Real-time Change Detection for Countering Improvised Explosive Devices

Dennis W.J.M. van de Wouw$^{a,b}$, Kris van Rens$^b$, Hugo van Lint$^b$, Egbert G.T. Jaspers$^b$ and Peter H.N. de With$^a$

$^a$Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, The Netherlands; $^b$ViNotion B.V., Horsten 1, 5612 AX, Eindhoven, The Netherlands;

1. ABSTRACT

We explore an automatic real-time change detection system to assist military personnel during transport and surveillance, by detection changes in the environment with respect to a previous operation. Such changes may indicate the presence of Improvised Explosive Devices (IEDs), which can then be bypassed. While driving, images of the scenes are acquired by the camera and stored with their GPS positions. At the same time, the best matching reference image (from a previous patrol) is retrieved and registered to the live image. Next a change mask is generated by differencing the reference and live image, followed by an adaptive thresholding technique. Post-processing steps such as Markov Random Fields, local texture comparisons and change tracking, further improve time- and space-consistency of changes and suppress noise. The resulting changes are visualized as an overlay on the live video content. The system has been extensively tested on 28 videos, containing over 10,000 manually annotated objects. The system is capable of detecting small test objects of 10 cm$^3$ at a range of 40 meters. Although the system shows an acceptable performance in multiple cases, the performance degrades under certain circumstances for which extensions are discussed.

Keywords: military driving assistance, counter-IED, image analysis, change detection, real-time

2. INTRODUCTION

Improvised Explosive Devices (IEDs) are one of the main causes of casualties amongst international-mission troops during transportation of people and materials in conflict zones. In order to reduce casualties, periodical surveillance of the high-risk areas is required in the areas where such transportation is planned. One effective method of surveillance is ground-based patrol. During such patrols, potential threats are localized by searching for suspicious patterns in the environment (e.g. suspiciously placed objects, markers, etc.) and comparing current and past environment situations. This comparison usually has to be made by military personnel, while the convoy is moving. This involves the need to remember the environment state, as observed during a previous patrol. Any suspicious change in the environment (new objects, moved structures, etc.) may form a potential threat, so that the soldier is effectively performing a manual intelligence task focusing on change detection to find possible IEDs. This is a very demanding task for a human, because his ability to concentrate on a task for a longer time interval in an unknown environment is limited. Furthermore, memorizing multiple details about the appearance of a specific environment is hard if the time and distance difference is significant. This paper proposes a real-time change detection system using automated image analysis. It can aid the personnel involved in detecting IEDs and thereby help prevent accidents during surveillance.

The development of such a system poses several technological challenges such as spatio-temporal localization, the proper visual interpretation of the scene and image comparisons, where this processing should be performed in real time. Furthermore, the ad-hoc nature of IEDs, which can have any shape or color, makes the detection of IEDs a difficult image processing task. Hence, appearance-based object recognition techniques are of limited use. Instead of searching for specific shapes, we assume that the placement of an IED results in visible changes to the environment (e.g. holes, digging tracks, wires, any newly placed or moved objects). This is a valid assumption as IEDs are often accompanied by markers. These markers are typically regular objects, such as stones or branches
near the road (Figure 1), which are used to time detonation of an IED when a vehicle passes it. Detection of either the IED or the marker is a sufficient warning for trained observing personnel that an IED may be present.

For the finding of the above markers and modifications, the proposed system uses change detection. The system captures and records videos during patrols (live video), which can be compared to previously recorded videos of the same scene (reference videos). Relevant changes between the live and reference scene are identified and shown to the user through a Graphical User Interface (GUI). Moreover, the live video is in turn stored to be used as a reference for future patrols.

The purpose of the change detection system is to facilitate the observers by attracting their attention towards possible markers and modifications in the scene. Thus, the key problem of the image processing system is to find notable differences in the image information at various locations and at different time moments, while these differences have basically an unknown nature. This requires that the signal processing concepts should have a generic nature and yet be powerful enough to detect such changes. Moreover, the image processing system should provide alerting signals in (near) real time, since the patrol may move at reasonable speed, e.g. when using vehicles. It will be shown that it is indeed possible to define a generic image processing algorithm that is capable of finding arbitrary object and or changes in the scene in real time.

The remainder of this paper is organized as follows. The system architecture is introduced in Section 3. Section 4 shows related work and our contributions. The algorithm is described in detail in Section 5 and real-time implementation aspects are presented in Section 6. Section 8 contains experiments and results and finally, conclusions and future work are given in Section 9. The latter also briefly discusses the drawbacks of a monocular change detection system.

3. SYSTEM ARCHITECTURE

The required change detection system, as depicted in Figure 2, evidently uses a camera for scene capturing. This camera features a high resolution, as this system should be looking far ahead of the patrol for timely warnings. The camera is typically mounted on a moving vehicle, which poses specific constraints. One of these constraints is severe motion, especially in open terrain. In order to avoid motion blur, a global shutter is applied.

The captured video-sequence frames are recorded in the processing system for later comparison. If the same area was already recorded earlier, the processing system can compare the two sequences in real time. To facilitate this comparison, an accurate positioning system is used, using both satellite reference signals (GPS) and inertial devices (IMU). This positioning system can be used in several ways. It can serve the personnel to localize the route accurately, but it can also enable quick search and retrieval of reference video sequences in the same area.

The whole system should be transportable and can be deployed on different vehicles. For this reason, the system features a mobile power generator and the processing system with its display are re-deployable in different vehicles. The display is a rugged LCD touch screen that enables the operators ease of use and provides new

Further author information: (Send correspondence to Dennis W.J.M. van de Wouw)
Dennis W.J.M. van de Wouw: E-mail: d.w.j.m.v.d.wouw@tue.nl
information continuously, also while driving. The resolution of the system is high enough to allow the operator to zoom in on far away objects of interest. The GUI also provides information to improve alignment of the driving trajectories.

4. RELATED WORK AND CONTRIBUTIONS

Change detection, the process of identifying differences in a scene by observing it at different times, has widespread interest in many fields, such as video surveillance, remote sensing, medical diagnosis, civil infrastructure and underwater sensing, as summarized in surveys by Radke et al.\textsuperscript{2} and Venkateswaran et al.\textsuperscript{3} Although the focus of these surveys lies on stationary cameras and remote sensing, the common identified processing steps are also valid for non-stationary cameras. The main difference is the additional temporal-alignment step required for non-stationary cameras, also referred to as image pairing or reference image selection. This is the process of retrieving a reference frame from a database or video that best fits the newly acquired live frame. Typically, this is based on the Euclidean distance between the GPS positions of both frames, taking into account the driving direction.\textsuperscript{4–7} In the absence of an accurate GPS device, an algorithmic alignment can be used, such as the Dynamic Time Warping (DTW) variant by Morton et al.,\textsuperscript{8} or the method described by Sand and Teller.\textsuperscript{9}

Although the other common processing steps such as image registration, intensity adjustment and change mask generation are similar to the stationary camera case, the implementations may be very different. For example, background modeling will not work for non-stationary cameras due to the moving background. Here, image registration is the process of mapping two or more images from possible different viewpoints to the same coordinate frame, which allows for direct comparison. If the scene is mostly rigid in nature and the camera motion is small, this registration can be performed using low-dimensional spatial transformations, such as a homography. Intensity adjustment is the pre-compensation for different illumination conditions. This ensures that the decision thresholds are independent of the intensity values. Alternatively, image or color-space transformations can be applied, such as the color conversion described by Elgammal et al. to reduce the effect of shadows.\textsuperscript{10} Differencing is the pixel-wise comparison of the reference and live frame. This processing is often implemented by a simple subtraction of the registered images, or by image ratioing,\textsuperscript{11} which uses the ratio of the pixel values instead of the difference. For multispectral images, change vector analysis (CVA) is often used, which represents pixels by feature vectors.\textsuperscript{12} Differencing is then performed on these feature vectors. For all three methods, a first version of the change mask is then generated by thresholding the resulting image. This change mask is refined through the decision rules, which generate the final change mask. While Radke et al.\textsuperscript{2} focus on probability-based solutions, such as significance and hypothesis tests for the decision rules, Venkateswaran et al.\textsuperscript{3} describe the decision rules as a segmentation problem, which is then solved by clustering the thresholded difference image.

As aforementioned, the general processing steps between stationary and non-stationary change detection are similar, while the implementations vary. Although implementations for the stationary case are widely available,
research into non-stationary change detection is still limited. Primdahl et al.\(^6\) propose a mobile change detection system that runs in a specific, well-defined corridor. They manually define a small search region of 6 × 6 meters with a fixed position w.r.t. the camera, which is not suited for counter-IED systems. No quantitative results are given.

Morales et al.\(^{45}\) use different temporal alignment rules. They do not necessarily chose the closest reference frame in the Euclidean sense, but select a reference frame with the smallest lateral displacement (similar orientation) that is still sufficiently close. Furthermore, they reduce the effects of external conditions (e.g. illumination, shadow, brightness) by choosing a reference frame recorded at a similar time of the day with similar brightness, instead of the most recent reference frame. This significantly reduces illumination and shadow effects. Any remaining changes due to shadows are identified with a thermal camera and removed from the change mask. Although searching for reference images with specific external conditions achieves good results, it is not recommended for counter-IED systems, where older frames may differ too much, overshadowing the typically small changes caused by an IED.

Morton et al.\(^8\) make the implicit assumption that changes, e.g. new objects, will result in new feature points. They propose to reject a large number of false positives by only considering alarms at feature point locations in the live frame that are far (in pixel-distance) from any feature point in the reference frame. The theory is that the vast majority of feature points occur in similar regions between reference and live frame. Although this allows for a fast implementation, experiments show that this assumption is not always true.

Our approach is similar to the aforementioned algorithms in that we use a number of the same common processing steps. However, we do not limit the search space, meaning changes can be found anywhere in the full-HD image. Furthermore, some objects may lack feature points on them, resulting in missed objects. This is more likely to happen with smaller objects at farther distance. In general, there may always be a mismatch between the object and the generic feature detection techniques deployed in a system, leading to at least a lower detection rate. For this reason, we only use features for image registration purposes. With respect to the real-time requirement, we propose a pipelined implementation which allows for real-time processing on a PC with off-the-shelf components, where we use GPUs to effectively process the large amounts of data involved in the change detection process. We conclude our research with a large-scale validation, involving more than 60 videos and 30,000 manually annotated objects of interest.

5. ALGORITHMIC DESCRIPTION

During patrols, live frames are acquired by the camera and stored in a database alongside their GPS position, driving direction, extracted features and additional metadata. Based on the GPS position, the live frame is compared in Euclidean sense to the nearest reference frame from the database. For this comparison, the reference frame is first registered to the live frame, using the extracted features. Next, a difference image is generated and adaptive thresholding is applied to generate the change mask. Post-processing is applied to remove inconsistencies and outliers from the change mask.

Figure 3 shows the change detection system as a chain of image processing functions. Here, some of the common processing steps are split up to give a better insight into the operation of the entire change detection chain. The resulting algorithmic blocks are explained in more detail in the sequel of this section.

5.1 Video and Position Capturing & Feature Extraction

Frames are captured with a state-of-the-art 20-Mpixels camera, which provides images directly in JPEG-2000 format. The frames are decoded at a resolution of 1920 × 1440 pixels, to allow for real-time processing\(^*\). Each frame is geo-tagged by an accurate GPS-INS system (SPAN-CPT) and additional information such as the standard deviation of the longitude and latitude, driving orientation, time stamp and extracted features, are stored in a database.

The extracted features are SURF features, which are robust to (small) viewpoint changes and can be efficiently computed. Here, the SURF feature extraction is slightly modified to prevent too small features to be found.

\(^*\)In future versions of the system, the resolution will be increased up to the maximum resolution of 5120 × 3840.
Experiments have shown these small features are tenuous and tend to change over time (e.g., small leaves that move in the wind or wind that changes the finest detail of a sand road). In practice, this allows to start feature extraction at a higher scale, resulting in a significant speedup. This reduces the feature extraction time to an average of 37 ms.

5.2 Temporal alignment

In the database, the GPS position is used for fast indexing of the video frames. As a result, reference frames from a specific position can be retrieved quickly, based on GPS position and driving direction. This results in a reference frame with the most similar viewpoint, as described in Algorithm 1.

**Algorithm 1** Temporal alignment

**Require:** GPS position (WGS84) and driving direction for live frame

Search all frames within $x$ meter radius from the current location;

if High GPS accuracy then

Retrieve closest reference frame (Euclidean distance) with similar driving direction;

else if Low GPS accuracy then

Match features of all reference frames within $x$ meter radius to those of live frame;

Retrieve the frame that has most feature matches with the live frame;

end if

return The reference frame that best resembles the live frame w.r.t. recording position and image content;

This algorithm is implemented using a PostgreSQL database with POSTGIS extension. This ensures reliable and very fast retrieval of images. A typical retrieval operation takes in the order of 1 millisecond (without image-data access) under normal conditions.

5.3 Image registration

The matched features between the reference and live frame are the basis for image registration. First, a bi-directional full-search matcher is applied to match the SURF features between the reference and live frame. These matches are refined by checking for scale and spatial consistency. The latter is done by rejecting matches whose translation, i.e., the differences in $x$- and $y$-positions between matched features, fall outside the standard deviation of the typical translation for that frame.

Once proper feature matches are available, a homography transform is estimated for the 3D ground plane of the scene. This global homography transform is then used to warp the reference frame to the live frame, involving also the parts that are not on the ground plane. Due to the usage of a single global homography, local warping errors may occur for objects that are not on the ground plane. By orienting the camera slightly downwards, the ground plane becomes the dominant plane in the scene and warping errors are only moderate.
5.4 Change mask generation

After image registration, the difference between the reference and live image can be computed at pixel level. Instead of applying simple differencing, we apply differencing per color channel (RGB) and take the maximum differences of these channels. Experiments have shown this yields additional contrast for camouflaged object detections. The resulting difference image is thresholded by an adaptive thresholding technique from Su and Amer, resulting in a raw change mask.

The attentive reader may notice that the intensity adjustment mentioned in Section 4 is not explicitly addressed here. In fact, the special form of differencing can be constructed with the individual color components. This can be exploited as an implicit color-space transformation. Furthermore, the adaptive thresholding is not affected by global illumination changes, making the intensity normalization redundant.

5.5 Post-processing

The changes resulting from the aforementioned steps are still noisy and inconsistent in space and time. To distinguish relevant, consistent changes from noise, a Markov Random Field (MRF) method, based on energy minimization is applied. After MRF processing, false detections are reduced by investigating their texture in terms of intensity values. This is achieved by employing template matching in the form of normalized cross-correlation. Thanks to the normalization, the chosen similarity measure is robust against homogeneous changes in illumination. In general, true changes do not match with the reference area and present low correlation values, while false changes show high correlation values.

We have found a better model for threshold calculation, by in-depth analysis of the population of cross-correlation values. When the cross-correlation values (one for each change) are sorted to their magnitude, the hull of the sorted amplitudes can be plotted as a curve. This curve shows a linearly decreasing plateau of high cross-correlation values, which exponentially drops at the end for the lowest cross-correlation values (see Figure 4). Close inspection of the changes and their corresponding cross-correlation values has revealed that the linearly decreasing plateau of high cross-correlation values originates from false detections. The cross-correlation values in the exponentially dropping part of the curve have been found to correspond to the true changes. These correspondences have been observed in practically all situations with a presence of true and false positives.

![Figure 4. Curve of ordered cross-correlation values for all detected changes (red bars). A linear regression is calculated using half of the values (starting from the highest values), represented by the blue line. The determined threshold is represented by the green line. For this particular example, 53 changes are detected, of which 40 will be rejected using the template matching with the linear regression model.](image)

Therefore, all changes with cross-correlation above a certain threshold can be rejected. This threshold is computed as follows. Let \( C = c_1, \ldots, c_n \) denote the set of cross-correlations corresponding to possible changes. Define the set of sorted cross-correlations \( B = (C, \geq) \) and let \( X = x_1, \ldots, x_n \) denote the index locations of the original changes in the ordered set \( B \). We solve the linear regression problem for the first half of the ordered
cross-correlation values, where $\Delta_{max}$ denotes the maximum difference between the actual value of any sample of the set and the linear regression curve at that position. This criterion is only applied to the first half of the sorted set, from which the linear regression is computed. More formally, this is specified by

$$\min_{\alpha, \beta} \sum_{i=1}^{n/2} (b_i - \alpha - \beta x_i)^2,$$

$$\Delta_{max} = \max_{i=1:n/2} (b_i - (\hat{\alpha} + \hat{\beta} x_i)),$$

where $\hat{\alpha}, \hat{\beta}$ denote optimal regression parameters. Here, $b_i$ refers to the $i$-th sorted cross-correlation and $x_i$ its index location. Next, all changes with cross-correlations above the threshold $\tau = 2\Delta_{max}$ are rejected, significantly reducing the number of false detections.

Finally, the temporal consistency of the detected changes is evaluated using a tracking algorithm based on the pyramidal optical flow method by Bouguet.\textsuperscript{15}

### 6. REAL-TIME IMPLEMENTATION

The used real-time image-analysis computer platform features a hexa-core CPU and two GPUs (GTX580). This section describes the mapping of algorithmic blocks of Section 5 on the processor type and the execution times.

#### 6.1 Using GPUs

Change detection involves processing large amounts of image data. Since data dependencies are limited for this application, parallel processing on GPUs is exploited. Table 1 lists the functions that are implemented for the GPU and the realized speedup factor with respect to a CPU implementation. It is clear from Table 1 that using GPUs, or similar parallel computation hardware (e.g. FPGAs), is required for our real-time change detection system. For example, change blob filtering on the CPU requires 429 ms, while the GPU implementation only takes 65 ms.

<table>
<thead>
<tr>
<th>Function</th>
<th>$t_{frame, CPU}$ (ms)</th>
<th>$t_{frame, GPU}$ (ms)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image perspective warping</td>
<td>80</td>
<td>0.50</td>
<td>160</td>
</tr>
<tr>
<td>Adaptive thresholding</td>
<td>7.42</td>
<td>0.814</td>
<td>9.12</td>
</tr>
<tr>
<td>Markov Random Field filtering</td>
<td>284</td>
<td>11.1</td>
<td>25.6</td>
</tr>
<tr>
<td>Change blob filtering</td>
<td>429</td>
<td>64.6</td>
<td>6.64</td>
</tr>
</tbody>
</table>

Table 1. Processing time differences between GPU and CPU implementations, including CPU-GPU data transfers. The 1st column shows the CPU execution time, the 2nd states GPU time and the 3rd presents the achieved speedup factor.

#### 6.2 Pipelining

To fully exploit the 6 cores of the CPU and the 2 GPUs, the set of algorithms is split into four stages. These stages are executed in parallel on the CPU cores and the GPUs. Table 2 shows the distribution of the change detection algorithms over these pipeline stages. This distribution has been manually optimized, such that the load is balanced over the pipeline stages.

The maximum stage execution time is 169 ms, which implies an ideal throughput rate of 5.92 frames per second. The real throughput of the system is somewhat lower, because scheduling of the pipeline involves certain overhead. This results in an effective throughput rate of 4.9 frames/second, corresponding to a latency of 0.73 s.

### 7. USER INTERFACE

The Graphical User Interface (GUI) is displayed on a touch screen and interactively visualizes detection results in real time, as well as various other statistics, such as GPS accuracy and navigation instructions. Figure 5 shows the mounted touch screen and the GUI layout.
<table>
<thead>
<tr>
<th>Pipeline stage</th>
<th>Function</th>
<th>Device</th>
<th>$t_{avg}$ (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read current image</td>
<td>CPU</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>SURF feature extraction</td>
<td>CPU</td>
<td>37.1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>139</strong></td>
</tr>
<tr>
<td>2</td>
<td>Read reference image</td>
<td>CPU</td>
<td>96.6</td>
</tr>
<tr>
<td></td>
<td>Feature matching</td>
<td>CPU</td>
<td>8.12</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>105</strong></td>
</tr>
<tr>
<td>3</td>
<td>State processing</td>
<td>GPU1</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Homography estimation</td>
<td>CPU</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Image perspective warping</td>
<td>GPU1</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>Image differencing</td>
<td>CPU</td>
<td>40.7</td>
</tr>
<tr>
<td></td>
<td>Adaptive thresholding</td>
<td>GPU1</td>
<td>0.814</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>169</strong></td>
</tr>
<tr>
<td>4</td>
<td>Markov Random Field filtering</td>
<td>GPU2</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>Change blob extraction</td>
<td>CPU</td>
<td>21.1</td>
</tr>
<tr>
<td></td>
<td>Change blob filtering</td>
<td>GPU2</td>
<td>64.6</td>
</tr>
<tr>
<td></td>
<td>Change tracking filtering</td>
<td>CPU</td>
<td>15.4</td>
</tr>
<tr>
<td></td>
<td>Draw changes</td>
<td>CPU</td>
<td>17.9</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td><strong>130</strong></td>
</tr>
</tbody>
</table>

Table 2. Distribution of the algorithmic blocks of Section 5 over the pipeline stages and execution times. The state processing is still under optimization and not discussed here.

The live view of the touch-screen based GUI, features two key user interactions:

- **The user taps on a detected change**
  The detected change area in the current view and the corresponding area in the reference view will be displayed in the zoomed views at the bottom of the GUI (see Figure 5 items 3 and 4). The zoom factor is dependent on the size of the detected change.

- **The user taps on a general position**
  The current view and the reference view will show the region around the selected position at full-resolution (i.e. 20 MPixels at 1:1 pixel correspondence).

### 8. EXPERIMENTS AND RESULTS

#### 8.1 Conditions for experiments

The system has been extensively tested on real-world data, including official live tests at the October 2012 NATO SCI-256 trial. For the evaluation in this paper, 28 videos are recorded by mounting the system on a car (Figure 6) and driving along rural and sand road environments. These videos are grouped in pairs, one recorded before a change (the reference video) and one recorded after a change of the environment (the live video), giving a total of 14 datasets for our evaluation.

The manually placed test objects, enforcing changes in the environment, consist of black, green, white and red wooden objects of $10 \times 10 \times 10$ cm, which are manually positioned in the environment. The number of manual test objects ranges between 4 and 20 test objects per dataset. The recording of each video starts at a distance of 60 m from the manually placed test objects. This distance of 60 m is 15 meters more than the safe detection distance of 45 m, which facilitates a safe halt of the vehicle driving at 20 km/h and stopping 20 m before the real position of a detected change. The safe detection distance of 45 m takes into consideration the latency of the change detection system, the reaction time of the user and the braking distance of the vehicle.

The manually placed test objects, enforcing changes in the environment, consist of black, green, white and red wooden objects of $10 \times 10 \times 10$ cm, which are manually positioned in the environment. The number of manual test objects ranges between 4 and 20 test objects per dataset. The recording of each video starts at a distance of 60 m from the manually placed test objects. This distance of 60 m is 15 meters more than the safe detection distance of 45 m, which facilitates a safe halt of the vehicle driving at 20 km/h and stopping 20 m before the real position of a detected change. The safe detection distance of 45 m takes into consideration the latency of the change detection system, the reaction time of the user and the braking distance of the vehicle.

The frame rate of the videos is adapted to the vehicle speed using GPS, such that an individual frame is captured every 0.5 m. This gives a total of approximately 120 frames for each of the two videos in a single...
Figure 5. (a) Graphical user interface displayed on a touchscreen mounted in the vehicle. This interface is displayed in more detail in the right image. (b) The GUI shows the (1) live video content, (2) detected changes highlighted in red on the live view, (3) zoomed-in version of the current frame at the location of the selected detection, (4) zoomed-in version of the reference frame at the location of the selected detection, (5) GPS navigation map showing current trajectory (magenta line) and reference trajectory (cyan line), (6) driving-speed assistant, which indicates whether the driving speed allows for real-time processing of each captured frame, or if the driver should slow down, (7) lateral-displacement indicators, which indicate the current lateral distance (left or right) with respect to the reference trajectory, (8) GPS-quality indicator that highlights the accuracy by horizontal and vertical standard deviation in meters, (9) Status bar showing processing information.

Figure 6. The change detection system integrated in a vehicle with the camera and GPS receiver mounted on top of the car and the computer inside of the vehicle.

dataset. The videos that are recorded after placing the test objects in the environment, are manually annotated to obtain ground truth for evaluating our change detection system. This ground truth consists of a bounding box around each test object in every frame, resulting in approximately 10,000 annotated test objects.

8.2 Experiments

The performance criteria of the system are based on the True Positive (TP) count, the False Positive (FP) count and the False Negative (FN) count. These counts 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 represents the
number of false alarms, i.e. detections that do not correspond to one of the placed test objects. The FN count represents the number of placed test objects that are not detected.

A detected change by the system is considered a TP if the ratio \((B \cap A) / (B \cup A) > t\) and as a 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. This approach is specifically chosen, as it ensures that a detection is only considered a TP, when its position, size and shape agree with the annotation, see Figure 7. The threshold \(t\) is set to 0.5 for all our experiments.

![Figure 7](image)

**Figure 7.** Examples of the effect of the decision criterion \((B \cap A) / (B \cup A)\) on detections. The bounding box \(B\) denotes the region of the detected test object and \(A\) that of the ground-truth annotation. The resulting metric value is shown below each example. This metric ensures that too small or too large detections have significant lower metric values, hence are rejected by the 0.5 threshold.

From the TP, FP and FN counts, the recall and precision of the system are calculated as \(TP / (TP + FN)\) and \(TP / (TP + FP)\), respectively. The recall represents the ratio of correctly detected test objects out of all annotated test objects, while the precision defines the ratio of correct detections out of all detections, i.e. including false alarms. The goal of our experiments is to validate the system and measure the recall and precision of our change detection system under various conditions. For this, we have performed 5 unit tests, which are described below.

1. **Object contrast test**

   In this unit test, the effect of test object contrast on the system recall and precision is investigated. For this test, we use 4 times different combinations of 2 datasets. Each dataset pair features 4 test objects of a single specific color (i.e. black, green, white or red). To ensure that the contrast of each of the 4 test objects to the background is similar, panels with uniform color are placed in the environment (see Figure 8). This background panel placement is done prior to recording the reference video, such that the panels themselves are part of the environment and do not ease the change detection task. This approach with the panels is used for all unit tests, except for unit test 5.

   ![Figure 8](image)

   **Figure 8.** Examples of the (a) black, (b) white, (c) green and (d) red test objects in the sand road environment, captured in full HD resolution at approximately 40-m distance.

2. **Lateral displacement test**

   In this unit test, the effect on the recall and precision of lateral displacement of a test object w.r.t. the road, is investigated. This unit test uses 4 times different pairs of datasets. Each dataset pair has test objects of a single specific color (i.e. black, green, red and white) at distances of 0, 4, 8 and 12 meters perpendicular to the road. This is visualized in Figure 9.
3. **Sampling interval test**
   In this unit test, the system is evaluated with different temporal sampling (snapshot) intervals related to physical distances between two successive frames of 0.5, 1.0, 1.5 and 2.0 m. These temporal intervals are obtained by sub-sampling the frames of the videos in the database.

   Because the frame rate of the system is adapted to the speed of the vehicle, the maximum frame rate upper bounds the speed at which we can still sample images at 0.5 m resolution. The frame rate used for all our experiments is 6 fps. With a sampling interval of 0.5 m, this is related to a driving speed of approximately 10 km/h. The sampling intervals of 1, 1.5 and 2 m are related to simulated driving speeds of approximately 20, 30 and 40 km/h, respectively. Therefore, this unit test simulates what would happen if we would drive at these higher speeds, without taking into account other side effects of driving at those higher speeds than the sampling interval, such as vibrations.

4. **Speed test**
   This test evaluates the performance of the system for various actual driving speeds of 10, 20 and 30 km/h. This unit test therefore takes into consideration both the sampling interval, as well as vibrations and other effects of driving at higher speeds. For this unit test, we only use the high-contrast black test objects.

5. **Trajectory deviation test**
   In this test, the effect of a lateral displacement between the reference video and the live video, on the recall and precision of the system, is investigated. In this test, 20 test objects of various colors are placed on the right border of the asphalt road, without using the uniformly colored panels. When recording the live videos, the car driver takes trajectories with lateral offsets of 0, 1, 2, 3, 4 and 5 meters w.r.t. the trajectory of the reference video. This simulates a realistic operational situation, in which a vehicle cannot exactly drive the same trajectory multiple times.

8.3 **Results**
This section presents the result of each unit test described in Section 8.2, followed by a general discussion on the performance of the proposed change detection system.

8.3.1 **Object contrast test**
The effect of the contrast of test objects on the recall of the system, is visualized in Figure 10. The low-contrast (i.e. white) test objects are not detected stably over time, while the medium-contrast (i.e. red) test objects are stably detected, though at a shorter distance from the camera than high-contrast test objects (i.e. black and green).

   A closer investigation of the results shows that low-contrast test objects are correctly detected when the post-processing step, i.e. template filtering, is not applied. In Table 3, we therefore show the effect of all post-processing steps on the recall and precision of the system. This table is based on data up to the safe stopping distance of 45 m. It shows that post-processing significantly reduces the recall for low-contrast test objects, but improves the overall precision.
Note that the precision of our system in this unit test is approximately 0.25. Thus, we obtain on the average, 3 false alarms for every correct detection. This is as expected, because apart from the placed test objects, the environment also changes over time, due to vehicle tracks, etc. A relatively low precision is inevitable for change detection under realistic operational conditions.

![Detection score for different objects colors](image)

Figure 10. Recall for objects of different color versus the distance at which the objects are detected. The objects involved are shown in Figure 8.

<table>
<thead>
<tr>
<th>Detection scores per color:</th>
<th>black</th>
<th>green</th>
<th>red</th>
<th>white</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>without post-processing</td>
<td>0.96</td>
<td>0.93</td>
<td>0.83</td>
<td>0.86</td>
</tr>
<tr>
<td>with post-processing</td>
<td>0.95</td>
<td>0.92</td>
<td>0.69</td>
<td>0.07</td>
</tr>
<tr>
<td>Precision:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>without post-processing</td>
<td>0.06</td>
<td>0.11</td>
<td>0.19</td>
<td>0.04</td>
</tr>
<tr>
<td>with post-processing</td>
<td>0.22</td>
<td>0.28</td>
<td>0.24</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 3. Recall and precision over an interval of 0-45 m detection distance from the test object, with and without post-processing. Post-processing clearly affects the detection of low-contrast (i.e. white) test objects.

### 8.3.2 Lateral displacement test

Figure 11 shows the recall of the change detection system for specific distances to and lateral displacements of test objects with high contrast. It can be observed, that the lateral displacement of test objects has no significant effect on the distance at which the recall is maximum and stable. This distance ranges between 40 and 45 m.

In general, high-contrast test objects are stably detected until they are at the border of the field of view, e.g. see the small drops in the recall curves of the test objects at 4 and 8 m displacement at 14 and 25 m distance, respectively.

The small drops in recall are caused by small differences in viewpoint between the reference video and the live video, hence part of the border pixels cannot be registered and changes inside this region are then missed. This phenomenon is inevitable and of no concern as the test objects have already been detected stably over time prior to this phenomenon.

The limitations due to the field of view of the monocular camera, become more significant as test objects are placed farther from the road. For example, the test objects placed at 12 m from the road, shift out of the field of view already at 42 m distance. This is a hardware limitation and can certainly be improved by using enlarged field-of-view imaging.

### 8.3.3 Spatial sampling test

Table 4 shows the effect of larger spatial sampling intervals on the change detection system. This table is based on data up to the safe detection distance of 45 meters. Using an interval of 1.5 m, which is three times larger than the baseline scenario of 0.5 m, the recall only slightly decreases by 2%. A further increase in the sampling
interval to 2 m decreases the recall w.r.t. the baseline by 7%. This is caused by errors introduced in the image registration, when viewpoints are too far apart. Using a sampling interval of 2 m is therefore not recommended.

The precision improves w.r.t. the baseline when using larger sampling intervals. The optimal precision is obtained by using a sampling interval of 1.5 m. This improvement in precision is caused by the fact that the tracking filter is better capable of suppressing false alarms when using larger intervals.

This test suggests an optimal sampling interval of 1.5 m, which is related to a driving speed of 30 km/h. However, this test does not consider vibrations and other effects that occur when driving at higher speeds. These effects are considered in the next unit test.

### 8.3.4 Speed test

Table 5 shows the recall and precision of the change detection system at different actual driving speeds. It can be seen that the recall reduces when driving at higher speeds, but the precision is more or less stable. Note that these results are based on high-contrast black objects only. Therefore, the values in this table are higher than those in Table 5, where we show results when using all test object colors.

A closer inspection on the cause of the reduction in recall reveals that test objects are in fact detected stably over time. However, the distance at which they are first detected decreases with higher speeds. This may be related to an increase of camera vibration, as the vehicle moves faster. Evidently, such vibrations influence the quality of the acquisitions (e.g. adding more artifacts) and therefore, the performance of the system.

### 8.3.5 Trajectory deviation test

Table 6 shows the recall and precision of the change detection system, for different lateral displacements between the reference trajectory and the live trajectory. The recall significantly decreases with an increment in lateral displacement of the driving trajectory. This occurs because for larger lateral displacements, the overlap reduces
between the field of views of the reference and live videos. This reduction in overlap harms the registration process, as the number of tracked image features reduces and their distribution becomes less uniform in the images. A decrease in the accuracy of the registration directly influences the recall and precision. This effect becomes stronger with larger displacements.

<table>
<thead>
<tr>
<th>Detection scores for different lateral displacements:</th>
<th>0m</th>
<th>1m</th>
<th>2m</th>
<th>3m</th>
<th>4m</th>
<th>5m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall:</td>
<td>0.75</td>
<td>0.41</td>
<td>0.17</td>
<td>0.10</td>
<td>0.07</td>
<td>0.02</td>
</tr>
<tr>
<td>Precision:</td>
<td>0.45</td>
<td>0.18</td>
<td>0.07</td>
<td>0.05</td>
<td>0.04</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 6. Recall and precision for parallel reference and live trajectories with a fixed lateral displacement.

### 8.4 Discussion

Under ideal circumstances, e.g., high contrast, no lateral displacement and limited driving speed, the change detection system is able to stably detect small test objects of $10 \times 10 \times 10$ cm. High-contrast objects, i.e., black and green objects, are already stably detected at 40 m distance. Medium-contrast objects (red), are stably detected at 25 m distance, while low-contrast objects are detected below 20 m distance with varying performance. Hence, under non-ideal operational conditions, the performance of the change detection system reduces, but the amount of this reduction depends strongly on the actual circumstances and the sensitivity of each of those circumstances. Some of the encountered degradations are discussed below, including recommendations for improvements.

Our template filtering approach shows weaknesses in discriminating low-contrast objects from their background and shadows. Therefore, detections associated to such low-contrast objects are often rejected. This harmful side-effect of the template filtering approach can potentially be reduced, by using texture correlation instead of intensity correlation. Another possibility to improve the detection of low-contrast objects, is to make better use of the full capabilities of the camera. Currently we use a dynamic range of 8 bits, but the camera supports a dynamic range of up to 12 bits. The full dynamic range of the camera is therefore not yet exploited.

The change detection system also shows a sensitivity to trajectory displacements and viewpoint changes between reference and live recordings. The reduced overlap between the field of views of the reference and live videos, deteriorate the registration process. This effect becomes stronger with larger displacements and can only be solved by improved registration techniques, such as multi-planar warping using depth information. Depth information can be obtained by incorporating a stereo vision system or a (complementary) laser-based imaging system.

The driving speed of the vehicle affects the recall and precision more than expected, taking into account the sampling interval test. The distance at which an object is stably detected, decreases with higher vehicle speeds. We assume that this is caused by strong vibrations caused by the vehicle driving at higher speeds. Such vibrations can be reduced by mounting the camera system on a gyro-based actively stabilized platform.

### 9. CONCLUSION AND FUTURE WORK

We have developed a real-time change detection system, which uses automated image analysis. The system is intended to serve as an aid for the personnel involved in detecting IEDs and thereby prevent or reduce accidents during transport and surveillance operations in conflict areas. The development of such a system poses several technological challenges such as spatio-temporal localization, the proper visual interpretation of the scene and image comparisons, all of which should be performed in real time. Furthermore, the ad-hoc nature of IEDs, which can have any shape or color, makes the detection of IEDs a difficult image processing task.

The proposed system detects IEDs in the environment by using image-based change detection techniques on videos recorded at different time moments. Such changes can be an indication of the placement of an IED in the environment between these time moments. While driving, images of the scene are acquired by the camera and stored with their GPS positions. At the same time, the best matching reference image (from a previous time moment) is retrieved and registered to the live image. A change mask is generated by differencing the
reference and live image, followed by an adaptive thresholding technique. Post-processing steps such as Markov Random Fields, local intensity comparisons using template matching and change tracking, further improve time-and space-consistency of changes and suppress noise. The resulting changes are visualized as an overlay on the live video content.

The system has been extensively tested on more than 28 videos containing over 10,000 manually annotated test objects. Under ideal operating conditions (e.g. similar trajectory and weather conditions between reference and live recordings) the change detection system is able to find all high-contrast test objects of $10 \times 10 \times 10$ cm at a 40-m distance. The system has proven to be robust to higher frame-sampling intervals up to 1.5 m, without a loss of performance. However, tests show that the distance at which test objects are first detected, decreases at higher speeds. Objects of low contrast also present a challenge for the current change detection system. Furthermore, the system is found to be rather sensitive to trajectory displacements and viewpoint changes between reference and live recordings. Potential solutions for these limitations, such as a stereo-vision system, are discussed and will be addressed in future work.

On the basis of our research, we conclude that automated real-time image analysis algorithms, as used in our system, offer a viable solution to detect IEDs and prevent their harmful impact on transports.

REFERENCES