar flow and optical flow fields enables them to segment the dominant plane region using the matched flow vectors. Chumerin and Hulle [45] describe an approach for disparity plane estimation and subsequently use it to segment the dominant plane for a calibrated stereo camera system. They use a predefined road mask on the disparity map to filter out the pixels which belong to sky and regions above the ground plane. Then, they try to fit a linear plane model to the masked disparity map using am iterative weighted least squares regression method. The main disadvantage of this method is that it relies heavily on the accuracy of the disparity map estimation, while they do not provide any disparity map refinement model. Cherian et al. [43] present an approach that uses monocular cues for the scene depth estimation. They utilize the assumption that the depth of an object in the scene can be approximated using the distance between the point at which the object connects to the ground and where the camera is located. The proposed method first reconstructs the three-dimensional depth map of the scene using the Markov random field and then smoothes it based upon the principal component analysis. Finally, segmentation of the image

is performed using texture features in order to find the boundaries of the dominant plane. Once the dominant plane is segmented, they use the intrinsic and extrinsic parameters of the camera to build a three-dimensional coordinate system for the image. The drawback of this method is that the systems needs be to trained for the parameter estimation.

2.6 Image Change Detection

In image processing and computer vision literature, the term change detection has been used to refer to different problems [148]. For example, the scene change detection methods deal with the problem for determining the frame at which an image sequence switches between scenes [205]. In this dissertation, we address the problem of determining the set of pixels which are pictorially different between spatiotemporally aligned images of the same scene captured at different times. The change between the images may be the result of a combination of different external factors, such as appearance or disappearance of objects, motion of objects relative to the background, and appearance changes of objects. The core requirement is that the change detection framework should be able to discriminate between the relevant changes and the unimportant forms of change that may be caused by the camera motion, parallax, sensor noise, and illumination variation.

Over the years, many standard image processing techniques have been leveraged to find a solution to the problem of change detection in images of a scene

acquired at different instances. Different change detection algorithms have their own advantages, and no single approach is optimal and applicable to all cases. There are plenty of change detection approaches attempting to exploit and compare domain specific properties of the images, ranging from video surveillance [131] to video coding [9], object tracking [120], monitoring the earths surface [17], and motion estimation [139]. Previous literature has shown that we can classify existing change detection techniques into three main categories: 1) pixel-based methods, 2) background modeling, and 3) hypothesis testing and predictive models. In the following sections, we will present these categories.

2.6.1 Change Detection Using Pixel-based Features

Because of its algorithmic simplicity, image differencing is the most popular method used for various applications [17, 38, 124] involving change detection. The apparent first step is to compute the absolute values of the difference between the corresponding pixels in two images. It is usually followed by a thresholding operation to indicate regions of change in a binary change mask. Nevertheless, choosing a proper value for the threshold is a very critical and data-dependent task. A too low value may overwhelm the difference map with pixels which are labelled as false positives. On the other hand, a too high threshold value may suppress salient changes and result in a large number of false negatives. Furthermore, the threshold value may depend on the scene and viewing conditions that may change over time. This requires the dynamic calculation of the threshold based on the

image content because selecting an empirical threshold value is not considered as proper for a robust autonomous vision system. Local thresholding can also improve change detection particularly when the scene illumination varies locally over time. An extension of change detection using pixel-based feature is change vector analysis method [40, 194, 208], which is developed for multispectral remote sensing images using a similar approach. A feature vector is first generated for each pixel by using the several spectral channels. Then, thresholded difference between two feature vectors at each pixel is used to estimate regions of change. Image ratioing [55] is another image differencing related technique that uses the ratio instead of the difference.

Pixel-based change detection methods are computationally efficient because only the pixel intensity is processed. Nevertheless, they are extremely sensitive to even minor image registration artifacts and illumination changes because they do not consider local structural information. Hence, scalability of pixel-based methods is limited, especially in the context of real-world applications with complex and dynamic scenes.

2.6.2 Background Modeling

In the context of detecting regions of change among the consecutive frames of a single video, background modeling is considered as a substep of the change detection problem [143]. The goal to determine which pixels belong to the scene background and should not be classified as change. The entire frame sequence

in the video is used as the basis for making decisions about change, in contrast to comparing a pair of images in an ordinary change detection problem. These methods are usually limited to the case of static or slowly varying background. The most common approach is to build a model for illumination changes and minor variations using Gaussian mixture models [109, 211]. The probability of observing an intensity value at a pixel location is modelled as the weighted sum of multiple Gaussian distributions. Gaussian mixture models are usually built on pixel-based structures and work well in detecting changes that can be described as independent events. In adaptive approaches, the mean and variance of the Gaussian are updated using simple adaptive filters to accommodate changes in lighting or objects that become part of the background. However, the background modeling tends to fail to model complicated change patterns that may be correlated in a spatiotemporal volume.

2.6.3 Hypothesis Testing and Predictive Models

Pixel-based methods have the advantage of the algorithmic simplicity, but region-based techniques yield results more robust to noise. A basic region-based approach is to compare if statistics of a specific region between two images have the same intensity distribution. The decision rule for the change is modeled as a statistical hypothesis testing. In order to make the change detection more reliable, decisions are made based on a set of differences inside a small window instead of a single pixel. The decision as to whether or not a change has occurred at a given

pixel corresponds to choosing one of two competing hypotheses: no-change and change [148]. Images to be compared are considered as random vectors. Knowledge of the conditional probability distributions allows us to choose the hypothesis that describes the intensity deviation at pixel [2, 3]. Rignot and van Zyl [153] propose hypothesis tests on the ratio and difference images of SAR data assuming intensities could modeled by a gamma distribution. Bruzzone and Prieto [26] present that using estimated variance values in the decision rule may increase the false alarm rate. Instead, they propose an automatic change detection technique that estimates the parameters of the mixture distribution from the difference image. Black et al. [26] describes an innovative approach, where they softly classify the pixels into mixture components corresponding to different generative models of change: 1) parametric object or camera motion, 2) illumination phenomena, 3) specular reflections, and 4) pictorial changes. Pixels which are poorly described by any of the four category are labeled as outliers. This algorithm uses the optical flow field between the images along with the expectation-maximization algorithm [132] to assign of each vector to one of the classes. Exploiting the relationships among the neighboring pixels has been shown to improve the change detection accuracy. A well-known approach is to divide the image into blocks and fit the intensity values in each block to a polynomial function of the pixel coordinates. Hsu et al. [92] discusses generalized likelihood ratio tests using constant, linear, or quadratic models for the blocks. Skifstad and Jain [167] improves Hsu's intensity modeling by developing an illumination-invariant model. They use partial derivatives as the change decision criterion.

Texture is a measure of the intensity variation of a region, and it quantifies properties, such as recurrence, smoothness, and regularity. Visual experiments show that the texture feature remains relatively stable with respect to noise and illumination changes. From this point of view, textural features for the change detection have been extensively investigated in different applications [15, 126, 149, 168]. Texture requires a processing step to generate the descriptors. There are various texture descriptors, such as Gabor filters, wavelet transforms, linear predictors, eigen filters, and several encoding methods [108, 128, 201]. Yokoi [203] proposes a texture-based change detection method combined with background learning. They use a ternary code for encoding the intensity difference between pixels in the texture. Miezianko and Pokrajac [126] propose a framework that is based on wavelet decomposition of localized texture. Objects left in the scene and objects moved or taken from the scene are considered as the change in the background. The main disadvantage of the texture-based change algorithms is that when the scene and the object share homogeneous regions, the texture difference measure will fail. Therefore, incorporating intensity and texture differences improves the robustness of the change detection framework. Li and Leung [112] proposes a change detection algorithm using a weighted combination of the intensity difference and a texture difference measurement.

Chapter 3

Spatiotemporal Alignment

Our goal is to detect regions of change between two videos of the same scene recorded at different times. In such a case, we may need to deal with videos that are neither temporally nor spatially aligned. To properly compare the videos, we first need to establish correspondences between the frames of the videos, in the sense of having the most similar view of the same scene. Namely, one needs to estimate the time instances that images of the same scene are captured from similar viewing points. After the temporal alignment, we can proceed to spatially align the synchronized frames of the two videos. To this end, in this chapter we propose temporal and spatial alignment methods that are necessary before the change detection analysis.

3.1 Temporal Alignment

With the advent of imaging techniques, using large sets of images and videos has become essential in computer vision and image processing applications [16, 35, 68, 73, 200]. This rapid increase in the amount of visual data introduces new challenges, such as temporal and spatial alignment of the data from different sources. When a pair of corresponding frames from different videos is provided, synchronization of the rest of the frames is a relatively easy task because of time and space continuity constraints. Nevertheless, the initial temporal correspondence between the asynchronous videos is not available in many applications. To overcome this limitation, different methods with specific constraints have been presented in the literature. Many solutions rely on the assumption that the temporal correspondence can be formulated as a linear function of time [35], or there is a constant time offset between the videos [141]. Various systems require the use of additional hardware, such as GPS [56]. A few studies allow the use of mobile platforms, but they require to follow almost the same trajectory [160]. Our goal is to find ways to relax these constraints and to design a synchronization method that can be applied to more flexible video acquisition scenarios. We have two main motivations: to develop a vision-based method for the synchronization of videos recorded by mobile platforms and to eliminate the need for examining all the frames (or inclusion of any kind of prior information) for the initial temporal correspondence. Majority of the existing solutions rely either on prior information or additional

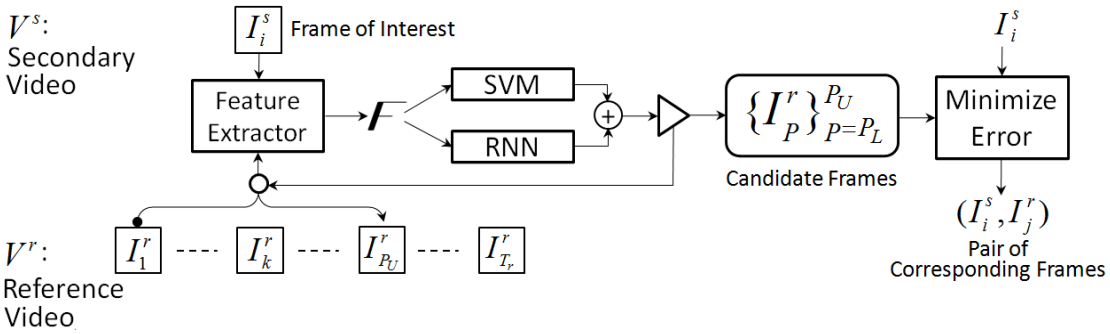


Figure 3.1: Block diagram of the proposed temporal alignment method. V^s and V^r are videos of the same environment recorded at different times. Given the frame I_i^s in V^s, the goal is to find its corresponding frame I_j^r, that is the frame having the most similar view of I_i^s in V^r. We examine V^r, and if the SVM and RNN based hybrid one-class learner detects a potential corresponding frame, it assesses a high similarity score. Frames having similarity scores greater than a threshold constitute the pool of candidate frames. $I_{P_L}^r$ and $I_{P_U}^r$ are the first and the last frames in the pool, and T_r ($T_r \gg P_U - P_L$) is the total number of the frames in V^r. The exact corresponding frame pair is determined by minimizing the similarity error in the pool.

hardware to avoid an exhaustive search for the initial match. It would be beneficial to have a method providing the initial match in an automated manner. To this end, we cast the video synchronization problem to a one-class classification problem.

Let us assume that we are given two frames from different videos of the same environment captured at different times, we propose a hybrid one-class learner that can assess a similarity score based on the visual features between two frames from the videos by combining the outputs of a Replicator Neural Network (RNN) and Support Vector Machines (SVM) (Figure 3.1). Using the output of the learner, we select potential corresponding frames and perform a secondary search to find the

exact match by minimizing the similarity error. The advantage of this approach is that it does not rely on any initial guess or assumption. Different from the previous studies, our system anticipates the possibility that when two asynchronous videos are given, the videos may contain frames that do not have a corresponding pair in the other one.

In the following sections, we use the terms *video synchronization* and *temporal alignment* interchangeably to address the same problem.

3.1.1 Temporal Alignment as a One-class Learning Problem

Given two videos of the same environment recorded at different times, we call one of them *reference video* and the other *secondary video* (denoted by V^r and V^s, respectively). The goal of the temporal alignment problem is to find a mapping set $\mathfrak{V}$:$\mathrm{V}^s \rightarrow \mathrm{V}^r$, providing the pairs of corresponding frames between the videos V^r and V^s. We here address a more general case of this problem where the videos are recorded by mobile platforms following unknown trajectories.

For a synchronized reference-secondary video pair, $\mathfrak{V}$ is a set of temporally aligned frame pairs: $\mathfrak{V} = \{(I_i^s, I_j^r)\}_{i=1}^{T_s}$. I_i^s is the i^{th} frame in V^s, I_j^r is the jth frame in V^r, and T_s is the total number of the frames in V^s. I_i^s is called *frame of interest*, and I_j^r is called *corresponding frame*, having the most similar view of I_i^s in the reference video V^r. In our acquisition scenario, because the mobile platform

carrying the camera follows unknown trajectories, some frames in V^s may not have corresponding frames in V^r. In the literature, the idea of using predictive modeling to detect the changes in remote sensing images is presented in some studies [46, 68]. We follow a similar approach and propose the idea that pairs of corresponding frames in the mapping set $\mathfrak{V}$ should be instances of a class called *similar enough*. If we formulate the relationship among these instances using a d-dimensional feature vector, each pair in $\mathfrak{V}$ can be represented by a point in the d-dimensional feature space (Figure 3.2). Because these points belong to the class *similar enough*, they should constitute a dense region in the space. We can model this region and train a one-class learner, which finds a boundary that separates in-class instances from out-of-class instances. The learner can then be used to produce a similarity score for a given pair of frames. If the similarity score is greater than a threshold, the pair is classified as potential corresponding frames.

3.1.2 Feature Selection

Experience shows that humans performing manual image retrieval or video synchronization tend to focus on the coarse details (e.g., a large field or a building) in the images first [186]. After finding potential corresponding images, they look at the fine details for the final matching. This observation is the main inspiration leading to selection of the discrete cosine transform (DCT) as one of the feature extraction methods. DCT is widely used in image compression and retrieval applications because of its superior energy compaction property and low

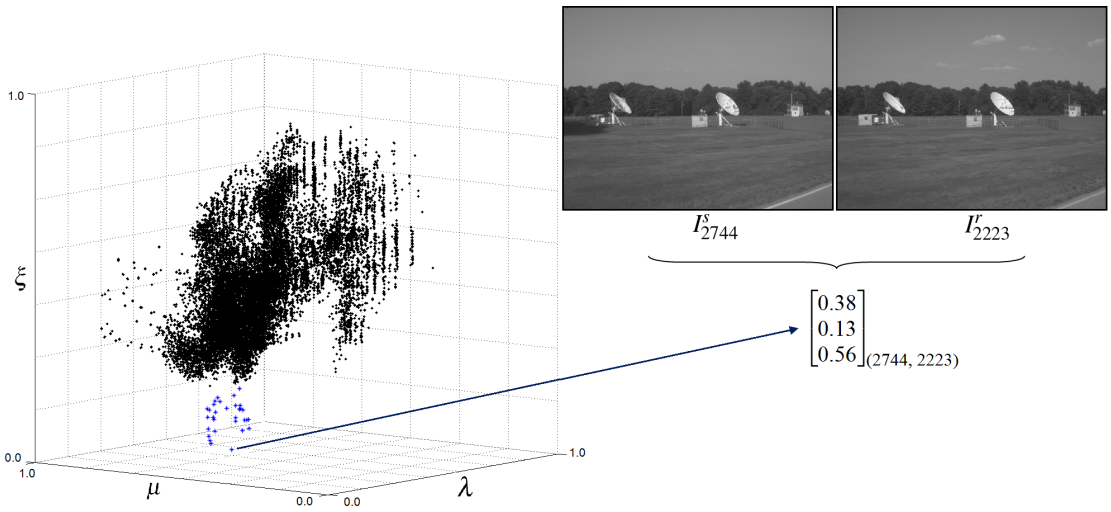


Figure 3.2: Illustrative example of a three-dimensional feature space. μ and λ represents features obtained using discrete cosine transform, and ξ is the gradient-based feature. These features are extracted from a reference-secondary video pair provided. The black points in the figure refer to feature values extracted from unmatched pairs of frames. On the other hand, the blue asterisks refer to the feature vectors extracted from pairs of corresponding frames. One of the corresponding frame pair, (I^s_{2744}, I^r_{2223}) and the value of its feature vector $[\mu \ \ \lambda \ \ \xi]^T$ are presented.

computational complexity. In these applications, the image is usually divided into blocks before computing DCT, and DCT is said to be sensitive to changes in the viewing angle. We here follow a different approach [206] and apply DCT to the entire image as a global descriptor. The resulting DCT matrix has the same size as the image. Nevertheless, it preserves the most important image characteristics in a small subset of the coefficients, while majority have very small magnitudes and the DC (i.e., direct current) component is simply ignored. The remaining coefficients represent distinctive coarse details. This can then be used to initialize

frame matching efficiently. This approach overcomes the problems resulting from the sensitivity of DCT to the viewing angle. To investigate the applicability of DCT as a global descriptor, we used the *Amsterdam Library of Object Images* [66]. This is an image database of a thousand objects, recorded under systematically varied viewing angles. We compute DCT-based feature μ (introduced below) for a subset of images in the database. For each object, we use the neutral image of the object (angle= $0\,^\circ$) and the images of the same object captured by different viewing angles (within the bound of [$-45\,^\circ$,$+45\,^\circ$]). Then, we compute the unbiased variation coefficient of the DCT-based feature values:

$$\widehat{c}_v^* = (1 + \frac{1}{4 * N_\alpha}) * (\frac{\sigma}{m}) \times 100\%, \tag{3.1}$$

where N_α is the number of viewing angles, m and σ are mean and standard deviation of the feature values. $\widehat{c}_v^*$ is a normalized measure of the dispersion of a distribution, and a low $\widehat{c}_v^*$ denotes a small extent of dispersion. We use $\widehat{c}_v^*$ to quantify precisely the amount of dispersion caused by the change in viewing angles. The average value of $\widehat{c}_v^*$ for the DCT-based feature is about 4% even though the viewing angle varies significantly.

In the last decade, scale-invariant feature transform (SIFT) keypoints [117] have been widely used for different applications which require the feature matching due to its superior reported results. The matching process is performed by individually comparing each feature from a secondary image to the features of a

reference image and finding candidate matching features based on Euclidean distance of the feature vectors. Nonetheless, SIFT has high computational complexity. For instance, we observe that for the same input images of the size 576x460, the matching process with SIFT usually takes about 30 times longer than DCT counterpart, while both yield very similar or the same result.

Let I be a gray-level image of the size $N \times M$, and let us denote two-dimensional DCT of I as DCT(p, q) [206], where p=0,...,M-1 and q=0,...,N-1. This yields a real valued DCT coefficient matrix, and it is a fast transform because of its linear separability. To decrease the sensitivity of DCT to illumination changes, we apply median filter and map the intensity values so that about 1% of data is saturated at low and high intensities of I. While it is not necessary, a square DCT matrix may be preferable for a symmetric spectrum. The total amount of low energy coefficients that will be neglected should be decided carefully because keeping too much energy may result in false matches. Let us denote the DCT matrix, which is thresholded to remove insignificant coefficients, as DCT$'$. Given two images I_1 and I_2 to be compared, our first feature μ is computed using $DCT_1^{'}$ and $DCT_1^{'}$ matrices. We first compute the absolute difference of $DCT_1^{'}$ and $DCT_2^{'}$ as follows

$$\Delta DCT_{1-2}^{'} = |DCT_1^{'} - DCT_2^{'}|. \tag{3.2}$$

Then, we compute the mean of non-zero elements in $\Delta DCT_{1-2}^{'}$

$$\mu(I_1, I_2) \triangleq \frac{1}{\eta} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \Delta DCT'_{1-2}(p, q) \tag{3.3}$$

where η is the number of non-zero elements in $\Delta DCT'_{1-2}$. We use a slope function to limit maximum value of μ in order to normalize it to the range $[0, 1]$. The normalization plays a crucial role in SVM and RNN training. Our second DCT-based feature relies on the number of the high energy coefficients that appear in both DCT matrices of I_1 and I_2. Let L represent the number of such frequency components. We define our second feature as $\lambda = \frac{L}{\mathfrak{N}}$, where $\mathfrak{N}$ is the normalization factor.

As a spatial domain descriptor, we use local intensity statistics. Intensity gradient [112] is known to be less affected by changes in the illumination. Magnitudes of intensity gradients of I_1 and I_2 are stored in G_1 and G_2 using the Sobel operator. We define our third feature ξ as being the mean absolute difference of G_1 and G_2:

$$\xi(I_1, I_2) = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} |G_1(p, q) - G_2(p, q)|. \tag{3.4}$$

Similar to μ, ξ is normalized to the range $[0, 1]$. For problems dealing with videos, the idea of exploiting the temporal consistency among the consecutive frames is a well-known factor improving the robustness of the system. Given a pair of *frame of interest* and *corresponding frame* (I_i^s,I_j^r), we expect that the neighboring frame pairs (I_{i-1}^s,I_{j-1}^r) and (I_{i+1}^s,I_{j+1}^r) should also exhibit similarity. Accordingly, the

feature vector $F_{(i,j)} \in \mathbb{R}^9$ for the frame pair (I_i^s,I_j^r) is defined as:

$$F_{(i,j)} = [\mu_{-1} \;\; \mu_0 \;\; \mu_{+1} \;\; \lambda_{-1} \;\; \lambda_0 \;\; \lambda_{+1} \;\; \xi_{-1} \;\; \xi_0 \;\; \xi_{+1}]^T, \tag{3.5}$$

where μ_ℓ=$\mu(I_{i+\ell}^s,I_{j+\ell}^r)$, λ_ℓ =$\lambda(I_{i+\ell}^s,I_{j+\ell}^r)$, and ξ_ℓ=$\xi(I_{i+\ell}^s,I_{j+\ell}^r)$ for ℓ=$\{-1,\ 0, +1\}$.

3.1.3 One-class Learner

We propose to model the similarity relationship between images in *9-dimensional feature space* and to consider dissimilar images as anomalies or outliers. These outliers constitute the negative class called *not similar*. It is not feasible to sample all the negative examples properly. In the literature, out of the approaches tackling the problem of one-class learning using only positive examples, the statistical models and neural network based methods are the most widely used and successful ones [88]. We propose a hybrid one-class learner incorporating Support Vector Machines and Replicator Neural Network. Following a hybrid approach exploits the advantages of multiple methods and overcomes their weaknesses and deficiencies [177].

3.1.4 Support Vector Machine

SVM is a kernel-based maximum margin method that allows the model to be described as a sum of the influences of a subset of the training examples [8]. SVM

has been initially applied to two-class classification problems and then extended to the multiple-class classification problems. One-class SVM is used a tool to estimate regions of high density in a feature space. In such a case, the goal is to find a boundary separating volumes of high density from volumes of low density. The estimated boundary is employed to detect the outliers. SVM kernel defines a hyperplane separating in-class instances from the others according to its notion of similarity. We select a radial-basis function (RBF) as the kernel [39]. RBF kernel is defined as

$$K(x, x') = exp(-\gamma \|x - x'\|^2), \tag{3.6}$$

where x is the center, $\gamma = 1/s^2$, and s is the radius of the kernel. γ and the fraction factor $\nu \in (0, 1)$ are the parameters of the learner. By tuning ν, one can control the fraction of support vectors. Following the SVM training approach proposed in [39], values of the parameters are found as $\gamma = 0.33$ and $\nu = 0.47$.

3.1.5 Replicator Neural Network

RNN is a multi-layer artificial neural network and trained in such a way that the input values are reconstructed at the output. Its effectiveness for one-class classification is presented by Hawkins et al. [85]. We use a replicated feed forward neural network with one hidden layer, 10 neurons in the hidden layer, with hyperbolic tangent sigmoid as the transfer function. The number of neurons in the hidden

layer is chosen experimentally to minimize the reconstruction error. Input layer has 9 units, and it is fed with the 9–element feature vector F. Accordingly, the output layer has 9 neurons with linear transfer function (Figure 3.3).

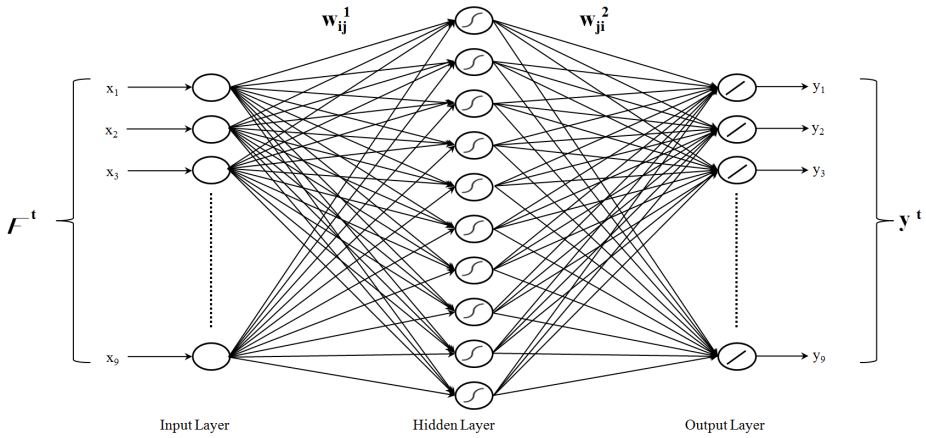


Figure 3.3: Illustrative example of the RNN used in the one-class learner. F^t is the 9-dimensional feature vector for the training sample t. x_i and y_i, $i = 1, \ldots, 9$ are the input and output units, respectively. w_{ij}^1 is the weight of the connection from the input x_i to the hidden unit j in the first layer. Similarly, w_{ji}^2 is the weight of the connection from the hidden unit j to the output unit y_i in the second layer. During the training, the weights w_{ij}^1 and w_{ji}^2, $i, j = 1, \ldots, 9$ are adjusted to minimize the mean reconstruction error for all training patterns. Eventually, RNN generates an implicit and compressed model of the training data. Then, an input that is correlated to the training samples is expected to be reconstructed at the output with with a low reconstruction error.

In the online learning step of RNN, we update the weights according to gradient descent with momentum to avoid large oscillations. The momentum parameter is set to 0.6, the learning rate is set to 0.3, and number of epochs is 200.

3.1.6 Combining Multiple One-class Learners

Manually generating a complete mapping set $\mathfrak{V}$ of temporally aligned frames for a reference-secondary video pair is not always feasible and not required for our method. Instead, we can use a small set of corresponding frames for the training purposes. Let χ denote the training set with $\chi = \{(I_i^s, I_j^r)^t\}_{t=1}^{\mathfrak{N}}$, where $\mathfrak{N}$ is the number of the pairs in the training set.

In our hybrid classifier, we use SVM and RNN as the base learners, and they work in parallel (Figure 3.4). First of all, we compute the training feature vectors $F_{(i,j)}^t$ for $t = 1, \ldots, \mathfrak{N}$ where i and j are the frame indices. Let y_{svm} be the output of the one-class SVM learner. y_{svm} is the sum of the influences of support vectors given by the RBF kernel. When SVM is used as a standalone classifier, the output is usually transformed into probability values, and SVM predicts the class label. In our one-class setting, we use the SVM output without transferring it into another form, and we do not rely on the output class labels. SVM produces the largest output values for the in-class inputs. Concurrently, we feed $F_{(i,j)}^t$ to the RNN. Let y_{rnn} be the output of the RNN learner. y_{rnn} is the mean reconstruction error of the network. The trained RNN is expected to reproduce in-class feature vectors with a small reconstruction error. Accordingly, feature vectors representing out-of-class instances result in large reconstruction errors.

We combine one-class learners by the weighted voting method. Let v_1 and v_2 be the weights of the votes of SVM and RNN outputs in the hybrid learner. We denote the normalized outputs as $\overline{y}_{svm}$ and $\overline{y}_{rnn}$. We combine them by taking a

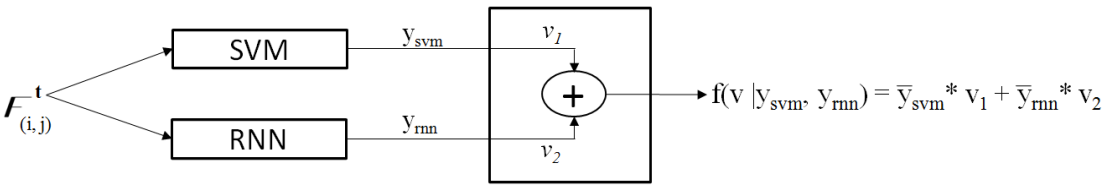


Figure 3.4: Illustrative example of combining the base learners SVM and RNN by the weighted voting method. $\overline{y}_{svm}$ and $\overline{y}_{rnn}$ are the normalized output values. Value of the weights v_1 and v_2 indicate the influences of each base learner to the hybrid framework.

linear combination of the votes:

$$y = \overline{y}_{svm} * v_1 + \overline{y}_{rnn} * v_2. \tag{3.7}$$

Normalization is required because the outputs are not posterior probabilities [98]. Developing a one-class learner requires a threshold which separates in-class instances from out-of-class instances. Let us denote the decision threshold as ϵ. If the output of the combined learner is smaller than ϵ, the input is labeled as in-class instance. Our goals during the combining process are: i) to *minimize* combined output value y and ii) to *maximize* the value of the decision threshold ϵ under the constraints:

$$v_1 + v_2 = 1, v_1 > 0, v_2 > 0, \text{ and } v_2 \geq v_1. \tag{3.8}$$

We start with equal weights of SVM and RNN votes. By the minimizing the synchronization error on the training data, the value of v_1 is determined as $v_1 = 0.43$, the value of v_2 is determined as $v_2 = 0.57$, the value of decision threshold ϵ

is determined as $\epsilon = 0.29$.

3.1.7 Synchronization Algorithm

Given a pair of reference-secondary videos and a frame of interest I_i^s in V^s, we use the algorithm below to find its corresponding frame I_j^r in the reference video V^r. The proposed temporal alignment algorithm is presented in Algorithm 1. Synchronization after the initial match is performed in a local search manner using a window surrounding the previous estimate following the same approach.

Algorithm 1 Video Synchronization

Initial Match:
Set the loop condition variable *similar* to $false$.
Set the threshold ε to *0.29*.
Set the frame index t of the reference video V^r to *1*.
▷ Generate the Pool of Potential Corresponding Frames
while t is less then T_r and *similar* is $false$ **do**
  Calculate $F_{(i,t)}$ for the current frame pair (I_i^s, I_t^r)
  Get the output y_t of the hybrid one-class learner
    for the input $F_{(i,t)}$
  if y_t is less than ε **then**
    Set the candidate frame pool counter p to 1
    repeat
      Get the output y_{t+p} for the input $F_{(i,t+p)}$
      Increase the value of the counter p
    until $t+p$ is greater than T_r *or*
        y_{t+p} is greater than ε
    Store the frame pairs (I_i^s, I_P^r) for $P = t, ..., t+p-1$
    Set the variable *similar* to *true*
    ▷ Note that t corresponds to P_L and
      $t+p-1$ corresponds to P_U in Figure 3.1
  else
    Increase the value of t
  end if
end while
▷ Minimize the Similarity Error Using the Feature μ
for all frame pairs $(P = t, ..., t+p-1)$ in the pool **do**
  Calculate the feature μ_P
end for
Select the frame I_j^r that minimizes μ_P,
namely $\arg\min_{j \in P}\{\mu_P\}$,
▷ T_r is number of frames in V^r, $F_{(i,t)}$ is the feature vector, P_L and P_U are the indexes of the first and last frames in the candidate pool.

3.2 Spatial Alignment

We can reduce the problem of registering two videos to that of registering multiple images after the video synchronization is accomplished. A classical problem in computer vision is to align images of the same scene taken at different instances. The spatial misalignment between the two images of the same scene results from the fact that the two acquisitions may have different external calibration parameters. The registration process can remove the effects of these parameters.

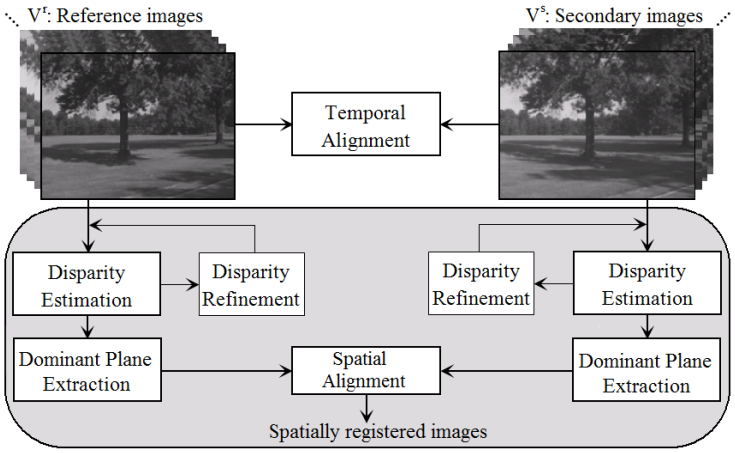


Figure 3.5: Illustration of the proposed spatial alignment method. Two videos (either stereo or monocular depending on the disparity estimation scheme) are first temporally aligned. The dominant planes are segmented using the disparity maps. We then register the dominant planes in the scene.

The residual pixels after the registration can be due to either areas of change or the parallax. Many change detection methods utilize the whole frame region for the spatial registration, whereas imposing the same transformation function for the whole scene, which contains different planes, may produce inadequate results. Instead, we propose a registration module (Figure 3.5) based on the assumption

that there is a dominant plane in the scene, and the relevant changes occurs on the dominant plane.

Employing the dominant plane for spatial image registration can overcome some difficulties related to the scene structure. Nevertheless, segmentation of the dominant plane still remains a challenging problem. We propose a *novel* approach that utilizes videos to extract depth information of the scene using stereo and monocular cues. Resulting disparity map and texture information of each disparity layer are combined to segment the dominant plane in the scene. Finally, extracted dominant planes from the images are used for the spatial alignment. In the following sections, we present the modules of the proposed spatial registration framework.

3.2.1 Disparity Information

The spatial displacement of corresponding points between the images of the same scene is called disparity. Disparity is inversely proportional to depth and determine a dense point-to-point correspondence. Disparity map of a scene can provide detailed content information that is less invariant to shadows and illumination changes compared to other vision-based features.

3.2.2 Depth Estimation using Monocular Cues

Disparity estimation is most widely performed using stereo imagery (i.e., disparity from binocular) [163]. Nonetheless, there are also various monocular visual cues, such as texture variation, focus, and gradients, that could be employed for the depth estimation from a single monocular image [51, 161, 162]. Saxena et al. [161] propose a supervised learning approach to the monocular disparity problem. They begin by collecting a training set of monocular images of outdoor environments and their corresponding ground-truth disparity maps. Finally, they apply supervised learning to predict the disparity map as a function of the image. Hoiem et al. [90] propose a method that divides image regions into geometric classes using an appearance-based learning model, which coarsely describe the three-dimensional scene orientation.

Another approach to the monocular disparity problem is structure from motion [52, 54]. Structure from motion is the process of estimating three-dimensional structures from a sequence of two-dimensional images which may be coupled with local motion. Geometric constraints along with sufficient and non-degenerate set of initial correspondences are used to construct the structure in uncalibrated images. Dellaert et al. [54] propose a novel way to solve the structure from motion problem without a priori correspondence information.

3.2.3 Estimation of Disparity Map from Binocular Video

Given two stereo frames $\mathbf{Fr}_i^r$ and $\mathbf{Fr}_j^s$ from temporally aligned reference and secondary videos, respectively. Our goal is to estimate the disparity maps $\mathbf{D}_i^r$ and $\mathbf{D}_i^s$ for $\mathbf{Fr}_i^r$ and $\mathbf{Fr}_j^s$ using the image pairs in the stereo frames. We need the disparity map to segment the dominant plane in the scene. Over the years, several approaches based on disparity differencing have been proposed to detect the regions of change directly from the disparity maps [13, 82, 83, 173]. In most cases, the distance between an object to be detected and the dominant plane (i.e., ground plane) in the scene is not large; therefore, when the object of interest is close to the background, there is a very good chance that the object is missed. This is a very common problem associated with the disparity-based change detection methods. In our setting, we follow a different approach and use the disparity maps $\mathbf{D}_i^r$ and $\mathbf{D}_i^s$ only for establishing spatial correspondences between the two stereo frames $\mathbf{Fr}_i^r$ and $\mathbf{Fr}_j^s$ of the same scene.

The necessity of accurate disparity map for spatial registration module lead us to employ a sophisticated method. The graph cut based energy minimization technique recently has attracted a lot of attention because of its optimality properties and the success of reported results [104]. Energy minimization approach processes the input images symmetrically and imposes spatial smoothness while preserving the discontinuities [104]. Let us assume that we are given a stereo pair of images of a scene, and both the intrinsic and extrinsic parameters of the camera are not available. In such a case, disparity map can be directly computed from

image intensities without dealing with any intermediate process such as image rectification [41]. In a stereo imaging system, two-dimensional projection of each physical point in the scene is represented in left and right images of the stereo frame. The only exception is the the occluded points. Accordingly, one pixel in one of the stereo image pair should correspond to at most one pixel in the other image. This is called as the uniqueness requirement [24]. Similarly, if a pixel does not have a corresponding pair, it is labelled as an occluded pixel. Nevertheless, in our disparity estimation setting, we do not specially treat the occluded pixels; instead, we assume that they are part of the closest disparity layer. In the literature, disparity estimation methods usually considers one of the images (either right or left) in a stereo frame as the reference image and compute the disparity based on the selected image [23, 155]. Kolmogorov and Zabih [104] presented that treating the left and right images in a stereo frame symmetrically is the only way to make full use of the information in both images. This is accomplished by taking both images into account while computing the cost of correspondences. In our disparity estimation module, we follow the symmetric approach and cast the correspondence problem to energy minimization problem using graph cuts.

We are given a stereo frame **Fr** which consists of two images called left image (denoted by I_L) and right image (denoted by I_R). Let P_L be the set of pixels in I_L, let P_R be the pixels in I_R, and finally let P_I be the set of all pixels: $P_I = P_L \bigcup P_R$. A pixel p_L in I_L is specified by its image coordinate pair (x_L, y_L). Similarly, a pixel p_R in I_R is specified by (x_R, y_R). Let us assume that p_R is the corresponding

pixel of p_L. In an ordinary correspondence problem, given the pixel p_L at (x_L, y_L) in I_L, the goal is directly to estimate the coordinate of the corresponding pixel p_R, (x_R, y_R) in I_R and then compute the disparity value $d_{(p_L)}$ for each pixel in I_L one by one. In the energy minimization approach, our goal is to first describe the disparity relationships among the pixels in terms of a cost function and then to minimize the total energy of the entire system simultaneously. Computing a strong local minimum for the energy function corresponds to implicit way of estimating the disparity map. Let d denote a correspondence configuration for the pixels in P_L and P_R with $\mathsf{d} : \{(p_i, p_j)\}_{i=1}^{N}$, $p_i \in P_L$, $p_j \in P_R$, and N is the number of the pixels in P_L. p_i and p_j are initially assigned in such a way that they are potentially corresponding pixels. The desired state for d is that d contains only pairs of pixels which indeed correspond to each other. Let us define an energy function (i.e., the cost function) for an arbitrary correspondence configuration d as follows:

$$\mathsf{E}(\mathsf{d}) = \mathsf{E}_{data}(\mathsf{d}) + \mathsf{E}_{smooth}(\mathsf{d}), \tag{3.9}$$

where the data term $\mathsf{E}_{data}(\mathsf{d})$ takes value based on the differences in intensity between corresponding pixels, and the smoothness term $\mathsf{E}_{smooth}(\mathsf{d})$ guarantees that neighboring pixels tend to have similar disparity values. We assume that disparity values can lie in a limited range,

$$d_{(p_i)} = (x_{p_i} - x_{p_j}, y_{p_i} - y_{p_j}), \tag{3.10}$$

where $0 \leq x_{p_i} - x_{p_j} \leq k_x$, $0 \leq y_{p_i} - y_{p_j} \leq k_y$; and k_x and k_y are user defined parameters. In this way, we anticipate the fact that the set of possible disparities could be two-dimensional. The data term $\mathsf{E}_{data}(\mathsf{d})$ is defined as

$$\mathsf{E}_{data}(\mathsf{d}) = \sum_{(p_i,p_j)\in\mathsf{d}} \mathfrak{D}(p_i, p_j), \tag{3.11}$$

where $\mathfrak{D}(p_i, p_j)$ is defined symmetrically [19] as follows

$$\mathfrak{D}(p_i, p_j) = \min\{D_L(x_{p_i}, x_{p_j}, I_L, I_R), D_R(x_{p_j}, x_{p_i}, I_R, I_L)\}, \tag{3.12}$$

$$D_L(x_{p_i}, x_{p_j}, I_L, I_R) = \max\{0, I_L(x_{p_i}) - I_{max}, I_{min} - I_L(x_{p_i})\}, \tag{3.13}$$

$$I_{max} = \max\{\frac{1}{2}(I_R(x_{p_j}) + I_R(x_{p_j} - 1)), \frac{1}{2}(I_R(x_{p_j}) + I_R(x_{p_j} + 1)), I_R(x_{p_j})\}, \text{ and} \tag{3.14}$$

$$I_{min} = \min\{\frac{1}{2}(I_R(x_{p_j}) + I_R(x_{p_j} - 1)), \frac{1}{2}(I_R(x_{p_j}) + I_R(x_{p_j} + 1)), I_R(x_{p_j})\}. \tag{3.15}$$

$D_R(x_{p_j}, x_{p_i}, I_R, I_L)$ is computed using the symmetric counterparts of the Equations 3.13, 3.14, and 3.15. In the Equation 3.9, the smoothness term $\mathsf{E}_{smooth}(\mathsf{d})$ imposes a penalty if two neighboring pixels have different disparity values. $\mathsf{E}_{smooth}(\mathsf{d})$ is defined as follows

$$\mathsf{E}_{smooth}(\mathsf{d}) = \sum_{(p_{i_1}, p_{i_2}) \in \mathfrak{N}} \mathbf{V}(p_{i_1}, p_{i_2}), \tag{3.16}$$

where $\mathfrak{N}$ is a neighborhood relationship and $\mathbf{V}(p_{i_1}, p_{i_2})$ is the penalizing factor. $\mathbf{V}(p_{i_1}, p_{i_2})$ is zero if neighboring pixels (p_{i_1}, p_{i_2}) have the same disparity, and if not, $\mathbf{V}(p_{i_1}, p_{i_2})$ has a positive value given by

$$\Delta d_{max} = \max(|I_L(p_{i_1}) - I_L(p_{i_2})|, |I_R(p_{i_1} + d_{p_{i_1}}) - I_R(p_{i_2} + d_{p_{i_2}})|) \text{ and} \tag{3.17}$$

$$\mathbf{V}(p_{i_1}, p_{i_2}) = \begin{cases} \sigma_{low} & \text{if } \Delta d_{max} < \Delta I \\ \sigma_{high} & \text{otherwise} \end{cases}$$

where σ_{low} and σ_{high} are empirically selected penalty coefficients, ΔI is a factor controlling the boundaries, $p_{i_1} + d_{p_{i_1}}$ is the corresponding pixel of p_{i_1} in I_R, and $p_{i_2} + d_{p_{i_2}}$ is the corresponding pixel of p_{i_2} in I_R.

Once the energy function $\mathsf{E}(\mathsf{d})$ is defined, one can minimize it by simulating all possible correspondence configurations in a brute force manner. If there are N pixels in P_L, the number of all possible configurations is $N!$ by enforcing the uniqueness requirement. This is not a feasible approach and requires enormous computational costs because a middle-size stereo frame consists of about one million pixels. Instead, we perform the minimization by using the graph cuts. We follow the method that Kolmogorov and Zabih [104] proposed to construct the graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, whose nodes (i.e., vertices) are image pixels, and whose edges

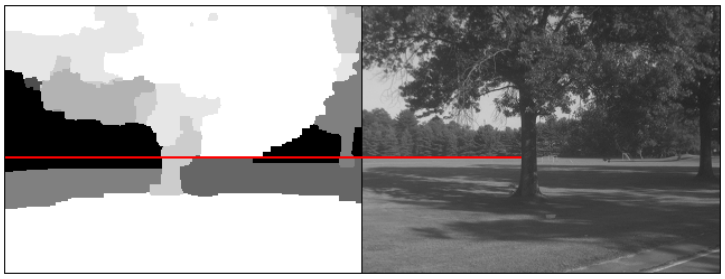
have weights obtained from the energy term. Cost of a cut is the summation of the weights of the edges, and the minimum cut problem deals with finding the cheapest way to cut the edges. The minimum cut on G results in a correspondence configuration which minimizes the energy term. This is achieved by $\alpha-$expansion algorithm [24]. Given a correspondence configuration $\mathsf{d}_{current}$ and a disparity value α, a transition from the configuration $\mathsf{d}_{current}$ to a new configuration d_{new} is called $\alpha-$expansion if any set of pixels changes their disparity labels to α. The $\alpha-$expansion algorithm is an iteration of transitions that generates local improvements for different disparity values α. The iteration continues until no more $\alpha-$expansion reduces the energy. The steps of the $\alpha-$expansion procedure is explained in Algorithm 2.

Algorithm 2 $\alpha-$expansion [24]

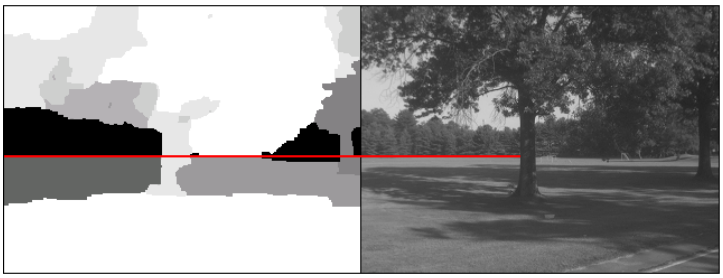
Initialization: Start with an arbitrary configuration $\mathsf{d}_{current}$
2: Set *success* to **false**
for all disparity values α **do**
  Find $\mathsf{d}_{new} = \arg\min_{\alpha} \mathsf{E}(\mathsf{d}^{\alpha})$ within a single α-expansion of d
  if $\mathsf{E}(\mathsf{d}_{new}) < \mathsf{E}(\mathsf{d}_{current})$ **then**
    $\mathsf{d}_{current} = \mathsf{d}_{new}$
    Set *success* to **true**
  end if
end for
if *success* is **true then**
  goto the line **2:**
end if

3.2.4 Noise Reduction of the Disparity Map

We observe that noise and the intensity variations caused by the changes in illumination may affect the disparity map estimation. This is actually an anticipated result considering the definitions of E_{data} and E_{smooth}. We perform intensity normalization, apply noise reduction, and enhance the sharp boundaries in the image. First of all, the pixel intensity values in $\mathbf{Fr}_i^r$ are normalized to have the same mean and variance as the pixel intensity values in $\mathbf{Fr}_j^s$. Then, we apply a smoothing filter with Gaussian kernel for removing the noise from images. Finally, we sharpen the images to enhance texture, edges and details. The disadvantage of image smoothing using the Gaussian kernel is that the process makes it difficult to highlight transitions in intensity (i.e., the sharp boundaries). These transitions may improve the accuracy of the disparity estimation. Even for the cases where the smoothing does not reduce sharp transitions, it tends to distort the fine structure of the image, which may result in inaccurate disparity values. One of the ways to overcome this problem is to apply the smoothing in restricted regions with localized parameters. Malladi and Sethian [122] introduce an alternative technique to the linear-filtering and propose the min/max curvature flow method for the denoising process. Min/max curvature flow employs a parameter whose value depends on the differential structure of the image. The advantage of min/max curvature flow filter is that it can remove small noise artifacts, while preserving the sharp boundaries among the objects. We present the effect of different different smoothing strategies to the disparity map estimation in Figure 3.6.



(a)



(b)

Figure 3.6: Effects of different smoothing strategies to the disparity map estimation. In (a), the left image is the estimated disparity map using Gaussian smoothing and the image on the right is the left image of the input stereo frame. In (b), we present the estimated disparity map using min/max curvature flow smoothing. Min/max curvature flow smoothing yields noticeable improvement in the disparity map. One of the enhancements is indicated with the red horizontal line.

3.2.5 Refinement of the Disparity Map Estimation as a Post-processing

Because of the ill-posed nature of the correspondence problem, estimated disparity maps $\mathbf{D}_i^r$ of the stereo frame $\mathbf{Fr}_i^r$ and $\mathbf{D}_j^s$ of the stereo frame $\mathbf{Fr}_j^s$ sometimes exhibit very high level of noise. A video can be treated as a volume of three-dimensional

data (i.e., two-dimensional image and the time), and it contains much more information than an individual frame does. In particular, when a mobile stereo image acquisition platform is used, the video captures depth information of the entire environment as the mobile moves. It therefore provides an additional clue for the estimation of disparity map. By employing this additional information, we propose a *novel* refinement method for the disparity map estimation as a post-processing. The key idea behind our proposed refinement algorithm is that consecutive disparity maps of a scene in a stereo video cannot change dramatically. Based on this assumption, we can examine if the disparity levels calculated in the sequential frames are reasonable values. In our refinement strategy, we perform following two controls: *point-based tracking* and *layer-based consistency*.

We are given two stereo videos $\mathbb{V}^r$ and $\mathbb{V}^s$ of the same environment recorded by a mobile data acquisition platform, with $\mathbb{V}^r : \{\mathbf{Fr}_1^r, \ldots, \mathbf{Fr}_i^r, \ldots, \mathbf{Fr}_N^r\}$ and $\mathbb{V}^s : \{\mathbf{Fr}_1^s, \ldots, \mathbf{Fr}_j^s, \ldots, \mathbf{Fr}_M^s\}$, where N and M are the number of the frames. Each stereo frame $\mathbf{Fr}_i^t$ consists of a pair of images (I_{iL}^t, I_{iR}^t), where I_{iL}^t is the left image, I_{iR}^t is the right, and $t = \{r, s\}$. We perform the disparity refinement of each stereo video individually. First of all, our algorithm selects control points in the horizontal and vertical directions at a step size of 3 pixels in stereo frame image I_{iL}^r (Figure 3.7), and then it tracks the control points across four consecutive frames $I_{(i-2)L}^r$, $I_{(i-1)L}^r$, $I_{(i+1)L}^r$, and $I_{(i+2)L}^r$ using the Lucas-Kanade tracker [180].

Let $d_p^{(i,r)}$ denote the estimated disparity value of a point p in the disparity map $\mathbf{D}_i^r$. If the estimated disparity value is not within the range of expected

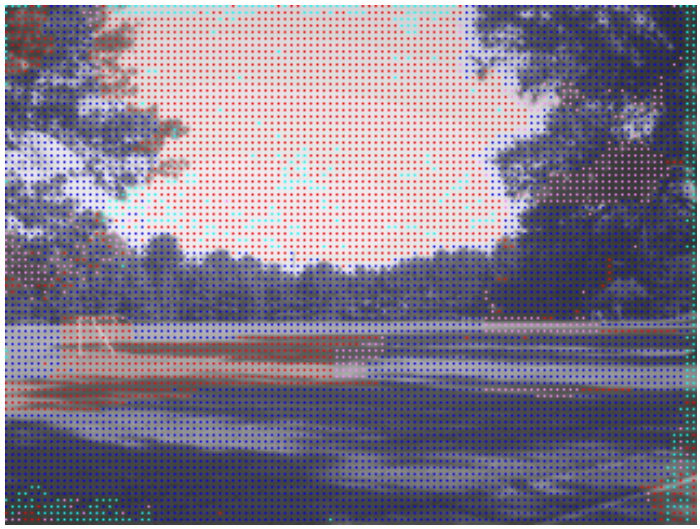


Figure 3.7: Control points in I^r_{1007L} of $\mathbf{Fr}^r_{1007}$ are presented. If an estimated disparity value is within the range of expected boundaries, it is depicted as blue; otherwise, as red. Purple points indicate that disparity values of a control point in the consecutive frames are not the same, but there may be a gradual depth change. Turquoise points are the ones where the tracking is failed.

boundaries, we update the disparity value the point p. Let $\Delta d_p^{(i,r)}$ denote the deviation of disparity values of the point p among the consecutive frames. $\Delta d_p^{(i,r)}$ is defined as follows

$$\Delta d_p^{(i,r)} = \max(|d_p^{(i,r)} - d_p^{(i+1,r)}|, |d_p^{(i+1,r)} - d_p^{(i+2,r)}|, |d_p^{(i,r)} - d_p^{(i-1,r)}|, |d_p^{(i-1,r)} - d_p^{(i-2,r)}|). \tag{3.18}$$

If the value of $\Delta d_p^{(i,r)}$ is larger than $level_{thredhold}$, we update the disparity value of p using the disparity level mean function

$$d_p^{(i,r)}{}_{new} = mean_{level}(d_p^{(i-2,r)}, d_p^{(i-1,r)}, d_p^{(i,r)}, d_p^{(i+1,r)}, d_p^{(i+2,r)}). \tag{3.19}$$

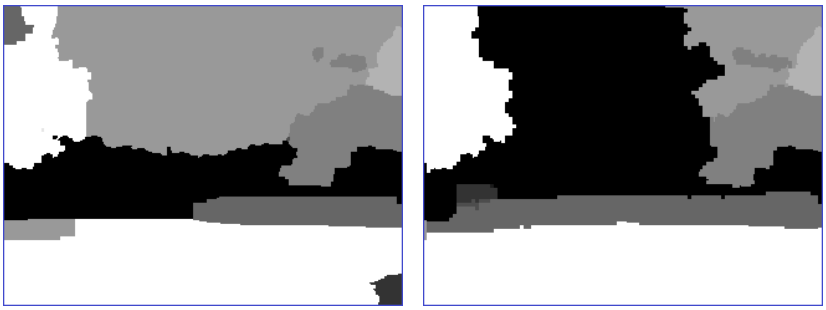
The function $mean_{level}$ computes the arithmetic average of its arguments and cast the mean value to the closest disparity level possible. Then, 8 neighbor pixels of the point p are set to the same disparity value $d_p^{(i,r)}{}_{new}$. Finally, we use a median filter to overcome problems because of noisy points and guarantee that adjacent control points have similar disparity values while preserving the edges. The drawback of our refinement strategy is that it assumes that majority of the estimated disparity values among the five consecutive frames calculated accurately. Therefore, before we start refinement process, we search the whole disparity map sequence until we find reliable disparity values.

The layer-based consistency relies on the idea that size of a disparity layer in consecutive frames cannot change dramatically, while we expect smooth disparity value changes of a layer because a layer may come closer to (or move away from) the camera in the next frame. Let $\mathfrak{s}_{i\mathfrak{L}}^r$ denote the size of a disparity layer $\mathfrak{L}$ in terms of the number of pixels having the disparity level $\mathfrak{L}$ in the disparity map $\mathbf{D}_i^r$. We compute a metric using the size of a disparity layer among the consecutive frames as follows

$$\Delta\mathfrak{S}_{i\mathfrak{L}}^r = \sum_{j=-2,-1,+1,+2} \frac{\left(\mathfrak{s}_{(i+j)\mathfrak{L}}^r\right)^2 + \left(\mathfrak{s}_{i\mathfrak{L}}^r\right)^2}{\mathfrak{s}_{(i+j)\mathfrak{L}}^r * \mathfrak{s}_{i\mathfrak{L}}^r}. \tag{3.20}$$

If the value of $\Delta\mathfrak{S}_{i\mathfrak{L}}^r$ is larger than the empirically selected value $size_{threshold}$, we

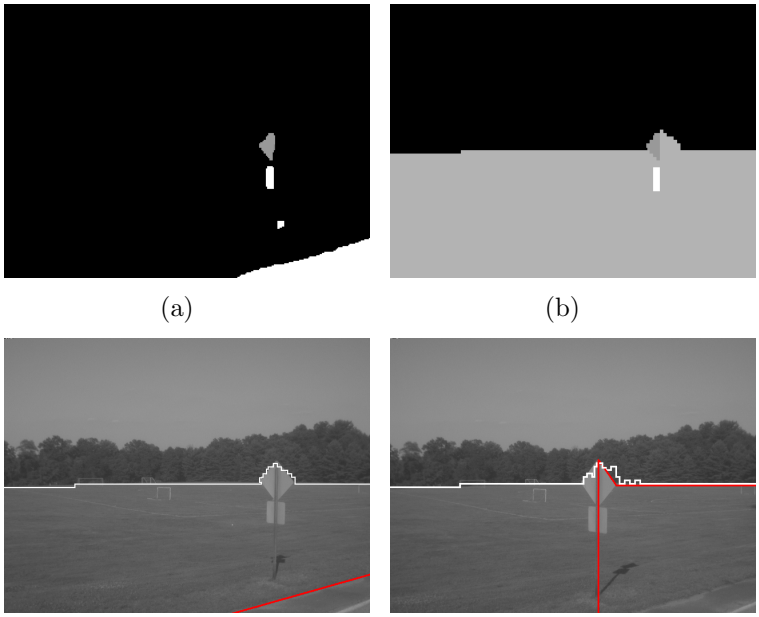
restore the layer $\mathfrak{L}$ in the disparity map $\mathbf{D}_i^r$ using the adjacent disparity maps in the sequence. Otherwise, we keep the layer $\mathfrak{L}$ as it is. We iterate the point-based tracking and the layer-based consistency controls until there are no more changes in the refined disparity map $\grave{D}_i^r$. We repeat all the steps for the disparity map $\mathbf{D}_i^s$ to obtain enhanced disparity map $\grave{\mathbf{D}}_i^s$. In Figure 3.8, we present the estimated disparity map $\mathbf{D}_{1007}^s$ before and after the refinement.



(a) Estimated disparity map $\mathbf{D}_{1007}^s$ (b) Refined disparity map $\grave{\mathbf{D}}_{1007}^s$

Figure 3.8: Disparity refinement module can compensate errors of disparity estimation step.

As will be demonstrated in the following section, building on accurately refined disparity map allows robust ground plane estimation and prevents many of the problems related to the registration. In Figure 3.9, we present an estimated ground plane line with and without disparity map refinement step.



(a) (b)

(c) I^s_{1394L} (d) I^s_{1400L}

Figure 3.9: Illustrative example of the disparity refinement and the ground plane estimation. In (a), we present the estimated disparity map $\mathbf{D}^s_{1394}$. In (b), the refined version of $\mathbf{D}^s_{1394}$, that is $\grave{\mathbf{D}}^s_{1394}$, is shown. In (c) and (d), we present the dominant plane (i.e., ground plane) estimation results before and after the refinement process. The white line in both images show the estimated ground plane line after disparity map refinement. On the other hand, the red line shows the estimated ground plane line without refinement of the disparity map.

3.2.6 Integrating Temporal Consistency Constraint into Disparity Estimation Framework

In the previous section, we presented a method that works on the already estimated disparity maps as an image enhancement tool. In this section, we will use the same idea, that is the temporal consistency of consecutive disparity maps;

however, this time we propose a way of directly integrating the temporal consistency constraint into an energy minimization framework. In addition to the E_{data} and E_{smooth} terms in the Equation 3.9, we can use the *temporal consistency constraint* if sequential stereo pairs are available. The idea of using temporal consistency for disparity estimation is not new [183], but it has not been extensively studied. The common drawback of current methods is that they add too much overhead to the system or rely on additional hardware [183, 209]. Tucakov and Lowe [183] compute the first disparity map in the sequence, and then they calculate the next one based on the relative motion of the mobile platform using an odometry. David et al. [53] add a temporal dimension to the neighborhood relationship and call it spacetime stereo. They enforce temporal consistency by using a set of spatial windows in the consecutive frames, but they assume that the camera is stationary. Min et al. [130] use a temporal coherence function combining disparity of the previous frame, motion probability and similarity based on the matched feature points. The closest techniques to our approach is probably Leung et al. [111], nonetheless, ours differs in several significant ways. Leung et al. initially follows a similar approach to ours but propose an energy minimization method based on dynamic programming. The drawback of this method is that smoothness and temporal constraints are encoded in the same energy term because of the limitation of the framework.

We are given N sequential stereo pairs continuously recorded by an *uncalibrated* mobile camera. Let $\mathbb{V}$ denote a binocular video, where $\mathbb{V} : \{\mathbf{Fr}_1, \ldots, \mathbf{Fr}_t, \ldots, \mathbf{Fr}_N\}$,

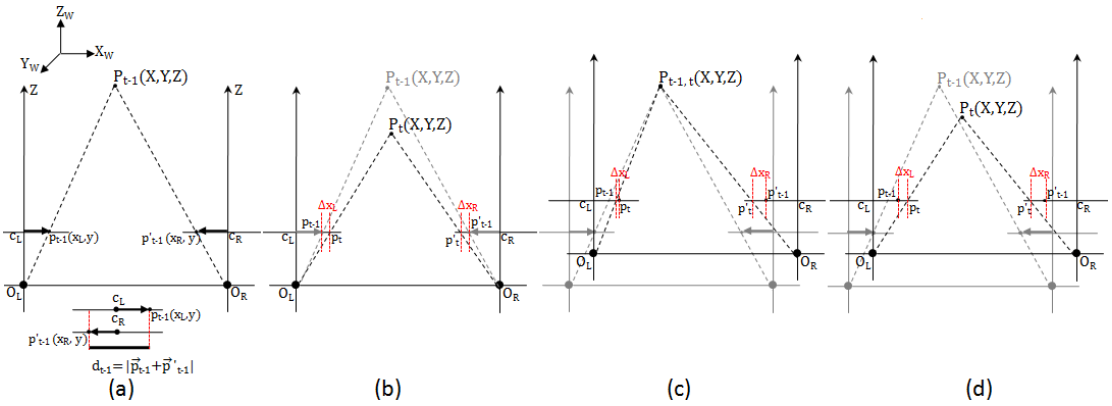


Figure 3.10: We delineate disparity changes Δx_L and Δx_R between consecutive frames caused by the relative motion between a mobile stereo rig and a physical world point P: (a) the disparity value of P at time $t-1$, where p and p' are the projections of P on the left and right image planes, c_L and c_R are the principal points of the two image planes, O_L and O_R are the centers of projection, (b) the stereo rig is stationary but P moves, (c) P is stationary but the stereo rig moves, and (d) both the stereo rig and P move.

$\mathbf{Fr}_t = (I_{tL}, I_{tR})$, I_{tL} is the left image, and I_{tR} is the right image of the stereo pair. Let p_{t-1} denotea pixel in $I_{(t-1)L}$ and let p'_{t-1} be the corresponding pixel in $I_{(t-1)R}$. p_{t-1} and p'_{t-1} are the projections of a physical worldpoint P onto image planes of the stereo rig (Figure 3.10). Similarly, p_t and p'_t denote the corresponding pixels of p_{t-1} and p'_{t-1} in the subsequent frame $\mathbf{Fr}_t$. Disparity value of P in $\mathbf{Fr}_t$, that is d_t, should be consistent with d_{t-1} in $\mathbf{Fr}_{t-1}$ because of the time and space locality. We call this factor *temporal consistency constraint* and add it to the general energy term given in the Equation 3.9. The new energy function becomes of the form:

$$\mathsf{E}(\mathsf{d}) = \mathsf{E}_{data}(\mathsf{d}) + \mathsf{E}_{smooth}(\mathsf{d}) + \mathsf{E}_{temporal}(\mathsf{d}), \tag{3.21}$$

where the value of E_{smooth} is larger when there is a strong disagreement between

the temporal samples of the disparity value and zero otherwise. Integrating a new term into an existing energy function is a demanding process. The new term should treat inputs fairly, its addition should not change dominant features of the function, and be a metric on the space of disparity labels. Under these assumptions, given disparity values of a point in the previous and current frame in the sequence, the temporal consistency energy term is defined as a function of the current d_t and previous disparity estimate d_{t-1} of a point of interest

$$\mathsf{E}_{temporal}(\mathsf{d}) = C(d_{t-1}, d_t). \tag{3.22}$$

To explicitly specify $C(d_{t-1}, d_t)$, one naive approach is simply to copy one of the existing terms in the energy function and modify it. Another one is to encode both prior and temporal constraints in the same term [111] although they exploit different characteristics. A better way is to examine the new external factor and to define a new function encoding unique features of it. Formulation of $C(d_{t-1}, d_t)$ requires to observe possible disparity values between consecutive frames. Therefore, the first step is to find out how much a pixel of interest may move between the consecutive frames. Relative motion of the pixel and mobile platform will determine the new location of pixel. This relative change may result in a disparity change in the consecutive frame (Figure 3.10). We can categorize all possible relative motions between a physical point and a camera in three-dimensional space into three main groups: the distance between them stays the same, increases, and decreases. We can anticipate a disparity change in the image

planes if the distance changes. Because we do not have external sensors [183] to calculate motion of the mobile platform, we need to estimate inter-frame motion in some way. A straightforward approach is to use optical flow [32]. This will provide us the exact location and disparity value of the pixel in the previous frame. We here want to investigate if we need to know the exact location and accordingly the exact disparity value of the pixel in the previous frame. If not, we may eliminate the optical flow. In almost all real world scenarios, pixels that belong to different parts of the same entity move together in groups if the entity is not deformable object. These groups may form different disparity layers, a set of pixels whose disparity values are the same. From this point of view, we anticipate that the disparity value of a pixel between consecutive frames is related to its neighbors in the previous frame because of the time and space locality. Furthermore, the new disparity values have to be within close range of the current disparity values. Given the coordinates of the pixel $p_t(x_t, y_t)$ in $\mathbf{Fr}_t$ and the disparity map $\mathbf{D}_{t-1}$ (Figure 3.11(a)), we can draw a circle with origin $\mathbf{D}_{t-1}(x_t, y_t)$ and diameter of $\tau * |d_{t-1}|$, where τ may vary depending on the stereo rig as shown in Figure 3.11(b).

We draw 4 more circles with the same diameter but different origins: $(x_t - \frac{\tau * |d_{t-1}|}{2}, y_t)$, $(x_t + \frac{\tau * |d_{t-1}|}{2}, y_t)$, $(x_t, y_t - \frac{\tau * |d_{t-1}|}{2})$, and $(x_t, y_t + \frac{\tau * |d_{t-1}|}{2})$. We then select four intersection points, $\tilde{p}^1_{t-1}$, $\tilde{p}^2_{t-1}$, $\tilde{p}^3_{t-1}$, and $\tilde{p}^4_{t-1}$, and the point in the center $(\tilde{p}^0_{t-1})$. One can select more points from the intersections of the circles in order to increase the accuracy. Finally, we use disparity values of these approximate corresponding pixels to estimate the disparity value of actual corresponding pixel

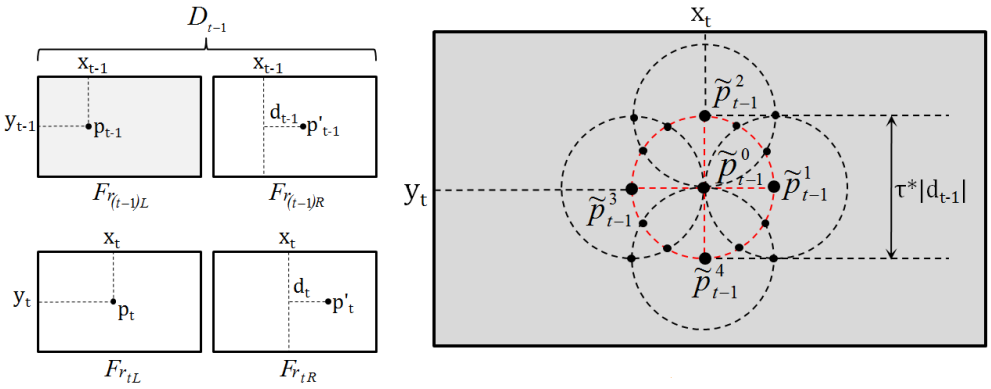


Figure 3.11: Estimation of the approximate disparity value $\tilde{d}_{t-1}$ of pixel p_t in disparity map $\mathbf{D}_{t-1}$ without optical flow: (a) frames used to compute the disparity map $\mathbf{D}_{t-1}$, the actual disparity value d_{t-1} of p_t, and the coordinates of pixel p_t in F_t, and (b) the candidate pixels correspond to approximate locations of p_t in $\mathbf{D}_{t-1}$.

p_{t-1} as follows

$$\tilde{d}_{t-1} = m * \left(\sum_{i=0}^{m} \frac{1}{\tilde{p}^i_{t-1}} \right)^{-1}, \tag{3.23}$$

where m is the number of the selected points. One can set values of m and τ in a way that the average error $\frac{1}{n} \sum_{i=1}^{n} \frac{|d^i_{t-1} - \tilde{d}^i_{t-1}|}{|d^i_{t-1}|}$ for all the pixels in P will be less than desired error value ϵ. Using either the optical flow or our approximation to track a pixel does not affect the performance our proposed method notably because we need the disparity layers to extract the dominant plane rather than the individual disparity values. In addition, our approach has constant time complexity, while the time complexity of a robust implementation of optical flow algorithm may grow

quadratically [32]. In fact, the real power of our method comes from the proper integration of temporal consistency into the energy function where we have the advantage of using an approximate disparity value instead of the exact disparity value.

After determining disparity value of p_t in the previous frame, we can assign a consistency penalty based on the disparity change. Nevertheless, the amount of the penalty should not become too large to tolerate. Therefore, the choice of $C(d_{t-1}, d_t)$ is very critical because the degree of penalization can affect the disparity estimation drastically. For example, when the value of penalty increases linearly with increasing $|d_t - d_{t-1}|$, for very large values of $|d_t - d_{t-1}|$, the penalty can be so immense that the disparity changes may not be possible. We therefore limit the maximum value of the penalty by the value κ_{max} as follows

$$C(d_i, d_j) = \begin{cases} \kappa_{max} & \text{if } |d_i - d_j| \geq \Omega \\ \kappa_{min} & \text{if } |d_i - d_j| = 0 \\ \frac{\kappa_{max} - \kappa_{min}}{\Omega} |d_i - d_j| + \kappa_{min} & \text{otherwise} \end{cases}$$

where κ_{max} is fixed to be 17.00, penalty factor Ω is 20.00, and κ_{min} is assigned to 1.00. The penalty κ_{min} may be greater than zero if disparity change is more likely between the consecutive frames.

3.2.7 Dominant Plane Estimation

Disparity map estimation is followed by a layer-based segmentation step allowing for the estimation of the position of the dominant plane within each image for spatial registration purposes. For temporally aligned videos, there are usually cases where there is not enough common spatial information within the two aligned frames to allow reliable spatial registration of the entire frame regions. One such example is illustrated in Figure 3.12.

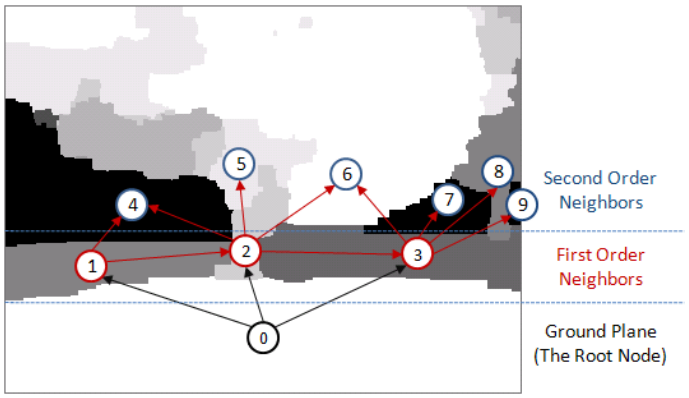
(a) I^r_{1398L} (b) I^s_{1950L}

Figure 3.12: Example of two temporally aligned frames from the reference and secondary videos. In (b), we present the left image I^s_{1950L} of the stereo pair in the secondary video. In (a), we present I^r_{1398L}, which is the corresponding frame of the image I^s_{1950L}. Although the two images are temporally matched, it is still challenging to spatially register I^s_{1950L} onto I^r_{1398L} because of the complex scene structure.

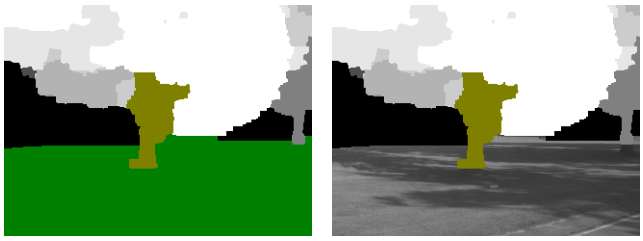
To overcome this challenge, a real world scene can be interpreted with respect to a dominant plane (e.g., ground plane) which is a planar surface and occupies the largest domain in the image. In this way, we can construct a mapping between two image using the dominant planes. In this section, we address the problem of

finding the dominant plane within the frame. The dominant plane estimation is an essential task for a robust scene registration. We develop an algorithm for dominant plane detection using disparity layers and their textural structure. We assume that the distance from the dominant plane to the mobile platform is finite. We represent the scene structure as a collection of disparity layers. Many stereo algorithms impose the assumption that all three-dimensional points in each layer lie on the same plane in the three-dimensional world and the disparities in each layer obey the same plane equation [163], whereas resulting disparity maps do not always satisfy this constraint and sometimes include over-segmented layers. We relax the constraint and assume that the extracted disparity layers do not adhere to a single scene layer. In contrast, we allow for a region merging step that can analyze the estimated disparity maps to generate a refined layer representation (Figure 3.13).

A hierarchical tree-like data structure with a set of linked nodes is built to estimate ground plane layer within each frame using neighborhood relationship among the disparity layers. We start by constructing a graph on the over-segmented disparity map, as shown in Figure 3.13, whose nodes represent the candidate disparity layers and an edge refers to the first order neighborhood relationship of a candidate layer. The first disparity layer, the node 0 in Figure 3.13(a), is assumed to be ground plane, knowing that other candidate layers may also be part of the ground plane. In many cases, there may also be more than one root node. We employ a log-Gabor filter bank to extract texture features of each candidate disparity layer



(a)



(b) (c)

Figure 3.13: Illustrative example of region merging process for an over-segmented disparity map. In (a), we present the over-segmented disparity map $\mathbf{D}^s_{1057}$. The nodes mark the disparity layers that may belong to ground plane. In (b), we present the merged regions in $\mathbf{D}^s_{1057}$. In (c), we present the final estimated ground plane $\mathbf{G}^s_{1057}$ overlaid with texture from the original input image.

[163]. We use this information to merge the disparity layers that belong to the ground plane.

In the over-segmented disparity map shown in Figure 3.13(a), nodes represent *candidate disparity layers* that may belong to the ground plane and links show the first and second order neighborhood. The hierarchical structure is actually not a tree, but a planar directed connected graph where a candidate disparity layer has a

link to the neighbor candidate disparity layer. The top area of the frame is usually not the part of ground plane. Hence, the first disparity layer (i.e., the node 0), the closest regions to the mobile platform, is assumed to be ground plane, while we know that other candidate layers may be parts of the ground plane. Let $N(n, m)$ denote m^{th} order neighbors of the node n. First-order neighbors of the root node 0 are defined as the layers (or nodes) connected to the first disparity layer: $N(0, 1) = \{1, 2, 3\}$. Similarly, first-order neighbors of the nodes 1, 2, and, 3 are $N(1, 1) = \{2, 4\}$, $N(2, 1) = \{3, 4, 5, 6\}$, and $N(3, 1) = \{6, 7, 8, 9\}$. We can obtain the second-order neighbors of the root node using above relationships assuming the transition property: $N(0, 2) = \{3, 4, 6, 7, 8, 9\}$. Please notice that there may be more than one root node for some cases. Our experiments showed that growing the tree to the second-order neighbors of the root node was required for reliable region merging. It is observed that the further growing does not change the result, while it requires more computational time. After candidate layers are determined, we perform pairwise comparison of them using their texture similarities.

Gabor filter banks are a traditional choice for obtaining localized frequency information. Nevertheless, they have two main limitations: (1) the maximum bandwidth of a Gabor filter is limited to approximately one octave and (2) Gabor filters are not optimal if one is seeking broad spectral information with maximal spatial localization [106]. An alternative to the Gabor function is the log-Gabor function. It is suggested that natural images are better coded by filters (e.g., log-Gabor function) that have Gaussian transfer functions when viewed on the

logarithmic frequency scale [61]. Please note that Gabor functions have Gaussian transfer functions when viewed on the linear frequency scale. Once the candidate disparity layers are determined, using their neighborhood relationship and texture similarities we perform the steps shown in Algorithm 3 to merge the layers.

Algorithm 3 Region Merging

Require: $\alpha \neq 0 \vee \beta \neq 0$
  for each candidate disparity layer $\mathbf{L}_i$ **do**
    Divide layer $\mathbf{L}_i$ into α regions
    for all k such that $1 \leq k \leq \alpha$ **do**
      Extract texture feature $\mathbf{T}_k^i$ of region $\mathbf{R}_k^i$ using log-Gabor filter bank
      Compute mean absolute deviation $\mathbf{MAD}_{\mathbf{T}_k^i}$ of region $\mathbf{R}_k^i$
    end for all
    Compute mean absolute deviation $\mathbf{MAD}_{\mathbf{L}_i}$ of the layer $\mathbf{L}_i$
    for each $\mathbf{L}_j$, where $\mathbf{L}_j \in N(\mathbf{L}_i, 1)$ **do**
      Divide layer $\mathbf{L}_j$ into β regions
      for all l such that $1 \leq l \leq \beta$ **do**
        Extract texture feature $\mathbf{T}_l^j$ of region $\mathbf{R}_l^j$ using log-Gabor filter bank
        Compute mean absolute deviation $\mathbf{MAD}_{\mathbf{T}_l^j}$ of region $\mathbf{R}_l^j$
      end for all
      Compute mean absolute deviation $\mathbf{MAD}_{\mathbf{L}_j}$ of the layer $\mathbf{L}_j$
    end for each
    Compute similarity metric $\mathbf{m}_{MAD}$
    if $\mathbf{m}_{MAD} < \mathbf{MAD}_{threshold}$ **then**
      Merge layers $\mathbf{L}_i$ and $\mathbf{L}_j$
    else
      Do not merge layers
    end if
  end for each

We first divide each candidate layer into number of square regions. The number of the square regions in a layer varies according to the size of the layer. To assess similarity between textures of the neighboring disparity layer, we use a mean absolute deviation (MAD)-based similarity metric [98]. MAD is known to

be insensitive to outliers and the points in the extreme tails of the distribution. Given a set of numbers, MAD is defined as the mean of the absolute deviations of the numbers from the mean of the set. We define a similarity metric using MAD values as follows

$$\mathbf{m}_{MAD} = \frac{(\mathbf{MAD}_1)^2 + (\mathbf{MAD}_2)^2}{\mathbf{MAD}_1 * \mathbf{MAD}_2}, \tag{3.24}$$

where $\mathbf{MAD}_1$ and $\mathbf{MAD}_2$ are MADs of the layers to be compared. We perform a pairwise comparison of MAD values of neighboring layers. The layers $\mathbf{L}_1$ and $\mathbf{L}_2$ are merged if the metric $\mathbf{m}_{MAD}$ is less than $\mathbf{MAD}_{threshold}$. The value of $\mathbf{MAD}_{threshold}$ is fixed to be 2.25, where a value of 2.0 indicates that two layer have exactly the same texture.

We first divide each candidate layer into number of square regions. The number of the square regions in a layer varies according to the size of the layer. Then, we compute MAD of regions and layers using a log-Gabor filter bank output [151], respectively. In our setting, MAD of a region can be defined as the mean of the absolute deviations from mean of the data

$$\mathbf{MAD}_{\mathbf{T}_i^L} = \frac{1}{N} \sum_{j=1}^{N} |\mathbf{x}_j - \overline{\mathbf{T}}_i^L| \tag{3.25}$$

where L is the index of the layer, $\mathbf{R}_i^L$ is a region in layer L, N is the number of pixels in $\mathbf{R}_i^L$, and $\mathbf{x}_j$ is the data element of $\mathbf{T}_i^L$. Using the definition in equation (3.25), we now have two sets of MAD values:

$$\mathbf{M}_{L1} = \{\mathbf{MAD}_{\mathbf{T}_1^{L1}}, \mathbf{MAD}_{\mathbf{T}_2^{L1}}, \cdots, \mathbf{MAD}_{\mathbf{T}_i^{L1}}, \cdots, \mathbf{MAD}_{\mathbf{T}_\alpha^{L1}}\}$$
$$\mathbf{M}_{L2} = \{\mathbf{MAD}_{\mathbf{T}_1^{L2}}, \mathbf{MAD}_{\mathbf{T}_2^{L2}}, \cdots, \mathbf{MAD}_{\mathbf{T}_i^{L2}}, \cdots, \mathbf{MAD}_{\mathbf{T}_\beta^{L2}}\} \quad (3.26)$$

where α and β are the number of regions in the layers $L1$ and $L2$, respectively. To be able to compare two sets in equation (3.26), we calculate MAD of each layer as follows

$$\mathbf{MAD}_{\mathbf{M}_{L1}} = \frac{1}{\alpha}\sum_{j=1}^{\alpha} |\mathbf{MAD}_{\mathbf{T}_j^{L1}} - \overline{\mathbf{M}}_{L1}| \quad (3.27)$$

$$\mathbf{MAD}_{\mathbf{M}_{L2}} = \frac{1}{\beta}\sum_{j=1}^{\beta} |\mathbf{MAD}_{\mathbf{T}_j^{L2}} - \overline{\mathbf{M}}_{L2}| \quad (3.28)$$

Finally, the layers $L1$ and $L2$ are merged or not based on the value of similarity metric $\mathbf{m}_{MAD}$ as follows:

$$decision = \begin{cases} merge & \text{if } \mathbf{m}_{MAD} \text{ is less than } \mathbf{MAD}_{threshold} \\ not\ merge & \text{otherwise} \end{cases}$$

where

$$\mathbf{m}_{MAD} = \frac{(\mathbf{MAD}_{\mathbf{M}_{L1}})^2 + (\mathbf{MAD}_{\mathbf{M}_{L2}})^2}{\mathbf{MAD}_{\mathbf{M}_{L1}} * \mathbf{MAD}_{\mathbf{M}_{L2}}} \quad (3.29)$$

and the value of $\mathbf{MAD}_{threshold}$ is fixed to be 2.25. For example, a value of 2.0 indicates that two layers have exactly the same texture.

3.2.8 Spatial Alignment of the Dominant Planes

Various spatial alignment strategies apply the same transformation function to the entire image region. The apparent disadvantage of this approach is that to impose the same transformation function for a scene that consists of different planes may produce inaccurate registration results. In our setting, a global spatial alignment strategy cannot ensure the accurate registration of the ground planes (i.e., dominant planes) because of complex scene geometry. Therefore, we propose a spatial registration approach that solely employs the dominant planes in the scene. We consider the spatial alignment problem as the process of transforming the ground plane $\mathbf{G}_j^s$ in the frame $\mathbf{Fr}_j^s$ in such a way that it is at the same position, orientation, and scale as the ground plane $\mathbf{G}_i^r$ in the frame $\mathbf{Fr}_i^r$. If this is accomplished, the ground planes can be compared for the changes pixel by pixel. Modules of a generic spatial alignment method are shown in Figure 3.14.

Registration is treated as an optimization problem with the goal of construct a spatial mapping that will bring $\mathbf{G}_j^s$ into alignment with $\mathbf{G}_i^r$. The transform module of the registration process represents the spatial mapping of points from the ground plane $\mathbf{G}_j^s$ space to points in the ground plane $\mathbf{G}_i^r$ space. The interpolator is used to evaluate image intensities at non-grid positions. The metric component provides a measure of how well $\mathbf{G}_i^r$ is matched by the transformed $\mathbf{G}_j^s$. The optimizer optimizes the quantitative criterion formed by the metric over the search space of the transformation function parameters.

When the cameras having widely spaced views are used to capture the scene,

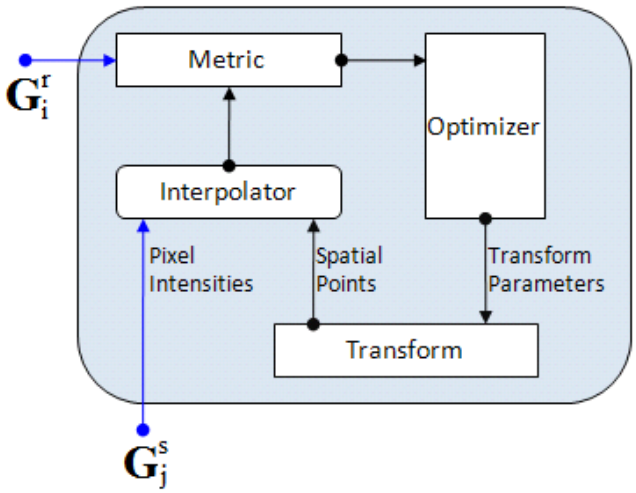


Figure 3.14: Illustration of the modules of a generic registration process [94]. $\mathbf{G}_i^r$ and $\mathbf{G}_j^s$ refer to the estimated ground planes in temporally aligned frames.

it may improve the registration accuracy to employ a non-global transformation function. We deal with a very challenging spatial image alignment task that requires the use of intra-structure information and sophisticated transformation models. Our experiments show that some ground planes may have diverse disparity values. Therefore, one should avoid using a single transformation function for all the segmented ground planes in a video. Instead, we dynamically estimate the suitable transformation function and the parameters for each frame. In the proposed method, an accurate spatial registration depends on both accurate temporal alignment and ground plane segmentation. These constraints add more challenges to design the modules of the registration process. Some basic error metrics tend to ignore the fact that some of the pixels being compared may lie outside the original image boundaries [174]. There may be a low overlap between the images to be aligned. Because of anticipated dominant plane segmentation

artifacts, this is quite important issue while selecting a proper metric for our framework. Furthermore, using metrics based on direct comparison of gray levels is not applicable because $\mathbf{G}_i^r$ and $\mathbf{G}_j^s$ may have varying illumination conditions. To overcome these limitations, we employ Viola-Wells mutual information [189] as the spatial alignment cost function to maximize mutual information [204].

Choosing an appropriate spatial transformation for the registration is a critical step to detect the changes between the image. When the scene has mostly rigid structures and if the mobile platform motion is reasonably small, registration can often be performed using a global spatial transformation function such as similarity (i.e., a combination of translation, rotation, and scaling), affine, or projective (i.e., homography) transformations [148]. In our case, while applying these transform functions to the entire frame region is probably not feasible, we can employ them to align the dominant planes. The similarity transformation is performed by applying the translation, the rotation, and the scaling. It is specified by four transformation parameters

$$\grave{x} = xs\cos(\theta) - ys\sin(\theta) + t_x \tag{3.30}$$

$$\grave{y} = ys\sin(\theta) + ys\cos(\theta) + t_y \tag{3.31}$$

where $\grave{x}$ and $\grave{y}$ are the transformed points, s is the scaling parameter, θ is the rotation parameter, and (t_x, t_y) are the translation parameters. The similarity transformation is commonly used for the registration of rigid structures where the

image acquisition platform is at a very large distance from the scene. The affine transformation function does not preserve the angles or lengths, but it retains the parallel lines in the scene. The affine transformation function has six parameters. If enough number of corresponding points are known, they can be estimated by solving the following equations

$$\grave{x} = ax + by + t_x \tag{3.32}$$

$$\grave{y} = cx + dy + t_y. \tag{3.33}$$

The six parameters in Equations 3.32 and 3.33 specify the rotation, the scaling, the shearing, and the translation. Image acquisition is a projective process where a three-dimensional world is projected to a two-dimensional space. When the camera is far from the scene, the projective nature can be approximated by an affine transformation function. On the other hand, to be able to estimate actual homography parameters, we employ assumption of the presence of a dominant plane (i.e., ground plane) in the scene. Let $\mathbf{p} = (x, y, 1)^T$ denote homogeneous coordinate of a point in $\mathbf{G}_i^r$. Let H be a nonsingular 3×3 homography matrix of the spatial transformation between $\mathbf{G}_i^r$ and $\mathbf{G}_j^s$. H is represented by eight transform parameters as follows

$$\mathrm{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}, \tag{3.34}$$

and the transformed point $\grave{\mathbf{p}} = (\grave{x}, \grave{y})$ is described by

$$\grave{x} = \frac{h_{11}x + h_{11}y + h_{13}}{h_{31}x + h_{32}y + 1} \tag{3.35}$$

$$\grave{y} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}. \tag{3.36}$$

Namely, we need to estimate the eight transformation parameters in the matrix H to calculate the homography that maps the points. Nevertheless, one should note that H may be different for consecutive frames because of movement of the mobile platform and the change in the scene structure. We want to point out that we are not restricted to always using the homography because a more complicated or simpler transform model may result better spatial registration. Therefore, we also include the homography transformation with radial lens distortion [199]. Image acquisition device may introduce a certain amount of nonlinear distortion. This adds two more transformation parameters k_1 and k_2, which are the radial distortion coefficients in the polynomial radial distortion model [195]. We categorize the transformation functions based on the number of the transform parameters (Table 3.1).

Considering all the aspects of the mobile image acquisition platform described below, we employ the dual-bootstrap iterative closest point algorithm [199]. SIFT features [117] and Harris corners [127] are used as the image descriptors to match localized regions between the images. The registration framework is capable of

Transformation	Number of Parameters
1. Similarity	4
2. Affine	6
3. Homography	8
4. Homography with radial lens distortion	10

Table 3.1: The order of the transformation functions used in the spatial registration. The similarity transform has the lowest order, while the homography plus radial lens distortion has the highest order.

automatically switching from the simplest to higher order transformation functions based on the registration error. The transformation function estimation is performed based on the criteria [199] employing the weighted average error of the matching of transformed features and the spatial alignment error of the local patches. In Table 3.1, we present the set of transformation models used in the registration where each one successively involves more parameters. The system has a hierarchy of following transformation functions: similarity, affine, homography, and homography with radial lens distortion.

Chapter 4

Change Detection

In this chapter, we present methods developed for the detection of the changes between spatiotemporally aligned frames from different videos. In particular, we deal with two data acquisition scenarios: a stereo video captured by a mobile platform following unknown trajectories and a monocular video captured by a stationary camera in a scene with dynamic background where there are several altering elements in the the background.

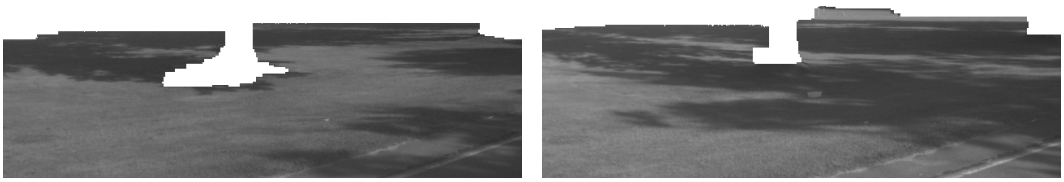
4.1 Image Change Detection in Stereo Videos

Let us assume that we are given two stereo videos $\mathbb{V}^r$ and $\mathbb{V}^s$ (i.e., reference and secondary videos) of the same environment recorded by a mobile camera platform

following unknown trajectories. A necessary preprocessing step for all change detection algorithms is accurate image registration, the alignment of the two images into the same spatial coordinate system so that changes at corresponding pixels in two images resulting from different camera positions alone are virtually never desired to be detected as real changes. In the Chapter 3, we presented a complete algorithm including techniques for the temporal alignment, for the segmentation of the dominant plane in the scene, and for the spatial registration. By applying these methods, we can bring the two stereo videos into an spatiotemporal alignment. Thereby, it is now feasible to analyze the changes between the videos.

4.1.1 Comparison of the Ground Planes

The goal of change detection step is to distinguish the new object pixels from the ground plane pixels, and the ability to quantitatively compare two registered ground plane (i.e., dominant plane) images is very crucial task. Illumination distributions of the scene in ground plane images $\mathbf{G}_i^r$ and $\mathbf{G}_j^s$ (Figure 4.1) are quite different because of temporal separation, and there are also large shadow regions within the images. Furthermore, it is a well-known fact that a highly accurate registration step is required to obtain good change detection results [59]. These factors may result in false detection in the absence of any change. Hence, we need a change detection strategy that is robust against illumination changes, and the comparison module should not be very sensitive to minor registration errors and a reasonable range of spatial distortion.



(a) Ground plane $\mathbf{G}^r_{639}$ (b) Ground plane $\mathbf{G}^s_{1057}$

Figure 4.1: Two extracted ground planes to be compared. In (a) and (b), we present dominant planes $\mathbf{G}^r_{639}$ and $\mathbf{G}^s_{1057}$ which were segmented from the stereo frames $\mathbf{Fr}^r_{639}$ and $\mathbf{Fr}^s_{1057}$ in the videos $\mathbb{V}^r$ and $\mathbb{V}^s$.

4.1.2 Feature Extraction for the Change Detection

Texture feature has been widely and successfully used for various application in computer science, such as content based image retrieval, segmentation, and classification. It represents the spatial arrangement of intensity values in an image and also remains relatively stable with respect to noise and illumination changes unless it is covered by objects. This aspect of the texture feature has given us the idea that differences in texture characteristics can be employed for the change detection. It is common for change decision at a given point to be based on a small block of pixels in the neighborhood of the point in each of the two images because interesting changes are often associated with localized groups of pixels. From this point of view, instead of performing pixel by pixel comparison, we first divide $\mathbf{G}^r_i$ and $\grave{\mathbf{G}}^s_j$ (that is the ground plane $\mathbf{G}^s_j$ after the spatial registration) into a number of regions of 13 by 13 pixels (Figure 4.2). We apply the decision reached at a block to all the pixels in it. Although texture representations are more accurate than local statistics, when regions of change and the scene are homogeneous,

the texture difference measure will fail. Therefore, we have decided to integrate gradient values into the change detection module. Our experiments show that Sobel gradient detection [170] is barely affected by changes in the illumination conditions between spatiotemporally aligned images. Hence, we calculate image intensity gradients of corresponding subregions in two images. Two important aspects of this approach is that minor spatial misalignments and changes in the illumination can be handled without further preprocessing.

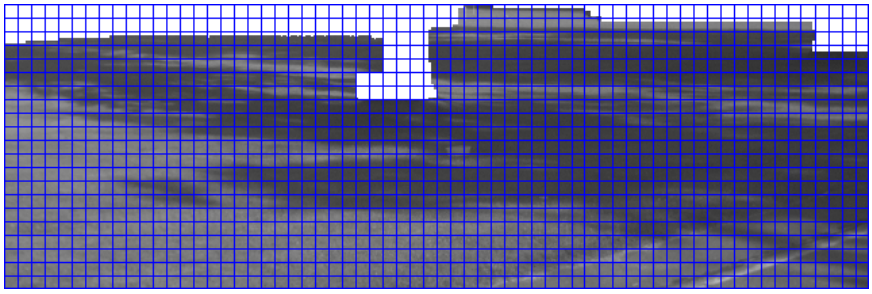


Figure 4.2: The comparison module should not be very sensitive to image scale and rotation, and it should provide robust matching across a reasonable range of affine distortion, addition of noise, and change in illumination.

We extract texture information using the similar approach explained in Section 3.2.7. Let $\mathbf{R}_k^{i,r}$ denote a square subregion k in $\mathbf{G}_i^r$, and let $\mathbf{T}_k^{i,r}$ the texture feature extracted from $\mathbf{R}_k^{i,r}$. We assume that each block contains N pixels. First, we extract the texture feature of $\mathbf{R}_k^{i,r}$ and its corresponding subregion $\mathbf{R}_k^{j,s}$ in the spatially transformed $\grave{\mathbf{G}}_j^s$ using a log-Gabor filter bank. Then, we calculate MADs of the texture features $\mathbf{T}_k^{i,r}$ and $\mathbf{T}_k^{j,s}$ as defined below

$$\mathbf{MAD}_{\mathbf{T}_k^{i,r}} = \frac{1}{N}\sum_{l=1}^{N}|\mathbf{x}_l - \overline{\mathbf{T}}_k^{i,r}|$$
$$\mathbf{MAD}_{\mathbf{T}_k^{j,s}} = \frac{1}{N}\sum_{l=1}^{N}|\mathbf{x}_l - \overline{\mathbf{T}}_k^{j,s}| \tag{4.1}$$

where $\mathbf{x}_l$ denotes the elements of $\mathbf{T}_k^{i,r}$ and $\mathbf{T}_k^{j,s}$. We define a metric denoted by $\mathbf{m}_T$ to assess the similarity between $\mathbf{R}_k^{i,r}$ and $\mathbf{R}_k^{j,s}$. The similarity metric $\mathbf{m}_T$ is computed using the formula

$$\mathbf{m}_T = \frac{(\mathbf{MAD}_{\mathbf{T}_k^{i,r}})^2 + (\mathbf{MAD}_{\mathbf{T}_k^{j,s}})^2}{\mathbf{MAD}_{\mathbf{T}_k^{i,r}} * \mathbf{MAD}_{\mathbf{T}_k^{j,s}}} \tag{4.2}$$

As a second phase, Sobel gradient operator is applied to the regions $\mathbf{R}_k^{i,r}$ and $\mathbf{R}_k^{j,s}$ in vertical and horizontal directions. The resulting gradient approximations are combined to give the gradient magnitude as follows

$$\mathbf{G}_{\mathbf{R}_k^{i,r}} = \sqrt{(\mathbf{G}_{\mathbf{R}_k^{i,r}{}_V})^2 + (\mathbf{G}_{\mathbf{R}_k^{i,r}{}_H})^2}$$

$$\mathbf{G}_{\mathbf{R}_k^{j,s}} = \sqrt{(\mathbf{G}_{\mathbf{R}_k^{j,s}{}_V})^2 + (\mathbf{G}_{\mathbf{R}_k^{j,s}{}_H})^2} \tag{4.3}$$

where the subscripts V and H stand for vertical and horizontal, respectively. Then, we use MAD values of $\mathbf{G}_{\mathbf{R}_k^{i,r}}$ and $\mathbf{G}_{\mathbf{R}_k^{j,s}}$ to define the same metric explained above for the gradient magnitude as follows

$$\mathbf{MAD}_{\mathbf{G}_{\mathbf{R}_k^{i,r}}} = \frac{1}{N}\sum_{l=1}^{N}|\mathbf{x}_l - \overline{\mathbf{G}}_{\mathbf{R}_k^{i,r}}|$$
$$\mathbf{MAD}_{\mathbf{G}_{\mathbf{R}_k^{j,s}}} = \frac{1}{N}\sum_{l=1}^{N}|\mathbf{x}_l - \overline{\mathbf{G}}_{\mathbf{R}_k^{j,s}}| \tag{4.4}$$

and

$$\mathbf{m}_G = \frac{(\mathbf{MAD}_{\mathbf{G}_k^{i,r}})^2 + (\mathbf{MAD}_{\mathbf{G}_k^{j,s}})^2}{\mathbf{MAD}_{\mathbf{G}_k^{i,r}} * \mathbf{MAD}_{\mathbf{G}_k^{j,s}}}. \tag{4.5}$$

Finally, combined similarity metric $\mathbf{m}$ is computed as

$$\mathbf{m} = \frac{\mathbf{m}_T + \mathbf{m}_G}{2} \tag{4.6}$$

for the corresponding regions $\mathbf{R}_k^{i,r}$ and $\mathbf{R}_k^{j,s}$. If the value of the combined metric $\mathbf{m}$ is greater than $\mathbf{MAD}_{threshold}^{T-G}$, we label the square subregion as a region of change (Figure 4.3). Choosing a threshold value is critical because low or high value will result different problems. We calculate it dynamically based on the current image content because experimentally selecting a value is not appropriate for a robust autonomous vision system. We anticipate that local thresholding may be useful especially when the scene illumination varies locally between the registered frames.

Our experiments show that we can obtain much better change detection results by exploiting both gradient and texture differences than by using the intensity differences directly.

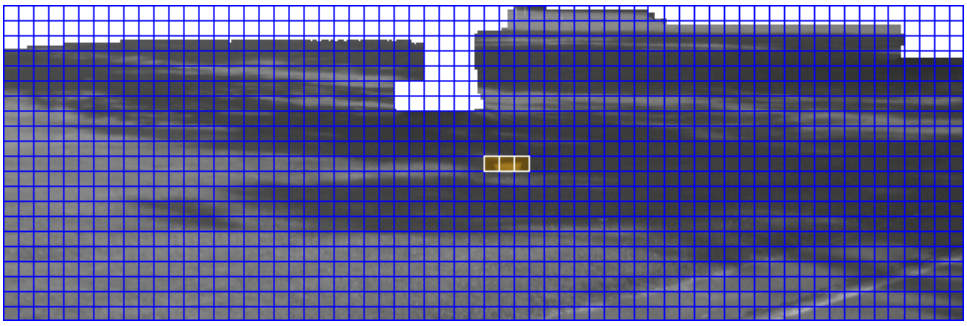


Figure 4.3: Illustrative example of a change detection result. We labeled white bordered square subregions as the regions of change because combined texture and gradient features are quite different in corresponding regions $\mathbf{R}_k^{i,r}$ and $\mathbf{R}_k^{j,s}$.

4.2 Detection of Changes in Monocular Videos

In this section, we investigate a more complicated change detection scenario where a stationary monocular camera is employed to capture videos in outdoor scenes where the background has several altering elements that may cause false alarms [69].

In such cases, there are almost always changes in the scene. To establish a clear distinction between what is a relevant change and what is not, we first categorize the change into two main classes; namely, *ordinary change* and *salient change*. The ordinary change is considered as irrelevant if they are recurrent elements and changes pertaining to the dynamic background of the scene. On the other hand, an alteration that does not conform to the expected pattern of ordinary change is defined as the salient change (e.g., transient changes). We need to distinguish ordinary changes from salient changes in order to avoid false alarms. To this end, we follow the data flow diagram illustrated in Figure 4.4 which consists of

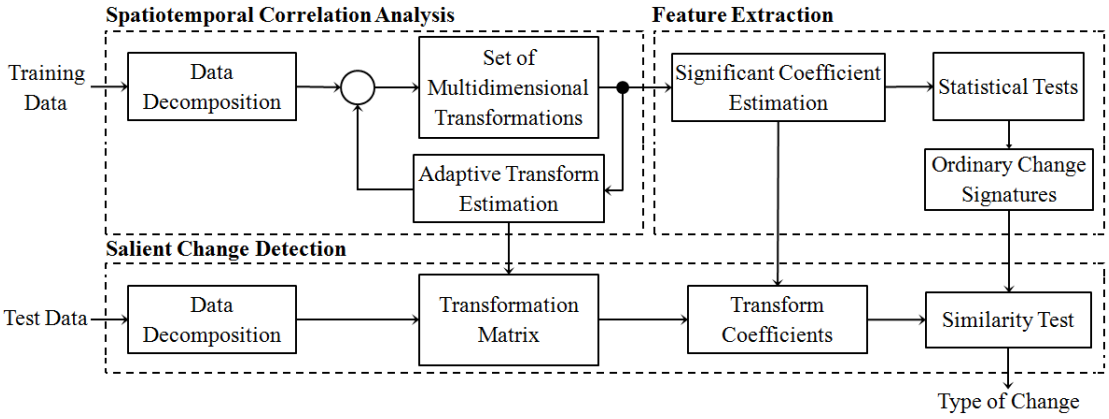


Figure 4.4: Our algorithm consist of three main blocks. First, given a video without salient changes, we are interested in finding representations where the spatiotemporal features of the ordinary changes can be captured. Then, we apply statistical tests on the training examples to extract spatiotemporal signatures of the ordinary change patterns. Finally, we estimate the existence of the salient change in given test input by interpolating from the training samples.

three major processing blocks: i) spatiotemporal correlation analysis, ii) feature extraction, and iii) salient change detection by interpolating from the ordinary change patterns.

Pixels, which belong to a region of an ordinary change pattern, are typically correlated in space and/or time among a set of consecutive frames. This correlation stems from the repetitive nature of the ordinary change patterns, and it induces spatiotemporal signatures specific to each local ordinary change pattern. The human visual system directly uses the dynamic information to identify the entities that surround us [135, 190]. Stone [172] demonstrated that such spatiotemporal signatures, which are encoded in the representation of an object,

could be used for object recognition. Similarly, we propose that one can make use of the spatiotemporal signature to discriminate salient changes from the ordinary change in a given local region. Capturing spatiotemporal signatures requires three-dimensional data processing, and the image pixel plane is usually not considered as suitable for extracting spatiotemporal features [67]. Dollar et al. [57] presented that the direct three-dimensional counterparts to commonly used two-dimensional descriptors are inadequate for spatiotemporal features. Instead, we propose to transform local three-dimensional regions containing ordinary change patterns to a transform domain where the pixels in the region are decorrelated. Thereby, we can capture spatiotemporal signatures which are unique to each local ordinary change. This will allow us to learn and to recognize ordinary change patterns. Then, when a change sample which is unrelated to ordinary change patterns occurs, the framework can label it as a salient change.

Because of the amount of the data in video processing, the chosen transform should be fast and simple to implement such as linear transforms. Orthogonal linear transforms redistribute the energy stored in the input data and decorrelate it [5]. They are successfully applied to the methods based on compact representations, such as image compression [159], watermarking [95], face recognition [49], and speech processing [7]. In this study, we use the data compaction capability of orthogonal linear transforms in order to exploit spatiotemporal signatures of local ordinary change patterns. Estimation of the optimal transform for each

local ordinary change pattern is not a trivial process. In terms of energy compaction, Karhunen-Lo'eve transform (KLT) has the best efficiency; however, KLT has high computational complexity and is mostly of interest for theoretical and historical reasons [4, 5, 159]. We propose to estimate not the optimal one but a suitable transform for each local ordinary change pattern from a pool of linear transformations which have complementary orthogonal basis vectors. A transform is considered as suitable for a local ordinary change pattern if the transform domain provides a compact representation of the local ordinary change pattern. Our approach is built up on localization by applying a data decomposition model that generates local three-dimensional blocks. Therefore, the estimated change mask may suggest if there is a salient change within the three-dimensional block, but we need to examine each frame region in the block to obtain individual pixels belonging to the regions of change in each frame. This may cause blocking artifacts. In order to compensate for these artifacts, we apply Markov random field regularization [113]. To evaluate the performance of the proposed method, experiments are performed using the test videos provided by *ChangeDetection.net* [69]. The quantitative comparison of the detection results from the proposed framework to other methods demonstrates improved accuracy.

4.2.1 Spatiotemporal Signature Analysis

The proposed framework exploits spatiotemporal signatures of local regions in consecutive frames in order to detect the salient changes. This requires three-dimensional block based processing and extends the change detection problem from comparing *two regions* between two frames to comparing *two sets of consecutive regions* between two sets of frames (i.e., sequence to sequence comparison).

An ordinary change in a local region is typically correlated in space and/or time among consecutive frames. This correlation induces a spatiotemporal signature specific to the ordinary change pattern in the region. Our goal is to find a representation space where we can capture this spatiotemporal signature.

4.2.2 Data Decomposition

In the literature, the term "data decomposition" has been used to refer to a number of different concepts. Throughout this study, we will use the term "data decomposition" to refer to the process of spatially splitting the three-dimensional input data into subgroups of the same size (i.e., cubes). This decomposition approach is usually used in data compression [159] and parallel programming [64] applications. In this study, we need the data decomposition to divide a frame sequence into three-dimensional subblocks such that local spatiotemporal signatures can be extracted. Before the data decomposition, we apply median filter and map the intensity values in a frame such that about 1% of data is

saturated at low and high intensities of the frame. The aim of the filtering and the intensity adjustment is to compensate for illumination variations between the data acquisitions.

Let $\mathbf{V}$ denote a sequence of frames, with $\mathbf{V} = \{F_1, \ldots, F_\tau, F_{\tau+1}, \ldots, F_\mathfrak{T}\}$. Let $\mathbf{V}_o$ be a subset of $\mathbf{V}$, including all the frames between F_1 and F_τ. We are given that the subset $\mathbf{V}_o$ contains only ordinary changes. Our focus in this study is solely on the change detection problem, but not the background modeling problem. This is a reasonable assumption for the change detection problem where two states of an entity are under investigation. The rest of $\mathbf{V}$ may contain ordinary changes, salient changes, or both. We first divide each frame into 8 by 8 pixels regions in order to improve the localized correlation. Then, 8 consecutive frames are grouped to form a stack as shown in Figure 4.5.

Let § denote the set of stacks, with $\S = \{\mathbf{S}_k\}_{k=1}^{K}$, where $K = \tau/8$. Each stack $\mathbf{S}_k$ is composed of $8 \times 8 \times 8$ blocks called *cube*s. Cube elements in the stack $\mathbf{S}_k$ are denoted by c_{ij}^k, where $i = 1, \ldots, I$, $j = 1, \ldots, J$, I, and J are the number of the cubes in vertical and horizontal directions, respectively. We anticipate that the spatiotemporal signature of every cube element in a stack may be different. After the data decomposition, a set of τ frames turns into a set of K stacks, each of which contains $I*J$ cube elements as shown in Figure 4.5(b). Cubes in different stacks are said to be *corresponding cubes* if $u = p$ and $v = r$ for $c_{uv}^{\kappa_1}$ and $c_{pr}^{\kappa_2}$, where $\kappa_1 \neq \kappa_2$. We perform a further grouping and collect the corresponding cube elements in *corresponding cube set*s. Let $\mathbf{C}_{ij}$ denote a corresponding cube set, where i and j

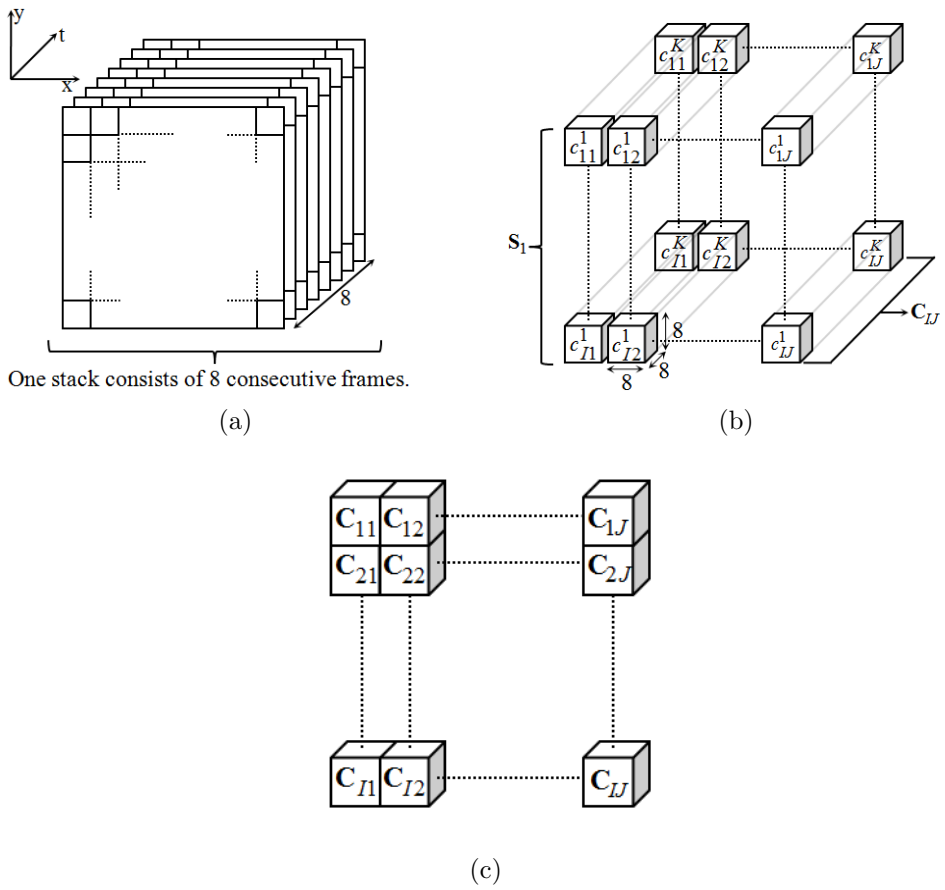


Figure 4.5: Illustration of data decomposition. Each frame is divided into 8 by 8 pixels regions, and every 8 consecutive frames are stacked as in (a). Stacking is performed across all the frames. $\mathbf{S}_1$ denotes the first stack. A stack is composed of $8 \times 8 \times 8$ blocks, called *cube*s. In (b), cube elements in the stack $\mathbf{S}_k$ are denoted by c_{ij}^k, where i=1,. . . ,I; j=1,. . . ,J; and I and J are the number of the cubes in vertical and horizontal directions, respectively. K is the total number of the stacks. In (c), we present corresponding cube sets. A corresponding cube set is composed of corresponding cube elements in different stacks. For example, the corresponding cube set $\mathbf{C}_{IJ}$ in (b) consists of the cube elements $\{c_{IJ}^1, \ldots, c_{IJ}^K\}$. We expect the cube elements in a corresponding cube set to share similar spatiotemporal signatures.

refer to the location of the cube elements in the data decomposition approach as shown in Figure 4.5(b). Namely, the entire frame set $\mathbf{V}_o$ becomes a $I \times J$ grid of corresponding cube sets as illustrated in Figure 4.5(c). Every corresponding cube set $\mathbf{C}_{ij}$ is considered as the summary of ordinary change pattern in the local region at i and j. We expect the cube elements in a corresponding cube set to share similar spatiotemporal signatures.

We now need to estimate a suitable transform for each corresponding cube set so that we can exploit their spatiotemporal signatures.

4.2.3 Adaptive Transform Estimation for the Ordinary Change Patterns

Let us define a $I \times J$ matrix $\mathbf{T}$ of transforms. An element T_{ij} of $\mathbf{T}$ refers to the transform that is suitable for the corresponding cube set $\mathbf{C}_{ij}$. A suitable transform is defined as the one where the transformed values are independent of one another, and the energy is compacted on a few transformed values regardless of their relative locations as opposed to the general assumption in data compression methods. Estimating $\mathbf{T}$ is not a straightforward procedure. Orthogonal linear transforms, which are widely used in data compression methods [44, 144, 191], redistribute the energy stored in the input data and provide a compact representation. The principle of minimum description length in the model selection [79] suggests that methods that yield compact representations should be employed

for recognition purposes. Accordingly, we employ three orthogonal linear transforms as the base transforms: i) discrete cosine transform (DCT) [4], ii) Walsh-Hadamard transform (WHT) [6], and iii) Slant transform (ST) [145]. DCT, WHT, and ST are incorporated together because they have complementary basis vectors that enable the framework to capture different types of ordinary change patterns. In addition, Fridrich [63] presents that DCT, WHT, and ST achieve an energy compaction efficiency close to the that of Karhunen-Lo'eve transform. DCT is a sinusoidal transform which is widely used in applications requiring compact representations such as video compression [110], watermarking [93], and content based retrieval [136]. WHT is a non-sinusoidal transform having basis vectors that are rectangular or square waves with values of $+1$ or -1; therefore, it can represent patterns with sharp discontinuities more accurately using fewer values than DCT. In the literature, WHT has been applied to variety of applications because of its low computational cost, such as image compression [100], speech recognition [137], segmentation [187], and face recognition [60]. The basis vectors of ST are derived from sawtooth waveforms and considered as a good complement to WHT [58]. Compared to DCT and WHT, ST has been applied to a limited number of problems, mostly to image watermarking [87, 188, 207]. Nevertheless, as it is seen our experimental results, SL is found to represent more ordinary change patterns than WHT does. Needless to say, we observe that DCT is superior to ST. These three transforms constitute our base transform space. Our goal is to estimate the most suitable transform to represent the ordinary change pattern encapsulated in each corresponding cube set using the spatial signatures of ordinary change patterns

and the base transform space. Namely, we need to estimate the transform T_{ij} that yields the most compact representation $\mathfrak{S}_{ij}$

$$\mathfrak{S}_{ij} = \mathbf{C}_{ij} T_{ij} \tag{4.7}$$

where T_{ij} is assigned to one of these three base transforms depending the ordinary change pattern in $\mathbf{C}_{ij}$.

Various performance criteria [150], such as energy packing efficiency, coding gain, and decorrelation efficiency, are proposed to determine how efficient a particular transform for a given input. Kitajima [103] defines the energy packing efficiency as the ratio of energy contained in the first retained m transformed values to the total energy, where m is coding dependent parameter. Nevertheless, one cannot guarantee that most of the input energy are always packed into low-frequency components or a specific sub-band of a generalized spectrum. In addition, Yip and Rao [202] presented that the energy packing efficiency may not be a suitable measurement for determining the efficacy of the discrete unitary transforms. The coding gain is the ratio of the arithmetic mean to the geometric mean of the transform coefficient variances. It measures how well a transform compacts energy into a small number of coefficients, regardless of frequency type. The decorrelation efficiency is the ratio of the sum of off-diagonal elements of the coefficient covariance matrix to the sum of the input values. Computation of the decorrelation efficiency and the coding gain require to estimate the covariance matrix of the transformed values by making assumptions for the inter-pixel

relationships. Accurately estimating the *true* covariance matrix from a limited number of data samples is very challenging problem [178]. Instead of such measurements, we propose a *novel* energy compaction criterion, called *compactness coefficient*, for the comparison of the different discrete transforms.

We call a transform *compact* if the energy of transformed values does not uniformly distributed in the transform domain. Let D be a set of real valued numbers, with $D = \{d_1, \ldots, d_N\}$, where N is the number of the data points. Our goal is to estimate the most suitable transform available for D from the pool of base transforms. In our setting, D refers to a cube element in a corresponding cube set. Out of the base transforms provided, the most suitable transform for D is defined as the one having transformed values where the energy in D is the least scattered in the transform domain. Let Ω denote the set of transformed values of D, with $\Omega = \{\omega_1, \ldots, \omega_N\}$. Let E be the total energy stored in Ω, with

$$\mathrm{E} = \sum_{i=1}^{N} \omega_i^2. \tag{4.8}$$

In terms of energy scattering, the worst case is a uniformly distributed energy across the transformed values. For such a case, all ω_i^2 values will be the same, namely

$$\omega_i^2 = \frac{\mathrm{E}}{N} \text{ for each } i. \tag{4.9}$$

Let us normalize Ω based on the energy stored in each transformed value so that

E is going to be 1. Accordingly, energy of the elements of Ω will be as follows

$$\frac{\omega_i^2}{E} = \frac{\frac{\mathrm{E}}{N}}{E} = \frac{1}{N} \text{ for each } i. \tag{4.10}$$

We shall use this extreme case (i.e., uniformly distributed energy in the transform domain) to define the *compactness coefficient*. Let ξ_s denote the compactness coefficient. ξ_s describes how compact the energy of D is redistributed in the transform domain. We will use ξ_s to assess the efficacy of each base transform for the input data set D.

For a moment, let us focus on an ordinary input data set D and forget about the extreme case explained above. We first compute the transformed values Ω using the transformation matrix. We do not have any assumption about the energy redistribution in Ω. We compute the total energy E of Ω using the Equation 4.8. Then, we apply the energy normalization and obtained the set of normalized transformed values $\grave{\Omega}$. The relationship between the elements of Ω and $\grave{\Omega}$ is straightforward: $\grave{\omega}_i = \frac{\omega_i}{E}$ for $i = 1, ..., N$. Finally, we define the compactness coefficient ξ_s as follows:

$$\xi_s \triangleq \sum_{i=1}^{N} \left(\frac{1}{N} - \grave{\omega}_i \right)^2 \tag{4.11}$$

where $\xi_s \in [0, 1 - \frac{1}{N}]$. If a transform can compact all the energy of the input in one single transformed value, the transform can be considered as the most suitable one for the input data D. In such a case, only one element of $\grave{\Omega}$ will be 1, while

the rest is zero. Accordingly, the value of ξ_s would be

$$\xi_s = \sum_{i=1}^{N} \left(\frac{1}{N} - \dot{\omega}_i \right)^2 = (N-1) * \frac{1}{N}^2 + (\frac{1}{N} - 1)^2 = 1 - \frac{1}{N}. \tag{4.12}$$

On the other hand, when the energy is distributed uniformly, ξ_s becomes

$$\xi_s = \sum_{i=1}^{N} \left(\frac{1}{N} - \frac{1}{N} \right)^2 = 0. \tag{4.13}$$

Namely, the greater the compactness coefficient, the more compact the transformed values. We can now establish a two-step model to estimate the most suitable transform from the base transforms DCT, WHT, and ST for a corresponding cube set $\mathbf{C}_{ij}$. First, we compute compactness coefficients ξ_s^{DCT}, ξ_s^{WHT}, and ξ_s^{ST} for the base transforms for each cube element c_{ij}^k in $\mathbf{C}_{ij}$, where $k = 1, \ldots, K$. Then, the transform having the largest compactness coefficient value is estimated as the most suitable transform for the cube element c_{ij}^k

$$T_{ij}^k = \arg\max \xi_s^{BT}, \text{ for BT=\{DCT, WHT, ST\}}. \tag{4.14}$$

This process is performed for all the cube elements in $\mathbf{C}_{ij}$ and results in K transforms for the corresponding cube set $\mathbf{C}_{ij}$: $\{T_{ij}^1, ..., T_{ij}^K\}$. Then, the transform that is the most common amongst estimated K transforms is assigned to T_{ij}. We repeat the process for all corresponding cube sets. In the rest of the framework, a local change pattern in a corresponding cube set is represented by the estimated

transform for that corresponding cube set.

4.2.4 Significant Transform Coefficients

Throughout the remainder of this dissertation, let us call the transformed values *transform coefficients*. With a suitable transform, majority of the transform coefficients tend to have small values. Our goal is to find a *significant subset* of transform coefficients for each corresponding cube set. A significant subset should contain a small number of coefficients that contribute to most of the energy. The estimated transform and *significant subset* of the transform coefficients would be considered as the *spatiotemporal signature* of the ordinary change pattern captured in the corresponding cube set.

In several approaches employing orthogonal linear transforms, it is assumed that values of a specific predefined subset of the transform coefficients can be neglected for different types of input data [11, 159]. The accuracy of this assumption solely relies on the characteristics of the input, and it may cause loss of distinctive features. Instead, we propose to estimate a significant subset based on the energy of each coefficient in cube elements of a corresponding cube set. We suggest that an adaptively selected subset of transform coefficients for a corresponding cube set will exploit spatiotemporal signature of the ordinary change pattern encapsulated in the corresponding cube set. Let Ω_{ij}^{k} be a set of transform coefficients computed for the cube element c_{ij}^{k} in the corresponding cube set $\mathbf{C}_{ij}$

$$\Omega_{ij}^{k} = \{\omega_{ij}^{k,1}, \ldots, \omega_{ij}^{k,N}\} \tag{4.15}$$

where N is the number of transform coefficients. For example, value of N for a cube element of the size $8 \times 8 \times 8$ would be 2^9. We compute Ω_{ij}^{k} for all the cube elements $k = 1, \ldots, K$ in $\mathbf{C}_{ij}$. Transform coefficients for every cube element are normalized to carry the unit energy. Let $\grave{\Omega}_{ij}^{k}$ denote the set of normalized transform coefficient values, with $\grave{\Omega}_{ij}^{k} = \{\grave{\omega}_{ij}^{k,s}\}_{s=1}^{N}$. Transform coefficients $\grave{\omega}_{ij}^{\kappa_1,s}$ and $\grave{\omega}_{ij}^{\kappa_2,s}$ in different cube elements of the corresponding cube set $\mathbf{C}_{ij}$ are defined as *corresponding coefficients*, where $\kappa_1 \neq \kappa_2$. For N transform coefficients, we have N corresponding coefficient sets, denoted by $\mathfrak{cc}_{ij}^{s} = \{\grave{\omega}_{ij}^{1,s}, \grave{\omega}_{ij}^{2,s}, \ldots, \grave{\omega}_{ij}^{K,s}\}$ and $s = 1, \ldots, N$. Using these correspondences, let us define a parameter ς_{ij}^{s} to assess significance to each corresponding coefficient set $\mathfrak{cc}_{ij}^{s}$ in $\mathbf{C}_{ij}$ using the average normalized energy. ς_{ij}^{s} is computed for each s as follows

$$\varsigma_{ij}^{s} = \frac{1}{K} \sum_{k=1}^{K} \grave{\omega}_{ij}^{k,s}. \tag{4.16}$$

This results in $\varsigma_{ij}^{s} \in [0, 1]$ values, where $\sum_{s=1}^{N} \varsigma_{ij}^{s} = 1$. Finally, we use an iterative forward selection algorithm [8] to form one significant subset for each corresponding cube set. We start with no coefficients and add them one by one based on ς_{ij}^{s} values, at each step adding the one that stores the most energy, until any further addition does not increase the total energy in the subset or increases it only slightly. This generates a significant subset $\mathfrak{C}_{ij} = \{x_{ij}^{l}\}_{l=1}^{L}$, where $L << N$ for

$\mathbf{C}_{ij}$. Elements of $\mathfrak{C}_{ij}$ represent coordinates of the selected transform coefficients for $\mathbf{C}_{ij}$. The significant subset and the estimated transform for a corresponding cube set is considered as the spatiotemporal signature of the ordinary change pattern captured in the corresponding cube set. Accordingly, in the rest of the framework only the transform coefficients in the significant subsets are used for change detection purposes.

4.2.5 Statistical Properties

The advantage of using statistical properties compared to strategies assuming a priori parametric distribution is that one can distinguish fluctuations due to the fact that the assumed model may not be valid over the whole input space. In our setting, a corresponding cube set $\mathbf{C}_{ij}$ is considered as a unified structure that captures the local ordinary change pattern within itself. $\mathbf{C}_{ij}$ is specified by the estimated base transform $T_{ij} \in \mathbf{T}$ and its significant subset $\mathfrak{C}_{ij}$. Let us define a function $\mathfrak{L}(l,k)$ that maps coordinates in $\mathfrak{C}_{ij}$ to actual transform coefficient values in the cube elements of $\mathbf{C}_{ij}$: $\mathfrak{L}(x_{ij}^{l},k) \rightarrow \omega_{ij}^{k,s}$ for $l = 1,\ldots,L$ and $s = 1,\ldots,N$. One should note that the distribution of each significant transform coefficient may be different, and the estimation of each unique distribution is not a trivial process. Instead, we construct a maximum likelihood model by interpolating from the training instances.

Let $\mathfrak{M}$ be a $I \times J$ matrix of the number of significant coefficients, with $\mathfrak{M} = \{m_{ij}\}_{i=1,j=1}^{I,J}$. m_{ij} is the number of significant coefficients in the significant subset

$\mathfrak{C}_{ij}$ for the corresponding cube set $\mathbf{C}_{ij}$. Let $\tilde{\mathfrak{u}}$ denote a vector parameter called *unbiased mean*. $\tilde{\mathfrak{u}}_{ij}^{k} \in \mathbb{R}^{m_{ij}}$ is defined for the cube element k in $\mathbf{C}_{ij}$. An element $\tilde{\mathfrak{u}}_{ij}^{k,l}$ of $\tilde{\mathfrak{u}}_{ij}^{k}$ is calculated as follows

$$\tilde{\mathfrak{u}}_{ij}^{k,l} = \frac{1}{K-1} \sum_{\kappa=1}^{K} \mathfrak{L}(x_{ij}^{l}, \kappa) \text{ and } \kappa \neq k. \tag{4.17}$$

We compute $\tilde{\mathfrak{u}}_{ij}^{k,l}$ for $l = 1, \ldots, m_{ij}$. We can represent every cube element in $\mathbf{C}_{ij}$ using the significant coefficients. Let $\mathcal{C}_{ij}^{k}$ be a m_{ij}–dimensional vector of significant coefficient values for the cube c_{ij}^{k}. We define a deviation vector $\mathfrak{d}_{ij}^{k} \in \mathbb{R}^{m_{ij}}$, which describes the deviation of $\mathcal{C}_{ij}^{k}$ from its unbiased mean $\tilde{\mathfrak{u}}_{ij}^{k}$ as follows

$$\mathfrak{d}_{ij}^{k} = |\mathcal{C}_{ij}^{k} - \tilde{\mathfrak{u}}_{ij}^{k}|, \tag{4.18}$$

where $k = 1, \ldots, K$. We then calculate standard deviation σ_{ij}^{k} and mean μ_{ij}^{k} of the elements of $\mathfrak{d}_{ij}^{k}$. In the next section, $\mathcal{C}_{ij}^{k}$, $\mathfrak{d}_{ij}^{k}$, σ_{ij}^{k}, and μ_{ij}^{k} for $k = 1, \ldots, K$ are to used to construct a maximum likelihood model for the salient change detection.

4.2.6 Salient Change Detection

Let us recall the given frame set $\mathbf{V} = \{F_1, \ldots, F_\tau, F_{\tau+1}, \ldots, F_{\mathfrak{T}}\}$. We used the subset $\mathbf{V}_o = \{F_1, \ldots, F_\tau\}$ to estimate spatiotemporal signatures of the local ordinary change patterns. We will now analyze the changes in the rest of the frames, namely in $\mathbf{V} - \mathbf{V}_o$. Let us assume that we initially process the first 8 frames

$\{F_{\tau+1}, \ldots, F_{\tau+8}\}$. As described in the Section 4.2.2, we group the frames to form the stack $\mathbf{S}_{test}$ and and decompose the stack into the cube elements c_{ij}^{test}. We compute the transform coefficients of c_{ij}^{test} using the estimated base transform $T_{ij} \in \mathbf{T}$. Then, the significant coefficient subset $\mathfrak{C}_{ij}$ and $m_{ij} \in \mathfrak{M}$ are used along with the mapping function $\mathfrak{L}(l, k)$ to construct a $m_{ij}-$dimensional descriptor $\mathcal{C}_{ij}^{test}$ for each i and j. We then compute the deviation of $\mathcal{C}_{ij}^{test}$ from the training samples $\mathcal{C}_{ij}^{k}$:

$$\mathfrak{d}_{ij}^{k,test} = |\mathcal{C}_{ij}^{k} - \mathcal{C}_{ij}^{test}|, \tag{4.19}$$

for $k = 1, \ldots, K$. We calculate standard deviation $\sigma_{ij}^{k,test}$ and mean $\mu_{ij}^{k,test}$ of the deviation values in $\mathfrak{d}_{ij}^{k,test}$.

We cast the analysis of the salient change as a significance testing. The decision rule for whether or not a salient change has occurred within a given cube corresponds to choosing one of two competing hypotheses: the null hypothesis $\mathcal{H}_0$ or the alternative hypothesis $\mathcal{H}_1$, corresponding to ordinary change and salient change decisions, respectively. The null hypothesis $\mathcal{H}_0$ is characterized using the training samples in $\mathbf{V}_o$, which are assumed to have only ordinary change patterns. A significance test on the difference between an observation and the training samples is performed to assess how well the null hypothesis describes the observation, and this hypothesis is accepted or rejected.

Given $\mathcal{H}_0$, let X, Y be two random variables with means μ_X, μ_Y, standard deviations σ_X, σ_Y, and correlation coefficient ρ_{XY}. The bivariate inequality of

Lal [107] is given by

$$P(\lambda_{L_X} < X < \lambda_{U_X}, \lambda_{L_Y} < Y < \lambda_{U_Y} | \mathcal{H}_0) \geq P_{XY}, \text{ and} \tag{4.20}$$

$$P_{XY} = 1 - \frac{1}{2k_X^2 k_Y^2}(k_X^2 + k_Y^2 + \sqrt{(k_X^2 + k_Y^2)^2 - 4\rho^2 k_X^2 k_Y^2}), \tag{4.21}$$

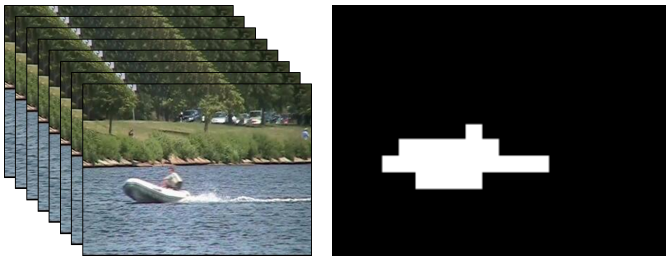
where $\lambda_{L_X} + \lambda_{U_X} = 2\mu_X$, $\lambda_{L_Y} + \lambda_{U_Y} = 2\mu_Y$, $k_X = (\lambda_{U_X} - \lambda_{L_X})/2\sigma_X$, and $k_Y = (\lambda_{U_Y} - \lambda_{L_Y})/2\sigma_Y$. In the Equation 4.20, P_{XY} gives a lower bound for the joint probability of the interval $[\lambda_{L_X}, \lambda_{U_X}]$ around μ_X and the interval $[\lambda_{L_Y}, \lambda_{U_Y}]$ around μ_Y for the random variables X and Y. We propose that if X and Y are dependent events, we expect P_{XY} to be large for the same interval $[\lambda_{L_X} = \lambda_{L_Y}, \lambda_{U_X} = \lambda_{U_Y}]$ around μ_X and μ_Y for X and Y. Accordingly, we define a symmetric interval $\lambda_{L_X} = \lambda_{L_Y} = (\mu_X + \mu_Y)/2 - 2 * (\sigma_X + \sigma_Y)$ and $\lambda_{U_X} = \lambda_{U_Y} = (\mu_X + \mu_Y)/2 + 2 * (\sigma_X + \sigma_Y)$ for X and Y. We can use the value of P_{XY} to estimate the likelihood of X and Y to be dependent random events. In our change detection setting, we consider the elements of the pair $(\mathfrak{d}_{ij}^k, \mathfrak{d}_{ij}^{k,test})$ as the values of the two random variables X and Y, with the means $(\mu_{ij}^k\ ,\ \mu_{ij}^{k,test})$ and the standard deviations $(\sigma_{ij}^k, \sigma_{ij}^{k,test})$. If $\mathfrak{d}_{ij}^{k,test}$ is found to be independent from $\mathfrak{d}_{ij}^k$, one can deduce that there is a salient change in the cube c_{ij}^{test}. Using the Equation 4.21, we can compute a joint probability P_{XY}^k for a stack $\mathbf{S}_k$ in the training samples, where $k = 1, \ldots, K$. Because the Equation 4.20 provides a lower bound but not the actual probability, we can compute an average P_{XY} for all the stacks in the training samples by

$$P_{XY} = \frac{1}{K} \sum_{k=1}^{K} P_{XY}^{k}. \tag{4.22}$$

4.2.7 Change Detection at Pixel Resolution

When a change having spatiotemporal signature different from ordinary change patterns is detected in a cube, the proposed method suggests that there may be salient change within the regions comprising the cube. At the frame level, this corresponds to a two-dimensional projection of spatiotemporal changes within the stack of 8 consecutive frames (Figure 4.6(a)). This summary image is called *binary change mask* (Figure 4.6(b)), where 1 and 0 indicate the salient and ordinary change, respectively.

We use the binary change mask to analyze the mid-frames to avoid large blocking artifacts. Then, we use the two-dimensional version of the estimated base transform within a window around pixels having the value of 1 in the binary change mask. The block-based nature of our approach may cause noise at the pixel level. To overcome this limitation, the resulting change mask is assumed to be a Markov random field, where each node (i.e., pixel) is connected using 4-connected neighborhood: 1) up, 2) down, 3) left, and 4) right. Then, the node is labeled as either salient change or ordinary change based on the probability maximization achieved by the Markov Random Field regularization [113]. We repeat the process by sliding the frame stack in order to evaluate all frames.



(a) Input frames (b) Change mask

(c) I_{2000} (d) Output (e) Ground truth

Figure 4.6: Illustrative example of a binary change mask. In (b), the binary change mask for 8 consecutive frames in (a) is presented. The binary change mask is a two dimensional projection of spatiotemporal changes in these 8 frames. In (c),(d), and (e), we present input I_{2000}, salient changes detected, and the ground truth, respectively. The gray levels in ground truth are 0:*ordindary change*, 255:*salient change*, 85:outside region of interest, and 170:unknown motion [69]. We are interested in detecting pixels labeled as *salient change*.

Chapter 5

Experimental Results and Discussion

We carried out several experiments to justify the effectiveness of each method proposed in this dissertation. We here present the results topic by topic.

5.1 Training and Testing Videos

The proposed methods were developed and tested on 19 videos obtained from different sources [56, 69, 73]. We present different properties of the videos in Table 5.1. All the videos were recoded in outdoor environments during daylight or at night. The distance between the camera and the objects in the scene usually varies throughout videos because of either camera or object movement.

ID	Video Name	Frame Size	Number of Frames	Frame Type	Data Acquisition	Obtained from
$\mathbb{V}^{1r}$	Forest	640×480	3,256	Stereo	Mobile	[73]
$\mathbb{V}^{1s}$	Forest	640×480	3,256	Stereo	Mobile	[73]
$\mathbb{V}^{2r}$	Game Field	640×480	1,385	Stereo	Mobile	[73]
$\mathbb{V}^{2s}$	Game Field	640×480	1,854	Stereo	Mobile	[73]
$\mathbb{V}^{3r}$	Parking Lot	640×480	1,381	Stereo	Mobile	[73]
$\mathbb{V}^{3s}$	Parking Lot	640×480	1,855	Stereo	Mobile	[73]
$\mathbb{V}^{4r}$	Soccer Field	640×480	3,256	Stereo	Mobile	[73]
$\mathbb{V}^{4s}$	Soccer Field	640×480	3,256	Stereo	Mobile	[73]
V^{5r}	Broadway	640×480	1,594	Monocular	Mobile	[160]
V^{5s}	Broadway	640×480	3,122	Monocular	Mobile	[160]
V^{6r}	Highway2	576×432	2,281	Monocular	Mobile	[56]
V^{6s}	Highway2	576×432	1,432	Monocular	Mobile	[56]
V^{7r}	Night	576×460	999	Monocular	Mobile	[56]
V^{7s}	Night	576×460	448	Monocular	Mobile	[56]
V^{8}	Boats	320×240	7,999	Monocular	Stationary	[69]
V^{9}	Canoe	320×240	1,189	Monocular	Stationary	[69]
V^{10}	Fall	720×480	4,000	Monocular	Stationary	[69]
V^{11}	Fountain01	432×288	1,184	Monocular	Stationary	[69]
V^{12}	Fountain02	432×288	1,499	Monocular	Stationary	[69]
V^{13}	Overpass	320×240	3,000	Monocular	Stationary	[69]

Table 5.1: List of test and training videos used in this dissertation.

The stereo video pairs $\mathbb{V}^{1r} - \mathbb{V}^{1s}$, $\mathbb{V}^{2r} - \mathbb{V}^{2s}$, $\mathbb{V}^{3r} - \mathbb{V}^{3s}$, and $\mathbb{V}^{4r} - \mathbb{V}^{4s}$ were acquired by a mobile camera platform in four different environments. For each pair, the secondary video was taken after objects of different sizes and textures were placed in the scene, and the reference video was taken without the objects. We used these videos for the verification of temporal alignment, dominant plane estimation, disparity enhancement, and change detection methods. The monocular video pairs $V^{5r} - V^{5s}$, $V^{6r} - V^{6s}$, and $V^{7r} - V^{7s}$ were recorded by mobile camera platforms in three different environments. We used these videos for the verification of the video synchronization method. Different from the rest of the videos, the videos $V^{8r} - V^{8s}$ were captured at night. The monocular videos V^{8}, V^{9}, V^{10}, V^{11}, V^{12}, and V^{13} were captured by a stationary camera in outdoor environments where there are a lot of altering regions, which are not considered as change, in the background. We used these videos for the verification of the change detection module.

Some high resolution frames are reduced to the lower resolution (e.g., from 640×480 to 384×288 pixels) using anti-aliased filtering and subsampling. There are two reasons behind the subsampling. First, it decreases the required computational time for the algorithms. Second, the subsampling improves the accuracy. For example, high-resolution frames produce disparity maps which contain too much detail. For the subsequent modules in the framework, when it requires, we resize the output image back the original size. The experiment results are evaluated both visually and/or quantitatively depending on the test case.

5.2 Disparity Estimation and Refinement

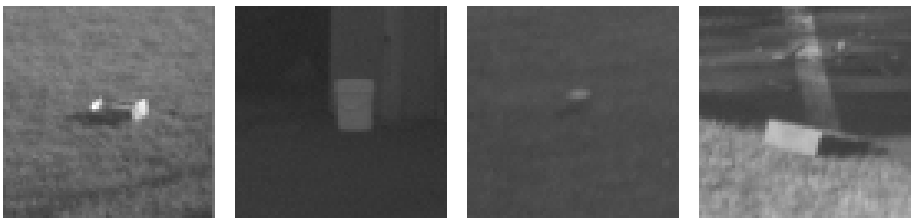
In this section, we will present the results obtained from the experimental test that mainly employ the stereo videos.

5.2.1 Results of Disparity Map Refinement Method as a Post-processing Approach

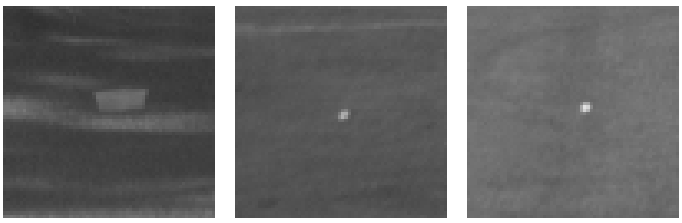
In Section 3.2.5, we presented a disparity map refinement method using consecutive estimated disparity maps. We evaluated the proposed method using four stereo video pairs (i.e., $\mathbb{V}^{1r} - \mathbb{V}^{1s}$, $\mathbb{V}^{2r} - \mathbb{V}^{2s}$, $\mathbb{V}^{3r} - \mathbb{V}^{3s}$, and $\mathbb{V}^{4r} - \mathbb{V}^{4s}$) that were acquired by a mobile camera platform in different real outdoor environments under different illumination conditions at different times. For each environment, we have two videos called $r$eference and $s$econdary where the secondary videos were recorded after seven objects of different sizes and textures (Figure 5.1) were placed in the environment, and the reference videos were taken without the presence of the objects.

In Table 5.2, we report the number of frames in which an object can be seen and the number of frames where we detected the object successfully. Results of detecting the change with and without disparity map refinement are presented and clearly show the advantage of the disparity refinement method.

For the objects O_2, O_3, and O_5 our change detection module failed for some



(a) O_1 (b) O_2 (c) O_3 (d) O_4



(e) O_5 (f) O_6 (g) O_7

Figure 5.1: Example images of the objects which were places in the outdoor environment.

of the frames because of the problems related to spatial registration. For the objects O_1 and O_6, most of the failures were caused by inaccurate estimation of the disparity maps. For objects O_4 and O_7, our ground plane segmentation method sometimes produced incorrect results, such as labeling closer objects as part of the ground plane. Nevertheless, since this was the same case for both reference and secondary frames, the texture comparison module was successfully able to detect the objects.

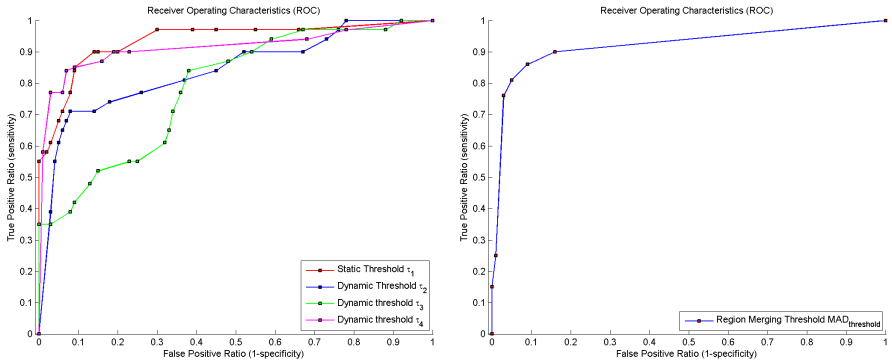
We performed a second experiment to observe the accuracy of the proposed framework when we do not place any objects in the scene. This was based on analyzing frames where there were no objects introduced in the scene. Our framework could recognize that there was no change with 92.18% accuracy.

Video ID	**Object ID**	**Viewable**	**Detected**		**Accuracy (%)**	
			Original	Refined	Original	Refined
$\mathbb{V}^{1s}$	O_1	68	53	59	77.94	86.76
$\mathbb{V}^{1s}$	O_2	11	5	8	45.45	72.73
$\mathbb{V}^{2s}$	O_3	24	15	20	62.50	83.33
$\mathbb{V}^{3s}$	O_4	21	12	17	57.14	80.95
$\mathbb{V}^{4s}$	O_5	43	26	34	60.47	79.07
$\mathbb{V}^{4s}$	O_6	61	23	51	37.70	83.61
$\mathbb{V}^{4s}$	O_7	57	45	53	78.95	92.98
	Total	**285**	**179**	**242**	**62.81**	**84.91**

Table 5.2: Results on the test set. Disparity map refinement module notably increases the change detection performance of the system.

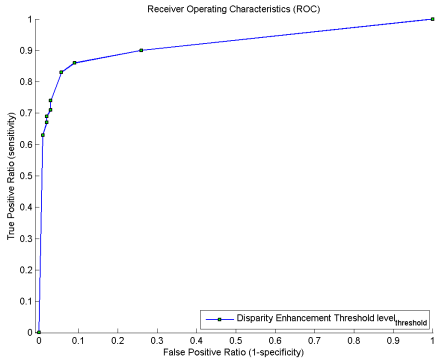
Our third experiment was performed to assess how well the different decision threshold models describe the change. The threshold $\mathbf{MAD}^{T-G}_{threshold}$ presented in Section 4.1.2 can be computed to produce a desired true positive and false alarm rate (i.e., false positive rate). The test is carried out using the models given in Table 5.3, where Max is the the maximum value of combined metric $\mathbf{m}$ values calculated for each block in the registered ground plane images, μ is mean of $\mathbf{m}$ values, and σ is standard deviation of $\mathbf{m}$ values. k_1 and k_2 are parameters ranging from 0.00 to 3.00. Similar tests are performed to observe how well the disparity enhancement threshold $level_{thredhold}$ (Section 3.2.5) and the region merging threshold $\mathbf{MAD}_{threshold}$ (Section 3.2.7) perform. We use Receiver Operating Characteristic (ROC) analysis to compare the different decision thresholds and to observe effects of the disparity enhancement and region merging modules on the system performance (Figure 5.2).

It is important to keep in mind that every change detection methodology in



(a) $\mathbf{MAD}_{threshold}^{T-G}$

(b) $\mathbf{MAD}_{threshold}$



(c) $\mathbf{level}_{threshold}$

Figure 5.2: ROC analysis: (a) ROC curves for the various change detection decision threshold models, (b) and (c) region merging and disparity enhancement components of the change detection framework are evaluated individually.

one way or another must determine if a given value of change is of sufficient strength to actually be called *change*. Figure 5.2 illustrates the ability to detect the changes of interest for the various thresholds in the framework. The closer the curve of a threshold approaches the upper left corner of the diagram, the better the threshold performs. Sensitivity and false positives are typically at odds with each other and are a strong function of the decision threshold. High

Threshold	**Values/Model**
Static $\mathbf{MAD}^{T-G}_{threshold}$ (τ_1)	2.10, 2.15, 2.25, 2.50, 2.87, 2.90, 3.20, 3.60, 4.00, 4.50, 5.00, 5.70
Dynamic $\mathbf{MAD}^{T-G}_{threshold\ 1}$ (τ_2)	$\mu * \left(k_1 + \frac{\sigma}{\mu - \sigma}\right)$
Dynamic $\mathbf{MAD}^{T-G}_{threshold\ 2}$ (τ_3)	$k_1 * Max$
Dynamic $\mathbf{MAD}^{T-G}_{threshold\ 3}$ (τ_4)	$k_1 * \mu + k_2 * \sigma$

Table 5.3: Different threshold models used in the change detection module.

sensitivity is desirable in change detection algorithms. On the other hand, a large fraction of false positives can cause unnecessary alarms which can lead to significant performance loss. Thus, a good change detection technique should have high sensitivity and a small fraction of false positives. False positives are computed as the fraction of frames where the change is detected, while there were no objects introduced in the scene.

The static threshold τ_1 and dynamic threshold $model\tau_4$ outperforms the other two dynamic models, yielding 0.86 true positive and 0.09 false alarm ratios. The threshold $model\tau_4$ shows slightly higher sensitivity than the threshold $model\tau_1$. It is evident that the threshold models τ_2 and τ_3 can not perform well with a a maximum sensitivity of only of 71.58% and 42.06%, respectively.

Results show that the disparity enhancement module achieves a maximum sensitivity of 86.43% for a selected false alarm ratio of 9.16% when the $size_{threshold}$ is set to allow for a 20% change in the size and the value of $level_{threshold}$ is set to

be 30 where normalized disparity values range from 0 to 255.

Presenting consecutive registered frames and fused pair of images, is direct way of showing the efficiency of an algorithm as a visual evaluation.

In Figure 5.3, we present change detection results in 8 consecutive frames from the stereo video pair $\mathbb{V}^{4r} - \mathbb{V}^{4s}$. We first warp segmented dominant planes in $\mathbb{V}^{4s}$ on top of the dominant planes in $\mathbb{V}^{4s}$. Then, the estimated change mask is superimposed on the fused images.

5.2.2 Results of Disparity Estimation using Temporal Consistency Constraint

In Section 3.2.6, we present a way of integrating the temporal consistency constraint among the consecutive stereo frames into an energy minimization framework. We perform three experiments to evaluate performance of our method for different test cases using the stereo videos $\mathbb{V}^{1r}$, $\mathbb{V}^{1s}$,$\mathbb{V}^{2r}$, $\mathbb{V}^{2s}$, $\mathbb{V}^{3r}$, $\mathbb{V}^{3s}$, $\mathbb{V}^{4r}$, and $\mathbb{V}^{4s}$ (Table 5.1).

In Section 3.2.6, we propose an approach to estimate the disparity value of a pixel in the previous frame using its location in the current frame in the sequence.

In this experiment, we use the stereo video pairs $\mathbb{V}^{1r} - \mathbb{V}^{1s}$, $\mathbb{V}^{2r} - \mathbb{V}^{2s}$, $\mathbb{V}^{3r} - \mathbb{V}^{3s}$, and $\mathbb{V}^{4r} - \mathbb{V}^{4s}$ to evaluate the performance of our method in a video change detection framework. In Section 3.2.5, we propose a post-processing disparity refinement strategy. In Section 5.2.1, we present how this approach improves the

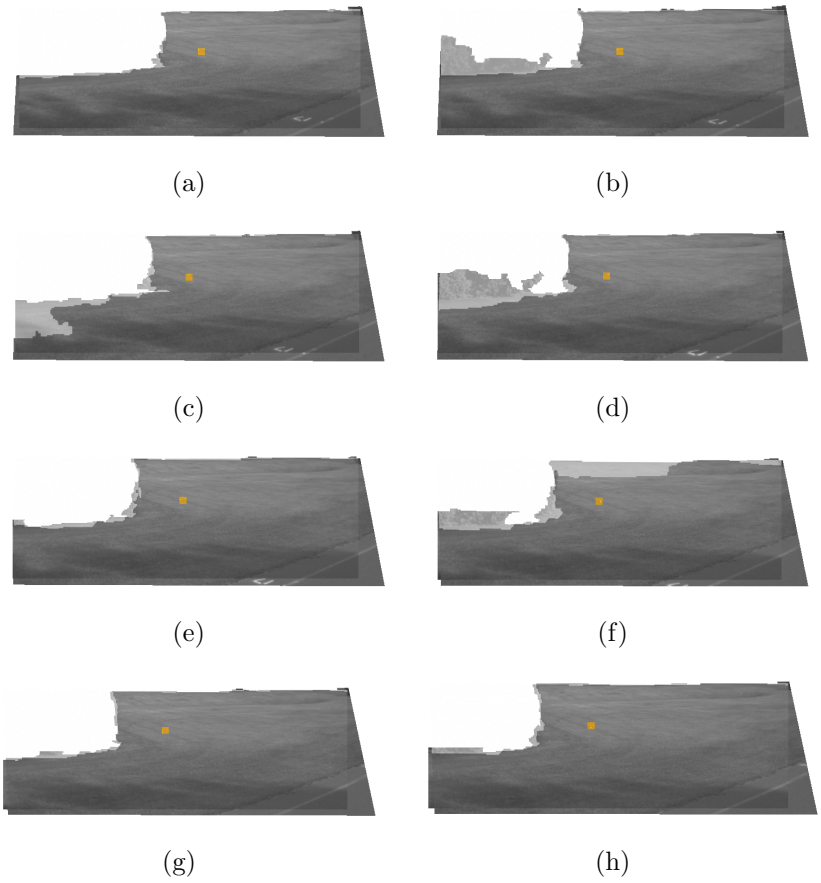


Figure 5.3: Illustrative examples of the change detection results in $\mathbb{V}^{4r} - \mathbb{V}^{4s}$.

change detection accuracy. We now repeat the same experiment after we integrate the temporal consistency constraint into the cost function. We report the change detection results in Table 5.4.

Especially the increase in accuracy for the objects O_1 and O_6 is very notable because most of the failures for these two objects were resulted from the inaccurate estimation of disparity maps in the post-processing strategy. We would

Video ID	Object ID	Viewable	Detected		Accuracy (%)	
			Post-processing	Energy Function	Post-processing	Energy Function
$\mathbb{V}^{1s}$	O_1	68	59	62	86.76	91.18
$\mathbb{V}^{1s}$	O_2	11	8	8	72.73	72.73
$\mathbb{V}^{2s}$	O_3	24	20	20	83.33	83.33
$\mathbb{V}^{3s}$	O_4	21	17	18	80.95	85.71
$\mathbb{V}^{4s}$	O_5	43	34	35	79.07	81.40
$\mathbb{V}^{4s}$	O_6	61	51	55	83.61	90.16
$\mathbb{V}^{4s}$	O_7	57	53	54	92.98	94.74
	Total	**285**	**242**	**252**	**84.91**	**88.42**

Table 5.4: Comparison of change detection results with two different disparity estimation approaches.

like to emphasize that in an energy minimization scheme, there may be several constraints or external factors affecting the final solution. The advantage of using the optimization approach in computer vision problems is that it provides a generic framework that enables us to incorporate different external factors in a balance. An apparent disadvantage of our approach is that very fast motions and the existence of very thin objects may cause errors.

5.3 Temporal Alignment

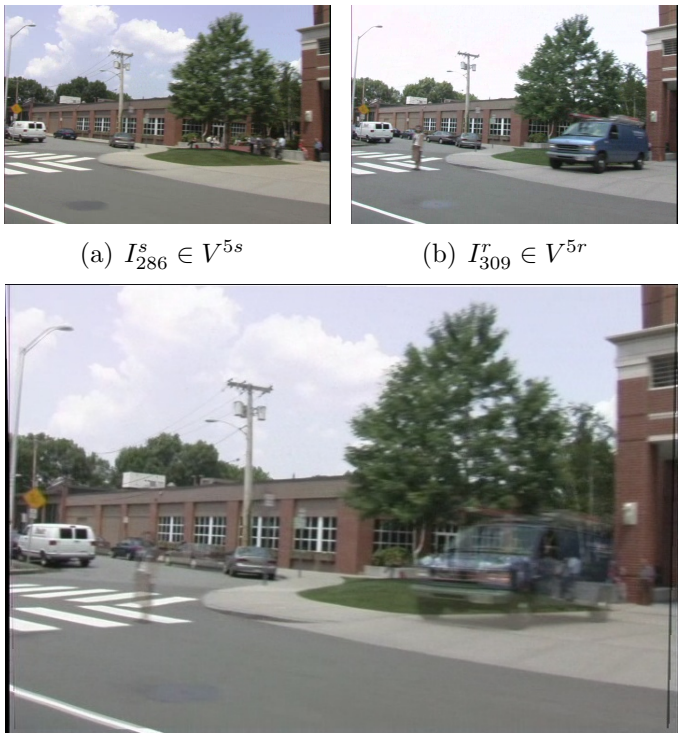
In Section 3.1, we propose a method for bringing two videos of the same scene recorded at different times to a temporal alignment. Namely, given a frame in one of the videos, the proposed method can find the corresponding frame, which has the most similar view, in the other video. Our experiments were conducted on one training (i.e., $\mathbb{V}^{4r} - \mathbb{V}^{4s}$) and six test reference-secondary video pairs (i.e.,

$\mathbb{V}^{1r} - \mathbb{V}^{1s}$, $\mathbb{V}^{2r} - \mathbb{V}^{2s}$, $\mathbb{V}^{3r} - \mathbb{V}^{3s}$, $V^{5r} - V^{5s}$, $V^{6r} - V^{6s}$, and $V^{7r} - V^{7s}$) recorded by mobile platforms in different dynamic outdoor environments (e.g., rural and urban) at daylight and night. In the videos, the mobile platform was driven in the same lane; however, the position of the vehicle in the lane changed continuously within a bound of ± 2 meters due to the driving conditions. Variations in the camera location introduced significant changes in the viewpoints of the same scene between reference-secondary videos (Figure 5.6(a)-(f)).

System parameters were set empirically on the training set extracted from the training set to obtain good synchronization results, and the same parameters were used for the rest of the videos. 45 pairs of corresponding frames from the training reference-secondary video pair $\mathbb{V}^{4r} - \mathbb{V}^{4s}$ are selected. We denote the training set as χ. χ consists of 45 pairs of corresponding frames: $\chi = \{(I_i^s, I_j^r)^t\}_{t=1}^{45}$.

First of all, the ground-truth data for all the test videos are generated to evaluate the performance of the video synchronization (i.e., temporal alignment) approach quantitatively. For some *frames of interest* in the secondary video, there is more than one *corresponding frame* in the reference video because some reference frames are quite similar. The other reason for multiple correspondence is the speed difference of the mobile platform between reference and secondary videos. For example, for the video pair $\mathbb{V}^{2r} - \mathbb{V}^{2s}$ (Table 5.5), there are 1227 (i.e., $1679 - 453 + 1$) *frame of interest* between I^s_{0453} and I^s_{1679}, whereas the total number of *corresponding frames* is 1385. We calculate temporal alignment error for a frame in two different ways: 1) single corresponding frame and 2) corresponding frame

interval. A corresponding frame interval is defined as the set of a few neighboring frames around the corresponding frame. We follow this approach because of the multiple correspondences mentioned above. In Figure 5.3, we present an example result of the proposed temporal alignment method.



(a) $I^s_{286} \in V^{5s}$ (b) $I^r_{309} \in V^{5r}$

(c) Superimposed Image

Figure 5.4: In (a) and (b), we present the result of the temporal alignment (i.e., corresponding frames) from the video pair $V^{5r} - V^{5s}$. In (c), we present spatially registered superimposed images, where the transparent regions refer to the changes between the corresponding frames.

Temporal alignment error is computed based on the difference between the index of corresponding frame determined by the the proposed algorithm and the

index of corresponding frame along with the frame interval in the ground-truth. Let us denote a corresponding frame interval of a frame of interest I_i^s as $[I_{j_L}^r, I_{j_U}^r]$, where L and U refer to lower and upper, respectively. Let I_j^r denote the output of the algorithm. The error function is defined as follows

$$erf(j) \triangleq \begin{cases} 0 & \text{if } (j - j_L) * (j_U - j) \geq 0 \\ |j - \frac{j_L + j_U}{2}|) & \text{otherwise.} \end{cases} \tag{5.1}$$

Based on the error function, let us define the synchronization accuracy as the ratio of the frame pairs with zero error to the total number of the frames. We first evaluate the synchronization accuracy by assigning only a single corresponding frame for every frame of interest. Then, we assign a frame interval to each frame of interest. In Table 5.5, we present the temporal alignment accuracy results for *Single* and *Interval* correspondence. The limitation of our approach arises from two main reasons. First of all, when the mobile platform is driven along a route in rural areas, the scene content among many frames is similar. In addition to scene content, the error increases when the speed of the mobile platform is very different between the videos. We compared our algorithm to the method proposed by Diego et al. [56] on their most challenging video pair *Highway2* (i.e., V^{6r}), where they observed the worst synchronization accuracy. We should point out that they tested this video with their algorithm using *only vision based features* and obtained 64% accuracy. The accuracy is defined as the percentage of the frames with zero error when a single corresponding frame in the ground-truth is allowed. Our algorithm outperforms their result and synchronizes the same video

with 79% accuracy. The accuracy increases to 87% when a frame interval is used instead of a single corresponding frame.

To evaluate the effect of combining two one-class learners, we conduct a different experiment using a standalone SVM based one-class learner, a standalone RNN based one-class learner, and the hybrid one-class learner. In a regular video synchronization problem, once the initial match is found, the space and time locality are taken into account, and the frame match is performed among the neighboring frames as explained in the algorithm. In this experiment setup (Figure 5.5), to measure the synchronization accuracy of different one-class learner models, we treat all the frames in the secondary videos as if they are the initial frames, and we never use the local search window.

(a) $I^s_{592} \in \mathbb{V}^{1s}$ (b) $I^r_{327} \in \mathbb{V}^{1r}$ (c) Trained SVM and RNN Outputs

Figure 5.5: In (a), we present a frame of interest in $\mathbb{V}^{1s}$. Our goal is to find its corresponding frame in $\mathbb{V}^{1r}$. For experimental purposes, we examine a subset of the frames in $\mathbb{V}^{1r}$ and compute the feature vector $\boldsymbol{F}_{(i,592)}$ for (I^r_i, I^s_{592}), $i = 1, \ldots, 786$. Output values of SVM and RNN are given in (c). Due to the nature of one-class learners, SVM maximizes and RNN minimizes the outputs for the most similar frame pairs. In (b), we present the estimated corresponding frame.

We observe that using the hybrid one-class learner decreases the average number of frames in the candidate pool from 28 frames to 22 frames. Furthermore, we observe that the hybrid one-class learner selects better candidate frames and results in more accurate synchronization. We observe that the stand-alone SVM classifier has average accuracy of 79%, stand-alone RNN has 82%, and hybrid one-class learner has 85% accuracy on the entire test videos.

Our experiments show that thresholded global DCT coefficients produces excellent results even when it is used for the synchronization of the frames, where there is a large variety of changes in the scene content of the corresponding frames, or there is a significant change in the view points. Examples of these cases are shown in Figure 5.6.

The proposed feature descriptor is invariant to changes in the illumination of the environment. Figure 5.6 (b) and (e), we present a pair of corresponding frames where the illumination conditions are significantly different. The feature descriptor is also robust to viewpoint changes (Figure 5.6(c) and (f)).

Video Pair	**Boundaries of the Candidate Frame Pool for the Initial Match**		**Mapping Set $\mathfrak{V}$**		**Synchronization Accuracy (%)**	
	P_L	P_U	Initial Match	Last Match	Single	Interval
$\mathbb{V}^{1r} - \mathbb{V}^{1s}$	132	149	(I^s_{0001}, I^r_{0143})	(I^s_{3256}, I^r_{2769})	81	88
$\mathbb{V}^{2r} - \mathbb{V}^{2s}$	1	18	(I^s_{0453}, I^r_{0001})	(I^s_{1679}, I^r_{1385})	84	92
$\mathbb{V}^{3r} - \mathbb{V}^{3s}$	442	471	(I^s_{0001}, I^r_{0456})	(I^s_{1678}, I^r_{1381})	91	95
$V^{5r} - V^{5s}$	1	29	(I^s_{0006}, I^r_{0005})	(I^s_{1438}, I^r_{1594})	89	94
$\mathbb{V}^{6r} - \mathbb{V}^{6s}$	558	585	(I^s_{0001}, I^r_{0576})	(I^s_{1432}, I^r_{2117})	79	87
$\mathbb{V}^{7r} - \mathbb{V}^{7s}$	63	82	(I^s_{0001}, I^r_{0071})	(I^s_{0448}, I^r_{0553})	86	93

Table 5.5: Total number of the frames (Table 5.1) in the secondary and reference videos of a pair are usually different because: 1) the speed of the mobile platform between the recordings may dynamically change, 2) the mobile platform does not follow the same trajectory, or 3) both. Candidate frame pool (Figure 3.1) contains the frames from the reference video that are considered as similar enough to the frame of interest in the secondary video by the hybrid one-class learner. It is also possible that some frames in the secondary video may not have corresponding frames in the reference video. For example, frames 1–452 in the secondary video $\mathbb{V}^{2s}$ do not have matches in the reference video $\mathbb{V}^{2r}$. The initial match for this pair is estimated as (I^s_{0453}, I^r_{0001}). When the one-class learner processes the frame I^s_{0453}, it labels 27 frames (i.e., $P_L = 1$ and $P_U = 27$) in the reference video as similar enough. These frames constitute the pool of candidate matches. Then, by minimizing the similarity error in the pool, I^r_{0001} is selected as the corresponding frame. $\mathfrak{V}$ denotes the mapping set which provides the pairs of corresponding frames between reference and secondary videos. The first of the last elements of the mapping sets are provided. *Single* synchronization accuracy refers to the case when only a single corresponding frame is assigned to the frame of interest in the ground-truth data. On the other hand, when multiple neighboring frames (i.e., corresponding frame interval) are assigned to the frame of interest, the case is called *interval*. In the interval case, the accuracy increases as expected.

(a) I^s_{976} (b) I^s_{1200} (c) I^s_{2200}

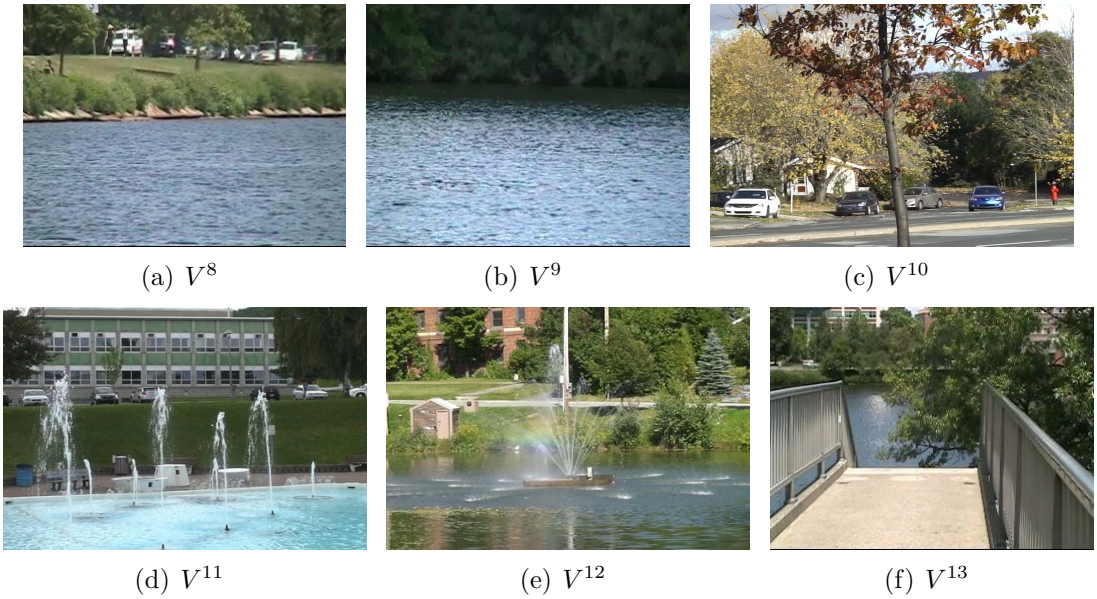
(d) I^r_{1516} (e) I^r_{749} (f) I^r_{1643}

Figure 5.6: Challenging corresponding frame pairs successfully matched. Frames (I^s_{976},I^r_{1516}) given (a) and (d) exhibit large region of change. In (b) and (e), the frame pair (I^s_{1200},I^r_{749}) presents a case that shadow regions within the images are significantly different due to weather conditions. In (c) and (f), the viewpoint between the corresponding frames (I^s_{2200},I^r_{1643}) changes significantly.

5.4 Change Detection in Dynamic Scenes

In Section 4.2, we propose an algorithm that is able to detect changes in outdoor scenes where the background has several altering elements that may cause false alarms. We perform experiments to evaluate performance of the change detection using the monocular videos V^8, V^9, V^{10}, V^{11}, V^{12}, and V^{13} (Table 5.1). Videos contain scenes with highly varying elements in the background such as shimmering water, fountains, and blowing trees (Figure 5.4).

The dataset includes a comprehensive set of annotated ground-truth change



(a) V^8 (b) V^9 (c) V^{10}

(d) V^{11} (e) V^{12} (f) V^{13}

Figure 5.7: Test videos with dynamic scenes.

areas to enable a precise quantitative evaluation.

5.4.1 Base Transform Estimation

The proposed method requires the estimation of suitable base transform for different types of ordinary change patterns. In Table 5.6, we present the ratio of regions modeled by different base transforms. The type of the estimated base transform can also be a good descriptor for the scene content. For example, DCT is known to have strong energy compaction property when applied to natural images [159]. In the videos V^{11} and V^{12}, there is a notable decrease in the overall use of DCT.

Video	Dynamic Background Description	Estimated Base Transform (%)		
		DCT	WHT	SL
V^8	shimmering water, blowing bushes	63.75	2.16	34.09
V^9	shimmering water, blowing trees	61.00	1.66	37.34
V^{10}	blowing trees	55.48	8.92	35.60
V^{11}	fountain, shimmering water	27.00	8.44	64.56
V^{12}	fountain, shimmering water, blowing bushes	36.00	14.81	49.19
V^{13}	shimmering water, blowing trees	52.58	5.16	42.26

Table 5.6: DCT, WHT, and SL are the base transforms: discrete cosine, Walsh-Hadamard, and Slant. We present the results of the transform estimation for the backgrounds in the six test videos. For example, in video *boats* 63.75% of the frame region is modeled by DCT. The type of the base transform used for the background actually gives hint about the scene content.

This is because of the fountains which jet water into the air, causing artificial ordinary change patterns. This result stresses the importance of employing different base transformations with complementary basis vectors.

5.4.2 Quantitative Evaluation of Salient Change Detection

A precise validation of a change detection method requires ground-truth at pixel resolution. Let p_{sc} denote a pixel in a region of salient change, and let p_{oc} denote a pixel in a region of ordinary change. If a change detection method labels p_{sc} as salient change, this case is called *true positive* (TP), and *false negative* (FN), otherwise. If a change detection method labels p_{oc} as ordinary change, this case is called *true negative* (TN), and *false positive* (FP), otherwise. For the entire test

	V^8	V^9	V^{10}	V^{11}	V^{12}	V^{13}	
Number of Test Frames	6,100	390	3,001	785	1,000	2,001	**Average**
Specificity (%)	99.96	99.74	99.96	99.49	99.95	99.99	99.83
Accuracy (%)	99.82	99.59	99.88	99.38	99.93	99.98	99.76

Table 5.7: The proposed method is able to identify ordinary changes with 99.83% specificity.

set, a joint probability value P_{XY} less than 0.33 is considered as an evidence that there is a salient change. Table 5.7 shows specificity and accuracy results at the pixel level.

ChangeDetection.net uses seven metrics to rank different change detection methods. Let us here present the two of the metrics, Recall (Re) and Precision (Pr), to compare our method to the other methods under the dynamic background category. The details of all the metrics and the ranking are presented in [69]. Re and Pr are given by: $Re = \frac{TP}{TP+FN}$ and $Pr = \frac{TP}{TP+FP}$. We present the comparison of our method to the three methods having the highest ranking for the dynamic background category on *ChangeDetection.net* in Table 5.8.

The major limitation of our method is that estimating base transforms requires a set of frames without salient changes. This a common issue for data-driven approaches. There may be scenarios where capturing training frames is not feasible. Another limitation arises from cube based computations, which may cause blocking artifacts. Compared to other methods demonstrated on the same test videos, our method shows significant improvement in change detection results.

Method	V^8		V^9		V^{11}		V^{12}		V^{13}		V^{10}		**Average**	
(*Ranking*)	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**
[75] (*4.71*)	0.63	0.92	0.95	0.79	**0.99**	0.68	0.80	0.50	**0.96**	0.86	**0.99**	0.92	0.89	0.78
[134] (*5.71*)	0.75	0.82	0.89	0.92	0.82	**0.90**	0.63	0.15	0.89	0.93	0.94	0.87	0.82	0.76
[96] (*6.14*)	0.53	**0.97**	0.79	**0.99**	0.91	0.89	**0.86**	0.40	0.86	**0.98**	0.70	0.92	0.77	**0.86**
Ours (*2.14*)	**0.78**	0.93	**0.96**	0.93	0.93	0.77	0.81	**0.58**	**0.96**	**0.98**	0.95	**0.97**	**0.90**	**0.86**

Table 5.8: In this table, we compare Recall (Re) and Precision (Pr) values of the top-three methods under the dynamic background category on ChangeDetection.net to ours. On the far left, we provide the rankings of each method. The overall ranking of a method across seven metrics is computed by taking the average of its ranking for each metric. The overall ranking of our method is 2.14, and the proposed method outperforms other 23 methods demonstrated for the dynamic background category on *ChangeDetection.net* (ranking results retrieved on June 2013).

Chapter 6

Conclusion

This dissertation has presented a set of methods for analyzing the regions of change between videos of the same environment recorded at different times: 1) temporal alignment of the unsynchronized videos, 2) estimation and refinement of the disparity maps using temporal consistencies, 3) segmentation of the dominant plane in the scene, 4) estimation of spatial transform for the dominant plane, and 5) detection of relevant changes in the presence of several altering background elements.

The ultimate goal of these methods is to be able to reliably detect all relevant changes between a recent video and a reference video of the same scene captured at different instances and/or from different viewing angles in an automated pipeline. There are a number of areas of applications, such as video surveillance, medical diagnosis, condition assessment, remote sensing, and driver assistance systems,

which may benefit from this ability. We here summarize our contributions to the field of spatiotemporal alignment by implementing this system and present the future work in this research area.

6.1 Summary of Key Contributions

Our primary contribution is a framework for spatially and temporally aligning videos. Spatial and temporal registration of the videos enables integration of information across multiple videos. The ability to align and integrate information across multiple videos both in time and in space can be applied to many real world scenarios. The following is a summary of our key contributions and novelties.

- A vision-based change detection framework.
- A method for utilizing sequential depth information for refining disparity maps.
- A method for integrating temporal disparity consistency constraint into energy minimization framework.
- A method for temporal alignment of videos which are recorded by a mobile platform following unknown trajectory at different instances.
- A method for spatial registration of scenes where complex image geometry and parallax are present.

- A method for detection relevant changes in the presence of several altering background elements.
- A method for efficacy comparison of unitary discrete transformations.

6.2 Future Work

In this section we mention potential additions or modifications to this dissertation which could provide topics of future research.

In Section 3.2.2, we discuss the methods for disparity estimating from single or a set of monocular images using motion and monocular cues. The problem of learning the depth from monocular images has not been extensively studied in the literature. Investigating new approaches for this problem would be highly beneficial.

In Section 3.2.6, we describe a way of integrating the temporal consistency constraint into an energy minimization framework. Nonetheless, there is a lack of the availability of a large ground-truth data for the disparity values of consecutive stereo frames. Considering the stereo videos available, it would be beneficial to prepare such ground-truth images for the research community working on the correspondence problem.

In Section 3.2.8, we assume that the change will occur within the dominant plane in the scene. Nonetheless, there may be cases where the regions of the

change are part of other planes in the scene. Therefore, introducing segmentation for all the planes in a frame and extending the spatial alignment method to the entire frame may improve the registration framework dramatically. Accordingly, the limitation because of the registration module would be overcome implicitly.

In Section 4.2.3, we propose a method for extracting spatiotemporal signatures of the local ordinary change patterns. Further improvements could be made to the set of base transforms in order to better capture different types of the change patterns. Similarly,the transform estimation scheme may be enhanced by improving the transform selection criterion. The proposed method may be extended to a background modeling algorithm by continuously updating the transformation matrix $\mathbf{T}$ with each new frames beyond the video subset V_o. With this extension, the proposed method become capable of incorporating new types of ordinary changes into the background of the scene.

Bibliography

[1] Combat Zones That See. Technical Report Solicitation Number SN03-13, The Defense Advanced Research Project Agency (DARPA), March 2003.

[2] T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Processing: Image Communication*, 7(2):147 – 160, 1995.

[3] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31(2):165–180, 1993.

[4] N. U. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, C-23(1):90–93, 1974.

[5] N. U. Ahmed and K. R. Rao. *Orthogonal Transforms for Digital Signal Processing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1975.

[6] N. U. Ahmed and K. R. Rao. Walsh-hadamard transform. In *Orthogonal Transforms for Digital Signal Processing*, pages 99–152. Springer Berlin Heidelberg, 1975.

[7] P. K. Ajmera, D. V. Jadhav, and R. S. Holambe. Text-independent speaker identification using radon and discrete cosine transforms based features from speech spectrogram. *Pattern Recognition*, 44(1011):2749–2759, 2011.

[8] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[9] T. Anselmo and D. Alfonso. Scene change adaptation for scalable video coding. In *17th European Signal Processing Conference (EUSIPCO 2009)*, pages 1819–1823, August 2009.

[10] K. Appiah, A. Hunter, J. Owens, P. Aiken, and K. Lewis. Autonomous real-time surveillance system with distributed ip cameras. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, pages 1–8, August 2009.

[11] F. Arman, A. Hsu, and M.-Y. Chiu. Image processing on compressed data for large video databases. In *Proceedings of the First ACM International Conference on Multimedia*, MULTIMEDIA '93, pages 267–272, New York, NY, USA, 1993. ACM.

[12] S. G. Armato, III, W. F. Sensakovic, S. J. Passen, R. Engelmann, and H. MacMahon. Temporal subtraction in chest radiography: Mutual information as a measure of image quality. *Medical Physics*, 36(12):5675–5682, 2009.

[13] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, and L. Scozzafava. Real-time people localization and tracking through fixed stereo vision. In *Lecture Notes on Artificial Intelligence LNAI*, pages 44–54, 2005.

[14] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[15] S. Berberoglu, A. Akin, P. M. Atkinson, and P. J. Curran. Utilizing image texture to detect land-cover change in mediterranean coastal wetlands. *International Journal of Remote Sensing*, 31(11), 2010.

[16] G. Bilgin, S. Erturk, and T. Yildirim. Unsupervised classification of hyperspectral-image data using fuzzy approaches that spatially exploit membership relations. *Geoscience and Remote Sensing Letters, IEEE*, 5(4):673–677, October 2008.

[17] R. A. Bindschadler, T. A. Scambos, H. Choi, and T. M. Haran. Ice sheet change detection by satellite image differencing. *Remote Sensing of Environment*, In Press, Corrected Proof:–, 2010.

[18] S. T. Birchfield, B. Natarajan, and C. Tomasi. Correspondence as energy-based segmentation. *Image and Vision Computing*, 25(8):1329–1340, 2007.

[19] S. T. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(4):401–406, 1998.

[20] M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Image Commun.*, 22(2):127–143, 2007.

[21] H. Boisgontier, V. Noblet, F. Heitz, L. Rumbach, and J.-P. Armspach. Generalized likelihood ratio tests for change detection in diffusion tensor images. In *ISBI'09: Proceedings of the Sixth IEEE International Conference on Symposium on Biomedical Imaging*, pages 811–814, Piscataway, NJ, USA, 2009. IEEE Press.

[22] L. Bombini, P. Cerri, P. Grisleri, S. Scaffardi, and P. Zani. An evaluation of monocular image stabilization algorithms for automotive applications. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 1562–1567, 2006.

[23] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 648–655, 1998.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[25] A. Broumandnia and M. Fathi. Application of pattern recognition for farsi license plate recognition. *ICGST International Journal on Graphics, Vision and Image Processing*, 05:25–31, Jan. 2005.

[26] L. Bruzzone and D. Prieto. Automatic analysis of the difference image for unsupervised change detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(3):1171–1182, May 2000.

[27] L. Bruzzone and D. Prieto. An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. *Image Processing, IEEE Transactions on*, 11(4):452–466, 2002.

[28] A. Buchanan. Novel view synthesis for change detection. In *6th Electro Magnetic Remote Sensing Defence Technology Centre Conference*, 2009.

[29] A. Bugeau and P. Prez. Detection and segmentation of moving objects in complex scenes. *Computer Vision and Image Understanding*, 113(4):459 – 476, 2009.

[30] S. Calderara, A. Prati, and R. Cucchiara. Video surveillance and multimedia forensics: an application to trajectory analysis. In *MiFor '09: Proceedings of the First ACM Workshop on Multimedia in Forensics*, pages 13–18, New York, NY, USA, 2009. ACM.

[31] J. B. Campbell and R. H. Wynne. *Introduction to Remote Sensing*. The Guilford Press, 2011.

[32] T. Camus. Real-time quantized optical flow. *Real-Time Imaging*, 3(2):71–86, 1997.

[33] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 8(6):679–698, 1986.

[34] Y. Caspi and M. Irani. Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002.

[35] Y. Caspi and M. Irani. Spatio-temporal alignment of sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(11):1409–1424, 2002.

[36] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006.

[37] P. Chakravarty, A. M. Zhang, R. Jarvis, and L. Kleeman. Anomaly detection and tracking for a patrolling robot. In *Proceedings of the 2007 Australasian Conference on Robotics and Automation (ACRA 2007)*, Dec. 2007.

[38] C.-S. Chan, C.-C. Chang, and Y.-C. Hu. Color image hiding scheme using image differencing. *Optical Engineering*, 44(1):017003, 2005.

[39] C.-C. Chang and C.-J. Lin. LIBSVM: A library for svms. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[40] J. Chen, X. Chen, X. Cui, and J. Chen. Change vector analysis in posterior probability space: A new method for land cover change detection. *Geoscience and Remote Sensing Letters, IEEE*, 8(2):317–321, 2011.

[41] Z. Chen, C. Wu, and H. T. Tsui. A new image rectification algorithm. *Pattern Recognition Letters*, 24(13):251–260, 2003.

[42] F.-C. Cheng, S.-C. Huang, and S.-J. Ruan. Advanced motion detection for intelligent video surveillance systems. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 983–984, New York, NY, USA, 2010. ACM.

[43] A. Cherian, V. Morellas, and N. Papanikolopoulos. Accurate 3d ground plane estimation from a single image. In *ICRA'09: Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, pages 519–525, Piscataway, NJ, USA, 2009. IEEE Press.

[44] K. Choi, S. Lee, and E. Jang. Zero coefficient-aware idct algorithm for fast video decoding. *Consumer Electronics, IEEE Transactions on*, 56(3):1822–1829, 2010.

[45] N. Chumerin and M. M. V. Hulle. Ground plane estimation based on dense stereo disparity. In *The Fifth International Conference on Neural Networks and Artificial Intelligence*, pages 209–213, 2008.

[46] C. Clifton. Change Detection in Overhead Imagery Using Neural Networks. *Applied Intelligence*, 18(2):215–234, March 2003.

[47] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Pittsburgh, PA, May 2000.

[48] P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: A review. *International Journal of Remote Sensing*, 25(9):1565–1596, 2004.

[49] S. Dabbaghchian, M. P. Ghaemmaghami, and A. Aghagolzadeh. Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition*, 43"(4):1431–1440, 2010.

[50] J. Daryaei. Digital Change Detection Using Multi-scale Wavelet Transformation and Neural Network, Master Thesis, The International Institute for Aerospace Survey and Earth Sciences, ITC, The Netherlands. 2003.

[51] S. Das and N. Ahuja. Performance analysis of stereo, vergence, and focus as depth cues for active vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(12):1213–1219, 1995.

[52] A. David and P. Jean. *Computer Vision: a Modern Approach*. Prentice-Hall, 2002.

[53] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: a unifying framework for depth from triangulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2):296–302, February 2005.

[54] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 557–564, 2000.

[55] L. Di Stefano, S. Mattoccia, and M. Mola. A change detection algorithm based on structure and colour. In *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 252 – 259, july 2003.

[56] F. Diego, D. Ponsa, J. Serrat, and A. Lopez. Video alignment for change detection. *Image Processing, IEEE Transactions on*, 20(7):1858–1869, July 2011.

[57] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72, 2005.

[58] M. El-Sharkawy, M. Aburdene, and W. Tsang. Parallel vector multidimensional slant, Haar, and Walsh-Hadamard transforms. *Multidimensional Systems and Signal Processing*, 3(4), October 1992.

[59] D. Farin and P. De. In *Misregistration Errors in Change Detection Algorithms and How to Avoid Them, In Proc. of IEEE International Conference on Image Processing*, volume 2, pages 438–441, 2005.

[60] M. Faundez-Zanuy, J. Roure, V. Espinosa-Dur, and J. A. Ortega. An efficient face verification method in a transformed domain. *Pattern Recognition Letters*, 28(7):854 – 858, 2007.

[61] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4:2379–2394, 1987.

[62] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[63] J. Fridrich. Key-dependent random image transforms and their applications in image watermarking. In *Proceedings of the 1999 International Conference on Imaging Science, Systems, and Technology, CISST*, volume 99, pages 237–243, 1999.

[64] E. Gabriel, V. Venkatesan, and S. Shah. Towards high performance cell segmentation in multispectral fine needle aspiration cytology of thyroid lesions. *Computer Methods and Programs in Biomedicine*, 98(3):231–240, 2010.

[65] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3):211–226, 1995.

[66] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

[67] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing using MATLAB*, volume 2. Gatesmark Publishing Tennessee, 2009.

[68] S. Gopal and C. Woodcock. Remote sensing of forest change using artificial neural networks. *Geoscience and Remote Sensing, IEEE Transactions on*, 34:398–404, 1996.

[69] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection.net a new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8, 2012.

[70] D. Gruyer, M. Mangeas, and R. Alix. Multi-sensors fusion approach for driver assistance systems. In *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pages 479–486, 2001.

[71] M. M. Grynbaum, W. K. Rashbaum, and A. Baker. Police Seek Man Taped Near Times Square Bomb Scene, The New York Times. http://www.nytimes.com/2010/05/03/nyregion/03timessquare.html. May 2010.

[72] W. Guo, L. Soibelman, and J. G. Jr. Automated defect detection for sewer pipeline inspection and condition assessment. *Automation in Construction*, 18(5):587 – 596, 2009.

[73] H. Haberdar and S. Shah. Disparity map refinement for video based scene change detection using a mobile stereo camera platform. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3890–3893, 2010.

[74] N. Haering, P. L. Venetianer, and A. Lipton. The evolution of video surveillance: an overview. *Mach. Vision Appl.*, 19(5-6):279–290, 2008.

[75] T. S. Haines and T. Xiang. Background subtraction with dirichlet processes. In *Computer Vision–ECCV 2012*, pages 99–113. Springer, 2012.

[76] A. Hampapur. Smart video surveillance for proactive security [in the spotlight]. *Signal Processing Magazine, IEEE*, 25(4):136 –134, july 2008.

[77] B. Han, W. Roberts, D. Wu, and J. Li. Motion-segmentation-based change detection. *Algorithms for Synthetic Aperture Radar Imagery*, 6568:207–218, 2007.

[78] M. J. Hannah. *Computer Matching of Areas in Stereo Images*. PhD thesis, Stanford, CA, USA, 1974.

[79] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[80] B. Hao, C. K.-S. Leung, S. Camorlinga, M. H. Reed, M. K. Bunge, J. Wrogemann, and R. J. Higgins. A computer-aided change detection system for paediatric acute intracranial haemorrhage. In *C3S2E '08: Proceedings of the 2008 C3S2E Conference*, pages 109–111, New York, NY, USA, 2008. ACM.

[81] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[82] G. D. Harville, G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 459–464, 1999.

[83] M. Harville, G. Gordon, and J. Woodfill. Foreground segmentation using adaptive mixture models in color and depth. In *in IEEE Workshop on Detection and Recognition of Events in Video*, pages 3–11, 2001.

[84] M. H. Hassan and S. L. Diab. Visual inspection of products with geometrical quality characteristics of known tolerances. *Ain Shams Engineering Journal*, 1(1):79 – 84, 2010.

[85] S. Hawkins, H. He, G. J. Williams, and R. A. Baxter. Outlier detection using RNNs. In *Data Warehousing and Knowledge Discovery*, pages 170–180, 2002.

[86] T. He, J. Pei, and Q. He. A target detection algorithm based on neighborhood structure measurement in video surveillance. volume 7495, pages 1–8. MIPPR 2009: Automatic Target Recognition and Image Analysis, 2009.

[87] A. T. S. Ho, X. Zhu, and J. Shen. Slant transform watermarking for digital images. *Visual Comm. and Image Processing*, 5150:1912–1920, 2003.

[88] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[89] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2137–2144, 2006.

[90] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[91] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *Computer Vision and Pattern Recognition, IEEE Computer Society*, volume 1, pages 74–81, 2004.

[92] Y. Z. Hsu, H. H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics, and Image Processing*, 26(1):73 – 106, 1984.

[93] H.-Y. Huang, C.-H. Yang, and W.-H. Hsu. A video watermarking technique based on pseudo-3-d dct and quantization index modulation. *Information Forensics and Security, IEEE Transactions on*, 5(4):625–637, 2010.

[94] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. The itk software guide: the insight segmentation and registration toolkit. *Kitware Inc*, 5, 2003.

[95] Y. Ishikawa, K. Uehira, and K. Yanaka. Optimization of size of pixel blocks for orthogonal transform in optical watermarking technique. *Display Technology, Journal of*, 8(9):505–510, 2012.

[96] M. M. Ismail, Mohamed Hamed and C. C. Chilufya. Object segmentation using full-spectrum matching of albedo derived from colour images, US Patent Pub. No. 2011/2374109, 12 2011.

[97] M. R. Jahanshahi and S. F. Masri. Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures. *Automation in Construction*, 22(1):567–576, 2012.

[98] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recogn.*, 38(12):2270–2285, December 2005.

[99] O. Javed and M. Shah. *Automated Multi-Camera Surveillance: Algorithms and Practice*. Springer Publishing Company, Incorporated, 2008.

[100] H.-Y. Jung, R. Prost, and T.-Y. Choi. A unified mathematical form of the walsh-hadamard transform for lossless image data compression. *Signal Processing*, 63(1):35 – 43, 1997.

[101] N. Kaempchen and K. Dietmayer. Fusion of laserscanner and video for advanced driver assistance systems. In *Proceedings of 11th World Congress on Intelligent Transportation Systems, Nagoya, Japan*. Citeseer, 2004.

[102] R. E. Kennedy, P. A. Townsend, J. E. Gross, W. B. Cohen, P. Bolstad, Y. Wang, and P. Adams. Remote sensing change detection tools for natural resource managers: Understanding concepts and tradeoffs in the design of landscape monitoring projects. *Remote Sensing of Environment*, 113(7):1382 – 1396, 2009. Monitoring Protected Areas.

[103] H. Kitajima. Energy packing efficiency of the hadamard transform. *Communications, IEEE Transactions on*, 24(11):1256–1258, 1976.

[104] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515, 2001.

[105] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 82–96, London, UK, 2002. Springer-Verlag.

[106] P. D. Kovesi. Matlab and octave functions for computer vision and image processing. http://www.csse.uwa.edu.au/∼pk, April 2010.

[107] D. Lal. A note on a form of tchebycheff's inequality for two or more variables. *Sankhyā: The Indian Journal of Statistics*, 15(3):317–320, 1955.

[108] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *Proceedings of Computer Vision and Pattern Recognition, IEEE Computer Society*, volume 2, pages 319–324, 2003.

[109] D.-S. Lee. Effective Gaussian Mixture Learning for Video Background Subtraction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):827–832, May 2005.

[110] M. Lee, R. K. Chan, and D. A. Adjeroh. Quantization of 3d-dct coefficients and scan order for video compression. *Journal of Visual Communication and Image Representation*, 8(4):405 – 422, 1997.

[111] C. Leung and B. C. Lovell. An energy minimization approach to stereo-temporal dense reconstruction. In *ICPR*, pages 72–75, 2004.

[112] L. Li and M. K. H. Leung. Integrating intensity and texture differences for robust change detection. *Image Processing, IEEE Transactions on*, 11(2):105 –112, feb 2002.

[113] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[114] W. H. Li, A. Tajbakhsh, C. Rathbone, and Y. Vashishtha. Image processing to automate condition assessment of overhead line components. In *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, pages 1–6, 2010.

[115] Y. Liu and P. Payeur. Vision-based detection of activity for traffic control. In *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, volume 2, pages 1347–1350, 2003.

[116] M. I. A. Lourakis, A. A. Argyros, and S. C. Orphanoudakis. Detecting planes in an uncalibrated image pair. In *Proceedings of British Machine Vision Conference*, pages 587–596, 2002.

[117] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91, Nov 2004.

[118] H. Lu, T. Zhang, and C. Yao. A novel automatic motion segmentation method based on optical flow. In *Multimedia Technology (ICMT), 2010 International Conference on*, pages 1–5, 2010.

[119] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.

[120] E. Maggio and A. Cavallaro. Learning scene context for multiple object tracking. *Trans. Img. Proc.*, 18(8):1873–1884, 2009.

[121] A. Makkeasorn, N.-B. Chang, and J. Li. Seasonal change detection of riparian zones with remote sensing images and genetic programming in a semi-arid watershed. *Journal of Environmental Management*, 90(2):1069 – 1080, 2009.

[122] R. Malladi and J. A. Sethian. Flows under min/max curvature flow and mean curvature: Applications in image processing. In *ECCV (1)*, pages 251–262, 1996.

[123] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[124] N. J. B. McFarlane and C. P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, May 1995.

[125] B. Meyer, T. Stich, M. Magnor, and M. Pollefeys. Subframe Temporal Alignment of Non-Stationary Cameras. In *Proc. British Machine Vision Conference BMVC '08*, 2008.

[126] R. Miezianko and D. Pokrajac. Texture dissimilarity measures for background change detection. In *ICIAR '08: Proceedings of the 5th International Conference on Image Analysis and Recognition*, pages 680–687, Berlin, Heidelberg, 2008. Springer-Verlag.

[127] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.

[128] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[129] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, November 2005.

[130] D. Min, S. Yea, and A. Vetro. Temporally consistent stereo matching using coherence function. In *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4, 2010.

[131] A. Mittal, A. Monnet, and N. Paragios. Scene modeling and change detection in dynamic scenes: A subspace approach. *Comput. Vis. Image Underst.*, 113(1):63–79, 2009.

[132] T. Moon. The expectation-maximization algorithm. *Signal Processing Magazine, IEEE*, 13(6):47 –60, nov 1996.

[133] H. P. Moravec. The Stanford Cart and the CMU Rover. *Autonomous Robot Vehicles*, 71(7):872–884, 1990.

[134] A. Morde, X. Ma, and S. Guler. Learning a background model for change detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 15–20. IEEE, 2012.

[135] F. N. Newell, C. Wallraven, and S. Huber. The role of characteristic motion in object categorization. *Journal of Vision*, 4(2), 2004.

[136] C.-W. Ngo, T.-C. Pong, and R. T. Chin. Exploiting image indexing techniques in dct domain. *Pattern Recognition*, 34(9):1841 – 1851, 2001.

[137] H. Ohga, H. Yabuuchi, E. Tsuboka, K. Mayumi, K. Adachi, and O. Nishijima. A Walsh-Hadamard transform lsi for speech recognition. *Consumer Electronics, IEEE Transactions on*, CE-28(3):263–270, 1982.

[138] N. Ohnishi and A. Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recogn. Lett.*, 27(9):1009–1021, 2006.

[139] T. Olson and F. Brill. Moving object detection and event recognition algorithms for smart cameras. In *DARPA Image Understanding Workshop*, pages 159–175, 1997.

[140] A. Osa, H. Yamashita, and H. Miike. Area-based estimation of stereo disparity using hierarchical windows. In *MVA*, pages 493–496, 2000.

[141] F. Padua, R. Carceroni, G. Santos, and K. Kutulakos. Linear sequence-to-sequence alignment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32:304–320, February 2010.

[142] J. Park, G. Lee, W. Cho, N. Toan, S. Kim, and S. Park. Moving object detection based on clausius entropy. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 517–521, 2010.

[143] D. Parks and S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, pages 192–199, 2008.

[144] W. Philips. An adaptive orthogonal transform for image data compression. In J. Vandewalle, R. Boite, M. Moonen, and A. Oosterlinck, editors, *Signal Processing*, pages 1255 – 1258. Elsevier, Oxford, 1992.

[145] W. Pratt, W.-H. Chen, and L. Welch. Slant transform image coding. *Communications, IEEE Transactions on*, 22(8):1075–1093, 1974.

[146] K. Primdahl, I. Katz, O. Feinstein, Y. L. Mok, H. Dahlkamp, D. Stavens, M. Montemerlo, and S. Thrun. Change detection from multiple camera images extended to non-stationary cameras. In *In Proceedings of Field and Service Robotics*, 2005.

[147] L. H. Quam. Readings in Computer Vision: Issues, Problems, Principles, and Paradigms. chapter Hierarchical warp stereo, pages 80–86. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[148] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *Image Processing, IEEE Transactions on*, 14(3):294–307, 2005.

[149] T. Randen and J. H. Husøy. Filtering for texture classification: A comparative study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(4):291–310, 1999.

[150] K. R. Rao, D. N. Kim, and J. J. Hwang. *Fast Fourier Transform: Algorithms and Applications*. Springer, 2010.

[151] R. Redondo, F. Šroubek, S. Fischer, and G. Cristóbal. Multifocus image fusion using the log-gabor transform and a multisize windows technique. *Inf. Fusion*, 10(2):163–171, 2009.

[152] P. Remagnino, S. A. Velastin, G. L. Foresti, and M. Trivedi. Novel concepts and challenges for the next generation of video surveillance systems. *Mach. Vision Appl.*, 18(3):135–137, 2007.

[153] E. Rignot and J. van Zyl. Change detection techniques for ers-1 sar data. *Geoscience and Remote Sensing, IEEE Transactions on*, 31(4):896 –906, jul 1993.

[154] H. Roemer, G. Kaiser, H. Sterr, and R. Ludwig. Using remote sensing to assess tsunami-induced impacts on coastal forest ecosystems at the andaman sea coast of thailand. *Natural Hazards and Earth System Science*, 10(4):729–745, 2010.

[155] S. Roy. Stereo without epipolar lines: A maximum-flow formulation. *International Journal of Computer Vision*, 34(2-3):147–161, 1999.

[156] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 492, Washington, DC, USA, 1998. IEEE Computer Society.

[157] A. Ruta, Y. Li, and X. Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43(1):416 – 430, 2010.

[158] M. Sabha and P. Dutre. Feature-based texture synthesis and editing using voronoi diagrams. *International Symposium on Voronoi Diagrams in Science and Engineering*, pages 165–170, 2009.

[159] D. Salomon. *Data Compression: The Complete Reference*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[160] P. Sand and S. J. Teller. Video matching. *ACM Trans. Graph.*, 23(3):592–599, August 2004.

[161] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168. MIT Press, 2005.

[162] A. Saxena, J. Schulte, and A. Y. Ng. Depth estimation using monocular and stereo cues. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2197–2203, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[163] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[164] S. Se and M. Brady. Ground plane estimation, error analysis and applications. *Robotics and Autonomous Systems*, 39(2):59–71, 2002.

[165] H. J. Seo and P. Milanfar. A non-parametric approach to automatic change detection in mri images of the brain. In *ISBI'09: Proceedings of the Sixth IEEE International Conference on Symposium on Biomedical Imaging*, pages 245–248, Piscataway, NJ, USA, 2009. IEEE Press.

[166] H.-Y. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[167] K. Skifstad and R. Jain. Illumination independent change detection for real world image sequences. *Comput. Vision Graph. Image Process.*, 46(3):387–399, 1989.

[168] P. Smits and A. Annoni. Toward specification-driven change detection. *Geoscience and Remote Sensing, IEEE Transactions on*, 38(3):1484 –1488, May 2000.

[169] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[170] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. Presented as a talk at the Stanford Artificial Project, January 1968.

[171] G. P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *In DARPA IU Workshop*, pages 521–527, 1998.

[172] J. V. Stone. Object recognition using spatiotemporal signatures. *Vision Research*, 38(7):947 – 951, 1998.

[173] W. Sun and S. P. Spackman. Multi-object segmentation by stereo mismatch. *Mach. Vision Appl.*, 7(2):1–14, 2009.

[174] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.

[175] G. Tanjung and T.-F. Lu. A study on indoor automatic change detection for a mobile-camera. In *Proceedings of the 2007 Australasian Conference on Robotics and Automation (ACRA 2007)*, Dec. 2007.

[176] H. Tao and H. S. Sawhney. Special issue on video surveillance research in industry and academia. *Mach. Vision Appl.*, 19(5-6):277–277, 2008.

[177] D. M. Tax and R. P. Duin. Combining one-class classifiers. In *Proc. Multiple Classifier Systems, 2001*, pages 299–308. Springer Verlag, 2001.

[178] J. Theiler, G. Cao, L. Bachega, and C. Bouman. Sparse matrix transform for hyperspectral image processing. *Selected Topics in Signal Processing, IEEE Journal of*, 5(3):424–437, 2011.

[179] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[180] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[181] H. Trinh, N. Haas, Y. Li, C. Otto, and S. Pankanti. Enhanced rail component detection and consolidation for rail track inspection. In *Applications of Computer Vision (WACV), 2012 IEEE Workshop on*, pages 289–295, 2012.

[182] G. Troglio, M. Alberti, J. A. Benediksson, G. Moser, S. B. Serpico, and E. Stefánsson. Unsupervised change-detection in retinal images by a multiple-classifier approach. In *MCS*, pages 94–103, 2010.

[183] V. Tucakov and D. Lowe. Temporally coherent stereo: improving performance through knowledge of motion. In *IEEE Robotics and Automation*, 1997. **3** (1999-2006).

[184] T. Tuytelaars and L. J. V. Gool. Synchronizing video sequences. In *CVPR*, pages 762–768, 2004.

[185] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *wisdomarchive.wisdom.weizmann.ac.il:81/archive/00000377/), Weizmann Institute of Science*, pages 538–550, 2006.

[186] E. L. B. van den. *Human-Centered Content-Based Image Retrieval*. PhD thesis, Radboud University, Nijmegen, September 2005.

[187] A. Vard, A. Monadjemi, K. Jamshidi, and N. Movahhedinia. Fast texture energy based image segmentation using directional walshhadamard transform and parametric active contour models. *Expert Systems with Applications*, 38(9):11722 – 11729, 2011.

[188] K. Veeraswamy, B. Chandra Mohan, and S. Srinivas Kumar. Hvs-based robust image watermarking scheme using slant transform. In *Second Int. Conf. on Digital Image Processing*, volume 7546, pages 1–6. SPIE, 2010.

[189] P. Viola and W. M. Wells, III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[190] Q. C. Vuong and M. J. Tarr. Structural similarity and spatiotemporal noise effects on learning dynamic novel objects. *Perception*, 35(4):497–510, 2006.

[191] H. Wang, D. Rosenfeld, M. Braun, and H. Yan. Compression and reconstruction of mri images using 2d dct. *Magnetic Resonance Imaging*, 10(3):427–432, 1992.

[192] D. Wedge, D. Huynh, and P. Kovesi. Using space-time interest points for video sequence synchronization. In *Proceedings of the IAPR Conference on Machine Vision Applications*, pages 190–194, May 2007.

[193] G.-Q. Wei, W. Brauer, and G. Hirzinger. Intensity- and gradient-based stereo matching using hierarchical gaussian basis functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1143–1160, 1998.

[194] X. Wen and X. Yang. Change detection from remote sensing imageries using spectral change vector analysis. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 2, pages 189–192, 2009.

[195] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965–980, October 1992.

[196] A. Whitehead, R. Laganiere, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In *Motion and Video Computing, 2005. WACV/MOTIONS '05 Volume 2. IEEE Workshop on*, volume 2, pages 132–137, January 2005.

[197] L. Wolf and A. Zomet. Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision*, 68(1):43–52, 2006.

[198] A. Y. Yang, S. Rao, A. Wagner, and Y. Ma. Segmentation of a piece-wise planar scene from perspective images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 154–161, 2005.

[199] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, decision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1973–1989, 2007.

[200] E. Yeniaras, Z. Deng, M. A. Syed, M. G. Davies, and N. V. Tsekos. A novel virtual reality environment for preoperative planning and simulation of image guided intracardiac surgeries with robotic manipulators. In *Proceedings of Medicine Meets Virtual Reality Conference*, volume 163, pages 716–722, 2011.

[201] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[202] P. Yip and K. Rao. Energy packing efficiency for the generalized discrete transforms. *Communications, IEEE Transactions on*, 26(8):1257–1262, 1978.

[203] K. Yokoi. Illumination-robust change detection using texture based features. In *Proceedings of the IAPR Conference on Machine Vision Applications*, pages 487–491, May 2007.

[204] T. S. Yoo. *Insight into Images: Principles and Practice for Segmentation, Registration, and Image Analysis*. AK Peters Ltd, 2004.

[205] H. H. Yu and W. Wolf. A hierarchical multiresolution video shot transition detection scheme. *Computer Vision and Image Understanding*, 75(1-2):196 – 213, 1999.

[206] A. Zergainoh, F.-Z. Nacer, and A. Merigot. Global discrete cosine transform for image compression. In *6th ISSPA Symposium*, volume 2, pages 545–548, 2001.

[207] X. Zhao, A. Ho, H. Treharne, V. Pankajakshan, C. Culnane, and W. Jiang. A novel semi-fragile image watermarking, authentication and self-restoration technique using the slant transform. In *Intelligent Information Hiding and Multimedia Signal Processing, 2007. IIHMSP 2007. Third International Conference on*, volume 1, pages 283–286, 2007.

[208] Y. Zhao and C. He. Improving change vector analysis in multi-temporal space to detect land cover changes by using cross-correlogram spectral matching algorithm. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 174–177, 2011.

[209] J. Zhu, L. Wang, J. Gao, and R. Yang. Spatial-temporal fusion for high accuracy depth maps using dynamic mrfs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):899–909, May 2010.

[210] Y. Zhuang, X. Hu, and J. Wang. The implement of an image stitching algorithm based on feature extraction. *Education Technology and Training, International Conference on*, pages 327–330, 2009.

[211] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, May 2006.