Hierarchical 2.5D Scene Alignment for Change Detection with Large Viewpoint Differences

Dennis W.J.M. van de Wouw, Gijs Dubbelman and Peter H.N. de With, Fellow, IEEE

Abstract—Change detection from mobile platforms is a relevant topic in the field of intelligent vehicles and has many applications, such as countering Improvised Explosive Devices (C-IED). Existing real-time C-IED systems are not robust against large viewpoint differences, which are unavoidable under realistic operating conditions in outdoor environments. To address this, we propose a new hierarchical 2.5D scene-alignment algorithm. First, the 3D ground surface of the historic scene is reconstructed by polygons, onto which historic image-based texture is projected. By estimating the 3D transformation between historic and live camera views, the historic scene can be rendered as if seen from the live camera viewpoint. To compensate for 3D alignment and reconstruction imperfections, local pixel-accurate registration refinement is performed in 2D. The proposed real-time 2.5D method thereby combines the accuracy of a 2D local image registration with the robustness of 3D scene alignment. It was found that the resulting change detection system detects small changes of only $18 \times 18 \times 9 \text{ cm}$ at distances of 60 meters under large trajectory deviations of up to 2.5 meters.

I. INTRODUCTION AND PROBLEM STATEMENT

Change detection aims at finding relevant changes in an environment between two different time moments, where the type of change is a priori unknown. This ability to find changes without prior knowledge of the scene is of great importance in many applications, such as video surveillance, road maintenance, but also countering Improvised Explosive Devices (IEDs). This paper focuses on the last application, where the objective is to identify suspicious changes in the environment that may indicate the presence of an IED, e.g. newly appeared or re-arranged objects along the road. In this research, we explore challenging scenarios in which changes as small as $18 \times 18 \times 9 \text{ cm}$ need to be detected up to a distance of 60 meters.

IEDs are responsible for the majority of fatalities amongst NATO troops in conflict areas such as in the Middle East, making early detection of such explosives a high priority. However, due to their ad-hoc nature, IEDs can have any shape, size and color. This makes it difficult to apply regular object detection techniques, since the appearance is a priori unknown. Instead, change detection can be employed to find suspicious changes by comparing the scene with respect to a previous visit, i.e. looking for differences in the environment, which may indicate the presence of an IED.

Manual large-scale inspection of the environment is in general not desirable and most of the time not even feasible, due to limited manpower. Change detection applied to video content allows these tasks to be automated using (live) video sequences, facilitating the observers by timely attracting their attention towards relevant modifications in the scene. The automation provides continuous support to a small group of operators to effectively monitor large areas with a constant accuracy and hence with higher safety.

In the context of countering IEDs, this automation consists of employing cameras on top of a vehicle to continuously monitor the environment. Change detection from such a moving vehicle is a challenging task, where images from different viewpoints and time instants, have to be accurately registered and compared in real-time. Moreover, for safety reasons, convoys try to drive in unpredictable trajectories, meaning they will drive on different parts of the road if possible. This results in large viewpoint differences between the live and historic video, as shown in Figure 1.

This change in viewpoints, is a major challenge for most mobile change detection systems, especially for monocular systems [1]. Systems that detect changes completely in 3D are better able to cope with large viewpoint differences, but typically lack the accuracy to find small changes in the scene, or are not capable to operate in real-time. Therefore, hybrid solutions have been proposed that exploit 3D geometry for robust scene alignment, but apply change detection to 2D images. Haberdar and Shah [2] apply ground-plane segmentation using texture and disparity estimates, e.g. depth information. They then align the ground planes of the historic
and live scene using a homography transform, after which change detection is performed in 2D. However, this method assumes a single linear plane, whereas the ground plane is often non-linear. For example, a slightly risen curb next to the road will result in significant misalignments when transformed with the homography transform based on the road surface, especially under large viewpoint differences.

Robust scene alignment is one of the most critical processing steps for change detection under realistic operating conditions. Related work on real-time change detection from moving vehicles with large viewpoint differences is limited, although 3D alignment itself has been broadly explored. Yang et al. propose a method to align scenes with large viewpoint variations using Structure From Motion [3]. After extracting only the dominant planes from the 3D point cloud, each dominant plane is transformed to a front-parallel view to achieve viewpoint invariance. Next, feature matching is performed to set up a point-to-point mapping between the two canonical views. Although this method works well in urban scenes that mainly consist of planar regions, the dominant planes will have insufficient accuracy to model natural scenes, such as forest environments.

In recent work [4], Lou and Gevers handle more complex environments by over-segmenting the scene and estimating a piecewise local geometric model for each segment, i.e. a local plane. Each local planar region is then separately aligned through an affine transformation. By re-segmenting and re-merging planar regions iteratively in an energy minimization framework, the method can align images even when they contain significant viewpoint changes. However, the piecewise local geometric model for each segment relies on accurate point-to-point correspondences within that segment. From our previous work, is has become clear that finding accurate feature matches on the ground plane is not always feasible [5]. The challenging road scenario in Figure 1, containing both texture-less asphalt and repeating textures, yields poor or invalid matches on the ground surface, even when using the A-SIFT features [6], as used in the work of Lou and Gevers. Moreover, the work of Lou and Gevers is computationally expensive and does not run in real-time. Therefore, the aforementioned approach is not directly suited for real-time change detection from a moving vehicle.

Instead, we focus on a real-time 2.5D alignment algorithm that is specifically designed for a mobile change detection system and can cope with large viewpoint differences, i.e. driving trajectories that are 2.5 meters apart (Figure 1). Moreover, the algorithm does not require a perfect or dense 3D reconstruction, as this is not feasible in real-time.

The remainder of this paper is organized as follows. Section II introduces the existing baseline change detection system along with our contributions. The proposed hierarchical alignment algorithm is discussed in detail in Section III. Experiments and results, using real-world data existing of 8 videos containing around 2,000 ground-truth annotations, are discussed in Section IV. Conclusions and possible directions for future work are provided in Section V.

II. BASELINE SYSTEM

The application context of the (baseline) change detection system is as follows. While driving, live video frames are acquired by a stereo camera and stored in a database alongside their GPS position, driving direction, extracted features and additional metadata. This initiates the capturing of a historical video. In a second drive, the historical and live video data can be synchronized in position using GPS, where the nearest historic frame is retrieved from the database. The historic and live frame are then compared to find changes in the environment.

For this comparison, feature matches are computed between the live and historic view, where depth information acquired from stereo disparity processing, is used to select features residing on the ground plane. Next, a global homography transform is used to align the historic ground plane to that of the live scene. Once the images are aligned, a difference image is generated by taking the maximum difference over the three color channels for each pixel. An adaptive thresholding algorithm by Su and Amer [7] is then applied to yield a binary change mask, which in turn is filtered by a Markov Random Fields algorithm [8], to distinguish relevant changes from noise. The resulting (binary) changes are tracked over time to improve the temporal consistency, after which post-processing techniques are employed to reduce the number of false detections. The latter exploit the 3D information acquired by stereo processing to reject unlikely changes, e.g. floating changes or changes without volume.

Extensive validation of the baseline system [5] has shown that it is not able to cope with large viewpoint changes in challenging urban scenarios, because the existing system has technical limitations:

1) Alignment is limited to a single (linear ground) plane, due to the usage of a global homography transform.
2) Point correspondences on the ground plane are not robust to large viewpoint differences, leading to an incorrect homography transformation.

Therefore, the objective of this work is to improve change detection by making the system robust to large viewpoint differences as follows.

1) Alignment is improved by using 3D methods that do not restrict the scene to a linear plane, e.g. no assumptions are made w.r.t. the shape of the road.
2) Point correspondences are no longer limited to the ground plane and 3D geometry is exploited to improve the robustness of the correspondences.
3) To compensate for imperfections in real-time 3D reconstruction and alignment, we add a local registration refinement in 2D.

The algorithmic details are discussed in Section III and Section IV shows the improvement w.r.t. the baseline system.

III. HIERARCHICAL 2.5D ALIGNMENT

This section describes the hierarchical 2.5D registration, where two images with clearly different viewpoints have to be aligned.
The resulting set of 3D point correspondences contains many outliers, i.e. incorrectly matched points. In fact, at large viewpoint differences, the set of outliers significantly outweighs the number of inliers, potentially leading to a poor alignment accuracy. Therefore, an additional feature selection technique is applied as introduced by Hirschmuller et al. [12]. This method finds the largest set of consistent point correspondences, based on the assumption that the distance between two 3D inliers remains the same, regardless of the viewpoint of the scene. Hence, if the (3D) distance between a pair of points in the historic view and a pair of points in the live view is different, at least one of those points must be incorrect. This approach allows to find a set of point correspondences that has significantly more inliers than outliers.

From this improved set of point correspondences, the rigid 3D transformation between the two scenes is estimated using RAndom SAmples Consensus (RANSAC), to remove the remaining outliers. The motion hypotheses employed by RANSAC are obtained by minimizing the following objective distance metric

$$
\arg\min_{R,t} \sum_{i=1}^{N} ||Rp_H^i + t - p_L^i||^2; \tag{1}
$$

where \(p_H^i\) and \(p_L^i\) represent the 3D coordinates of the point correspondences in the historic and live image, respectively, and \(R\) and \(t\) denote the rotation and translation defining the rigid 3D transformation.

B. 3D scene reconstruction

In order to simulate the historic scene under a different viewpoint, an accurate 3D model of the scene is required. In this work, the model is a (polygon) mesh of the ground surface, although it may be extended to a mesh of the entire scene.

The ground-surface model of the historic scene is constructed from a Digital Elevation Map (DEM) as follows:

1) Pre-filtering of the disparity map. The historic depth map (Figure 3) obtained from stereo disparity estimation is pre-filtered to remove all front-parallel objects. This is performed efficiently by removing stixels with constant disparity, i.e. neighboring pixels in the same column with constant disparity.

2) Historic 3D world point cloud. After pre-filtering, the majority of remaining pixels belong to the ground surface. These points are converted to a 3D world point cloud using the depth map in combination with calibrated cameras.

3) 2.5D Grid sampling. Instead of building a complete voxel representation of the 3D scene, the 3D point cloud is sampled into a 2.5D grid map to reduce the computational complexity, as discussed by Broggi et al.[13]. An \(m \times n\) grid is constructed lying on the \(x\)-plane, where each cell can be thought of as a cube along the \(y\)-axis. Each cell is constructed by initial and subsequent refined registration.

Figure 2 summarizes the proposed hierarchical registration algorithm and the relevant processing steps of the change detection system. The hierarchical registration consists of two phases. In the first phase, an initial registration is acquired by simulating the historic scene as if it was viewed by the live camera. The historic texture is projected onto a 3D mesh of the historic scene, acquired from stereo disparity estimation, which is then transformed according to the 3D transformation between the live and historic scene. The latter transformation is performed efficiently on a GPU. Due to small inaccuracies in estimating the 3D transformation or the 3D scene reconstruction, this registration may have local misalignments, e.g. a shift of several pixels (Figure 4-b). In the second phase, the registration is refined to a pixel-precise alignment using an optical flow approach, which is required for the 2D change detection system. Each block from Figure 2 is now discussed in detail.

A. Robust 3D transformation estimation

The task is to find the 3D transformation between the live and historic scene and then transform the historic scene as if seen by the live camera. Obtaining a proper set of feature correspondences is crucial, as found in our previous work, where a lack of point correspondences on the ground surface caused poor registration accuracy.

In this work, the FAST corner detector by Rotsten and Drummond [9] and [10] is used to find points of interest, while ORB features [11] are computed for feature matching. These are efficient alternatives to the well-known Harris corner detector and SIFT descriptors, which have shown similar performance in the proposed alignment algorithm. Bi-directional feature matching is applied, where a match is only accepted if the match holds in forward and backward direction. The resulting set of validated point correspondences is converted to a set of 3D correspondences, using the world coordinates obtained from stereo disparity estimation.

The ground-surface model of the historic scene is constructed from a Digital Elevation Map (DEM) as follows:

1) Pre-filtering of the disparity map. The historic depth map (Figure 3) obtained from stereo disparity estimation is pre-filtered to remove all front-parallel objects. This is performed efficiently by removing stixels with constant disparity, i.e. neighboring pixels in the same column with constant disparity.

2) Historic 3D world point cloud. After pre-filtering, the majority of remaining pixels belong to the ground surface. These points are converted to a 3D world point cloud using the depth map in combination with calibrated cameras.

3) 2.5D Grid sampling. Instead of building a complete voxel representation of the 3D scene, the 3D point cloud is sampled into a 2.5D grid map to reduce the computational complexity, as discussed by Broggi et al.[13]. An \(m \times n\) grid is constructed lying on the \(x\)-plane, where each cell can be thought of as a cube along the \(y\)-axis. Each cell is constructed by initial and subsequent refined registration.
point that belongs to its volume. Finally, for each
cell, a single 3D point is computed from all points
accumulated in that cell by:

\[(x_i, y_i, z_i) = \max(\min(y_c), \text{mean}(y_c) - \sigma_c), \tag{2}\]

where \(x_i\) and \(z_i\) represent the center coordinates of
cell \(i\), \(\min(y_c)\) represents the lowest \(y\) value in that
cell and \(\text{mean}(y_c)\) and \(\sigma_c\) represent the average \(y\) value
and the standard deviation of the cell \(c_i\), respectively.
If no 3D points lie inside a cell, the cell is marked
invalid and will be assigned a value during L1 spline
smoothing, which is discussed next.

4) Fast L1 spline smoothing. The resulting samples are
smoothed using fast L1 smoothing splines, as intro-
duced by Tepper and Sapiro\[14\]. This is an iterative
algorithm for computing L1 splines, based on split-
Bregman iteration\[15\], which in turn can be efficiently
solved by combining DCT and shrinkage operators. To
cope with invalid cells, i.e. cells that had no 3D world
points projected into them, a weighting vector is used
that assigns a zero weight to such cells. The resulting
smoothed elevation map yields elevation data for the
entire grid range.

5) Polygon mesh. The resulting digital elevation map is
converted to a 3D polygon mesh. An example is shown
in Figure 3.

---

**C. Live viewpoint simulation**

Once a polygon mesh representing the 3D ground-surface
shape of the historic scene is available, the texture of the
historic view is projected onto this 3D surface, after which
the 3D surface is transformed according to the estimated
3D transformation between the live and historic view. This
is implemented efficiently on a GPU through OpenGL. By
simulating the intrinsic camera parameters of the live camera,
the (registered) historic scene is back-projected to 2D, as if
seen by the live camera.

If significant viewpoint variations are absent, the resulting
registration is (often) pixel-accurate. However, in case of
large viewpoint variations, e.g. a lateral displacement of
4 m between the camera positions, the initial alignment
may be slightly off, as shown in Figure 4. This minor
misalignment depends both on the estimation accuracy of
the 3D transformation, as well as on the accuracy of the
scene reconstruction at that location.
D. Pixel-accurate registration refinement

The minor registration errors from the initial alignment have to be refined in the second phase of the hierarchical alignment. For each pixel in the live image, the best matching pixel in the (initially) registered historic image is located, using dense block matching according to a correlation measure (specified soon), using a template \( T \) from the live image (a block of pixels having the size of the support window) to be matched with the initially aligned historic image \( I \). Using these parameters, this correlation measure is specified by:

\[
R(x,y) = \frac{\sum_{x',y'} (T(x',y') - \mu_T)(I(x+x',y+y') - \mu_I)}{\sqrt{\sum_{x',y'} (T(x',y') - \mu_T)^2 \sum_{x',y'} (I(x+x',y+y') - \mu_I)^2}}.
\]

(3)

In Eq. (3), \( R(x,y) \) denotes the correlation measure of the match and \( \mu_T \) and \( \mu_I \) represent the mean value of the local support block in the live and historic image, respectively. Since the initial alignment is rather accurate, the local search area, i.e. the range of variation over \((x,y)\), is kept small (approximately 1.5\% of the image resolution).

This block matching results in a horizontal offset \( \Delta x_{i,j} \) and a vertical offset \( \Delta y_{i,j} \) for each pixel \( p_{i,j} \), similar to a dense optical flow field. A median filter is applied to this flow field, yielding a smoothed version \( \Delta \hat{x} \) and \( \Delta \hat{y} \). The refined historic image \( B \) is then constructed by considering the initially aligned historic image \( I \) and compensating for the shift at each position, thus \( B(i,j) = I(i - \Delta \hat{x}_{i,j} - \Delta \hat{y}_{i,j}) \).

A visual result is shown in the bottom sub-figure of Figure 4.

E. 2D Change detection

Change detection can now be performed on the live image and the registered historic image. A summary of the 2D change detection is shown in Section II and is not further discussed here.

IV. EXPERIMENTAL EVALUATION

For the experiments, the same eight videos are used that were also employed in our previous work [5], but now we specifically focus on large viewpoint differences. These videos are recorded under realistic conditions, by mounting the complete system on a car (Figure 5) and driving in an urban environment, which has proven to be challenging for our baseline change detection system. The videos are grouped in pairs, one captured prior to a change (the historic video) and one captured approximately an hour after changing the environment (the live video), giving a total of 4 datasets for our evaluation.

Experimental validation is two-fold. First, the hierarchical 2.5D scene alignment is validated separately by investigating the alignment error between the live image and the registered historic image after alignment. This experiment shows the validity of the proposed alignment method. Second, the proposed registration method is evaluated in the context of a change detection system, where it is shown that the proposed method results in an improved change detection performance. Both experiments are now discussed in detail.

A. Validation of the proposed registration

The proposed registration method discussed in Section III is validated by investigating the alignment error between the live image and the aligned historic image. For this purpose, approximately 500 characteristic points in the live scene were manually annotated, after which the exact same points were annotated in the registered historic images. The Euclidean distance between the annotated points in the live and registered frame are a direct measure of the accuracy of the 2.5D alignment method, where an error of 0 pixels represents a pixel-accurate registration.

Figure 6 shows the alignment error in pixels after registering the historic image to the live image for different lateral driving offsets, i.e. different viewpoint variations. Here, Figure 6(a) shows the alignment error caused by the 3D model and viewpoint simulation. Instead of simulating the live viewpoint, the 3D model of the historic scene is projected back to the historic viewpoint, i.e. an identity transform is applied instead of the 3D motion from Section III-A. The resulting average alignment error of 0 pixels shows that both the 3D model of the ground surface as well as the viewpoint simulation are very accurate. Next, Figure 6(b-e) show the alignment error after initial registration, i.e. after viewpoint simulation but prior to the local refinement. These figures demonstrate that the initial alignment is not yet pixel-accurate, although already quite accurate considering the high resolution of 1920 \( \times \) 1440 pixels and the large lateral displacement of the camera. Figure 6(f-i) show the alignment error after local refinement, as discussed in Section III-D. Alignment is now accurate up to a pixel, where it should be noted that the manual annotations are not necessarily pixel accurate.

To give a better comparison, Figure 7 shows similar plots for the baseline system, which applies a single homography transform to align the scenes. From this figure it is clear that the baseline alignment degrades significantly at a lateral driving offset of 1 meter, while the baseline system is not even able to align images with higher offsets.

It can be concluded that both the 3D model and viewpoint simulation yield pixel-accurate registration. The slight misalignments in the initial registration are caused by small inaccuracies in the estimation of the 3D transformation between the live and historic scene. It is also shown that the proposed
local registration refinement is able to correct for these slight inaccuracies, provided that the initial misalignment is not too severe.

B. Validation of the improved change detection system

Now that the proposed alignment was found accurate, it is also evaluated in the context of a change detection system. For this purpose, black, green, white and red wooden test objects of $18 \times 18 \times 9$ cm are placed in the environment in between the historic and live recordings, enforcing changes in the environment. Each dataset contains 4 of such manually placed test objects, including an accurate GPS position of each object. For a specific object, evaluation of the change detection starts at a distance of 60 meters from the manually placed test object, based on the GPS position of the car and the test object.

The time difference of an hour between historic and live acquisitions resulted in small changes (e.g. leaves moved by the wind), while maintaining a controlled test environment, e.g. no large undesired changes to be detected.

The frame rate of the videos is adapted to the vehicle speed using GPS, such that an individual stereo frame is captured every 1.0 m. After placing the test objects, the videos are manually annotated to obtain ground truth for evaluating the change detection system. This ground truth consists of a bounding box around each test object in every frame, resulting in approximately 2,000 annotated test objects.

C. Change detection evaluation criteria

The performance criteria of the system are based on the amount of True Positives (TPs), the False Positives (FPs) and the False Negatives (FNs). These numbers are calculated on the basis of the manually annotated ground truth. The TP count represents the number of correctly detected test objects. The FP count denotes the number of false alarms, i.e. detections that do not correspond to one of the placed test objects. The FN count stands for the number of placed test objects that are not detected.

We consider the Jaccard index as a scoring metric, i.e. the ratio of cut and union. This index implicitly takes into account the position, size and shape of the detected changes. A detected change by the system is considered a TP if the Jaccard index exceeds a predefined threshold: $J(A, B) = \frac{|A \cap B|}{|A \cup B|} > T$ and FP otherwise, where $B$ represents the rectangular bounding box containing the detected change and $A$ the annotated rectangular bounding box of a placed test object. The threshold is set to $T = 0.3$ for all our experiments.

From the TP, FP and FN counts, the recall and precision of the system are calculated by $TP/(TP + FN)$ and $TP/(TP + FP)$, respectively. The recall represents the fraction of correctly detected test objects from all annotated test objects, while the precision defines the fraction of correct detections from all detections, i.e. including false alarms.

D. Change detection results

In our experiments, the influence is measured of an increased lateral displacement between the historic and live video on the obtained recall and precision of the system. We have placed 4 test objects of various colors at the left sidewalk of an urban street (Figure 1). When capturing the live videos, the car driver takes trajectories with lateral offsets of 0, 1, 2.5 and 4 meters w.r.t. the trajectory of the historic video. This mimics increasingly challenging operational conditions, in which a vehicle cannot exactly drive the same trajectory multiple times.

Table I depicts the recall and precision values of the proposed change detection system with and without the hierarchical 2.5D registration. Whereas in the baseline system
a significant performance drop occurred for trajectories with a lateral offset, the proposed method is significantly more robust to lateral displacements. At an offset of 2.5 meters, the system is still capable of properly aligning live and historic images, albeit with some decrease in the recall of the system. Moreover, the majority of FN, i.e. missed objects, occur in close proximity to the car, i.e. within 20-m distance, which is less critical for a safe operation of the system. Each test object has already been detected over a distance of at least 30-40 meters before the detection finally fails. The baseline system was not even capable of aligning scenes with such viewpoint changes.

Figure 8 shows a typical graph denoting the distance at which a specific test object is detected. The object is already found at 56-m distance, but is sometimes missed.

<table>
<thead>
<tr>
<th>Lateral displacements:</th>
<th>0 m</th>
<th>1 m</th>
<th>2.5 m</th>
<th>4 m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous homography based system [5]</td>
<td>0.94</td>
<td>0.16</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Recall:</td>
<td>0.29</td>
<td>0.14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Precision:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recall:</td>
<td>0.97</td>
<td>0.89</td>
<td>0.71</td>
<td>0.19</td>
</tr>
<tr>
<td>Precision:</td>
<td>0.22</td>
<td>0.20</td>
<td>0.19</td>
<td>0.31</td>
</tr>
</tbody>
</table>

**TABLE I**

**RECALL AND PRECISION FOR FIXED LATERAL DISPLACEMENTS BETWEEN HISTORIC AND LIVE TRAJECTORIES.**

Table II shows the processing time of the current system and an expected processing time that can be achieved through system optimization. The change detection system has a pipelined execution. Therefore, the separate processing tasks of the proposed alignment method can be subdivided amongst different pipeline stages to optimize the throughput of the system.

In this work, the real-time constraint is determined by the maximum framerate of our custom stereo camera, which is 5 frames per second. Hence, the maximum delay of a single stage in the pipeline can be at most 200 ms. This can be achieved by algorithm optimization and by integrating the hierarchical 2.5D alignment into the pipelined design of the change detection system. An additional pipestage will be added for the stereo disparity processing, which by itself will take the full 200 ms that is allowed for a single stage. The expected 200 ms for stereo disparity processing is based on an actual measurement of a prototype under development, which is not further discussed here. By replacing the current stereo disparity processing by our prototype under development, the proposed hierarchical alignment is already near-real time.

**E. Discussion**

To decide whether a change is relevant for further consideration, automatic thresholding is used. Figure 9 shows an example, where this function clearly needs further refinement. While the test object is clearly distinguishable in the difference image, the thresholded image shows that the adaptive thresholding becomes too sensitive.

It should be noted that the current change detection approach focuses on relatively static scenes with only minor changes in weather conditions. We plan to extend our experimental datasets with more diverse recordings at different times of the day and various weather conditions. We expect that the proposed alignment is robust to such lighting conditions. The 3D transformation between the scenes relies on features, which are already robust to varying illumination conditions. The ground plane model, which relies solely on the historic disparity map, is also not affected by different lighting conditions between the live and historic drive. The weakest link will be the change detection itself, which may be influenced by the color differences introduced by different lighting conditions. However, this is outside of the scope of the current work and will be investigated in future work. Dynamic environments are also outside the scope of the current work, although operation in such environments is possible with some modifications. By extending the change detection with an additional filter for objects that move w.r.t. the scene, changes caused by such dynamic objects can be (partly) avoided.

**V. Conclusions**

We have developed a novel 2.5D hierarchical alignment algorithm specifically designed for detecting Improvised Explosive Devices (IEDs). The system combines the robustness of real-time 3D scene alignment with local registration refinement in 2D, to cope with large viewpoint differences.

The change detection system with the proposed alignment algorithm has been validated on 4 pairs of video sequences, containing over 2,000 manual annotations, which correspond to test objects of $18 \times 18 \times 9$ cm up to a distance of 60 meters. The results show that the alignment is key to the robustness of change detection with large viewpoint differences, where lateral displacements of up to 2.5 m are supported. This robustness is also clear from the obtained recall and precision values, where the recall only gradually decays with increased lateral displacement and a relatively stable precision.

Although the alignment is accurate, it was observed that the existing thresholding approach is not suited for the new registration. In future work, this will be addressed by researching descriptive histogram techniques to generate the
### Processing task (Fig. 2)

<table>
<thead>
<tr>
<th>Task</th>
<th>Current time (ms)</th>
<th>Future time (ms)</th>
<th>Expected time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo disparity processing</td>
<td>CPU: 7500</td>
<td>GPU: 200**</td>
<td></td>
</tr>
<tr>
<td>Find feature correspondences &amp; Estimate 3D transformation</td>
<td>CPU: 256*</td>
<td>GPU: &lt;100*</td>
<td></td>
</tr>
<tr>
<td>3D Scene reconstruction</td>
<td>CPU: 260*</td>
<td>GPU/GPU: &lt;100*</td>
<td></td>
</tr>
<tr>
<td>Simulate live viewpoint</td>
<td>GPU: 55</td>
<td>-</td>
<td>35</td>
</tr>
<tr>
<td>Registration refinement</td>
<td>CPU/GPU: 120</td>
<td>GPU: 75</td>
<td></td>
</tr>
</tbody>
</table>

#### Table II

**Timing of the proposed 2.5D hierarchical alignment, including required prior processing tasks. The third column shows the timing of the current implementation, while the fifth column shows the estimated execution time after (future) algorithm optimizations. All results are based on our C++ implementation with an image resolution of $1920 \times 1440$ pixels.**

(*) can be executed in parallel, (**) based on a prototype under development.

---

**Fig. 8.** Detection graph for a test object with a lateral displacement of the driving trajectory of 2.5 meters. The graph shows the object detection at a specific distance in meters (horizontal axis). A value of unity means proper detection and a zero value signals the full absence of detections at that specific distance. The top row of images represents the appearance of the test object used in this experiment, at multitudes of 10-m distance. The 2nd row of images shows the same objects in a zoomed view.

---

difference mask, which in turn will improve robustness to varying lighting conditions.

It can be concluded that the proposed algorithm for scene alignment significantly improves the robustness of the change detection system to viewpoint changes, making it better suited for operational usage by facilitating a broad margin of driving trajectories.

---

### REFERENCES


