Real-time estimation of the 3D transformation between images with large viewpoint differences in cluttered environments

https://doi.org/10.2352/ISSN.2470-1173.2017.13.IPAS-209

Real-time Estimation of the 3D Transformation between Images with Large Viewpoint Differences in Cluttered Environments

Dennis W.J.M. van de Wouw [a,b] ***; Martin A.R. Pieck*** [a] ***; Gijs Dubbelman*** [a] ***and Peter H.N. de With*** [a]
[a] ***Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, The Netherlands;***
[b] ***ViNotion B.V., Daalakkersweg 2-58, 5641 JA, Eindhoven, The Netherlands;***

Abstract

This work focuses on estimating an accurate 3D transformation in real time, which is used to register images acquired from different viewpoints. The main challenges are significant image appearance differences, which originate from lateral displacements and parallax, inconsistencies in our 3D model and achieving real-time execution. To this end, we propose a feature-based method using a single synthesized view, which can cope with significant image appearance differences. The 3D transformation is estimated using an EPnP refinement to minimize the influence of inconsistencies in the 3D model. We demonstrate that the proposed method achieves over 95% transformation accuracy for lateral displacements up to 350 cm, while still achieving 85% accuracy at displacements of 530 cm. Additionally, with a running time of 100 milliseconds, we achieve real-time execution as a result of efficiency optimizations and GPU implementations of time-critical components.

Introduction

Image registration or image alignment, consists of placing two images depicting the same scene, in the same coordinate system, where one image is recorded from an unknown changed camera pose. This effectively means that the images are placed in the same coordinate system and thus nullify the camera-pose change, where at least one of the images is required to undergo a transformation. This transformation ensures that pixels representing the same world objects in both images, map to the same image coordinate after successful registration. The challenge is then defined as achieving a pixel-accurate alignment, regardless of the camera pose.

We study image registration in the context of a mobile application, where images are recorded from a moving vehicle and need to be aligned to images acquired during an earlier drive, where the latter will be referred to as historic images. The acquisition setup consists of a stereo camera pair, so that the 2D images are complemented with depth information. Both the images and depth map are acquired once every meter at a resolution of 1920×1440 pixels. A GPS receiver, IMU (Inertial Measurement Unit) and Real Time Kinematic (RTK) correction are employed for improved positioning accuracy. The resulting stereo images are geo-referenced with an accuracy up to tenths of centimeters.

Our mobile application presents additional registration challenges. First and foremost, the relative order of objects in the scene changes due to parallax effects, as shown in Figure 1. This effect occurs when images are captured from a moving vehicle with little restriction to trajectories, i.e. driving trajectories can be meters apart. This results in significant perspective differences and different relative positions of objects in the scene. Second, the time difference between the images in a cluttered environment often results in visual changes between the two scenes, e.g. a parked vehicle or strong shadows with a different orientation due to a different daytime. This becomes especially relevant in combination with different viewpoints, which limits the viewpoint overlap, meaning that a significant part of the scene may look different.

To overcome the first challenge and deal with the parallax effects, the 2.5D hierarchical alignment introduced in previous work [25] is adopted. To summarize this 2.5D method, a textured 3D model of the historic scene is obtained, which is transformed (in 3D) to the live camera pose. The transformed 3D model is projected back to a 2D image, resulting in an image of the historic scene, as if captured by the live camera. Since the transformation took place in 3D, the parallax effects are handled correctly. This approach is visualized in Figure 2 and explained in more detail in the Section '*Baseline system*'.

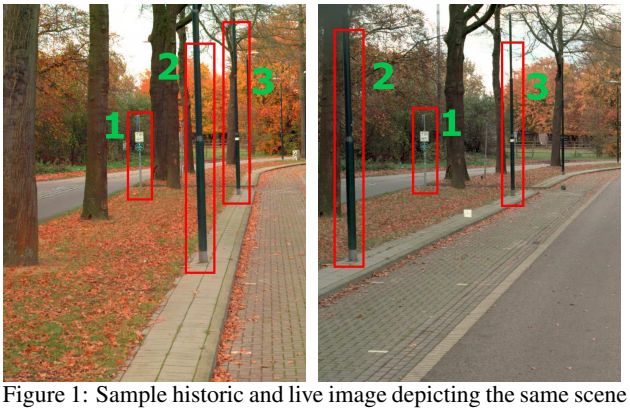


Figure 1: Sample historic and live image depicting the same scene from different viewpoints. Note the significant appearance difference and the strong parallax effect, which changes the relative position of objects from 1-2-3 to 2-1-3.

The second challenge, the cluttered environment and varying recording conditions, make it difficult to accurately estimate the 3D transformation between the live and historic camera pose. In fact, this is the bottleneck of the existing 2.5D system [25], which shows good performance for (lateral) displacements up to 2.5 meters, but the performance drops rapidly when displacements become larger. The objective of our method is therefore to estimate a robust and accurate 3D transformation between the live and historic camera pose under large viewpoint differences and dynamic changes in the scene. Here, our real-time requirement is satisfied at 6 fps, because that is the maximum capturing speed of our

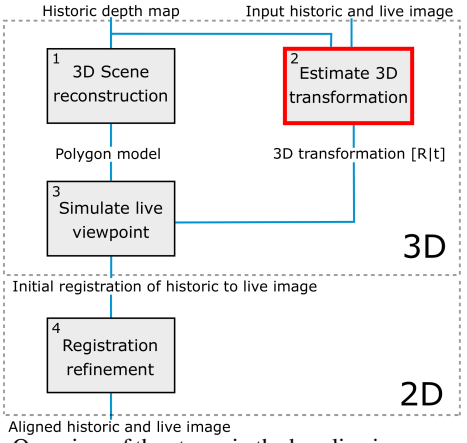


Figure 2: Overview of the stages in the baseline image registration method [25]. This work focuses on estimating the 3D transformation, the stage outlined in red.

custom-made stereo camera. The accuracy requirement of the final (2D) alignment is a maximum alignment error of a few pixels within images of HD resolution (1920×1440 pixels).

The main contributions of this paper are twofold. First, we propose a robust 3D transformation estimation method, that has a significantly higher accuracy than the baseline system and is on-par with computationally more expensive state-of-the-art methods. Second, we demonstrate real-time execution resulting from efficiency optimizations and a GPU implementation of the time-critical elements in our method.

The remainder of this paper is organized as follows. In the next section, an overview is given of related work in image registration. Then, in Section '*Baseline system*', the baseline image registration method as well as our acquisition setup is described. Section '*Approach*' details our proposed method, followed by a thorough validation of our method in Section '*Results*'. Finally, we discuss the limitations of the proposed work in Section '*Discussion & Recommendations*' and draw conclusions in Section '*Conclusions*'.

Related work

Image registration has been widely researched for a broad range of applications. In this section we briefly overview the two approaches most relevant to our context: feature-based and point-based registration.

Point-based registration

The point-based registration approaches aim at minimizing the distance between two 3D point clouds, derived from the depth map only. The most widely used algorithm is Iterative Closest Point (ICP) [2], which iteratively converges to an optimal solution. Although reliable for synthetic 3D models, ICP is neither robust to noise nor outliers [21], which are unavoidable when using passive stereo cameras to generate a depth map.

Various improvements to ICP have been proposed. Most notably, probabilistic variants [23][15] show increased robustness to noise and outliers, but remain prone to converging at local optima. Additionally, ICP-like algorithms often suffer from long execution times due to their iterative nature. Typically, downsampling is applied to improve the execution time. However, such downsampling increases the registration error beyond the required alignment accuracy. For these reasons, we consider point-based registration to be less suitable for our high-resolution mobile application and do not pursue this approach any further in this work.

Feature-based registration

Typically, feature-based registration approaches first compute features in both images. Second, these features are matched to generate point-to-point correspondences between the two images. Finally, a transformation is estimated from these correspondences. A distinction can be made between methods that use 2D (image) features and 3D (point-cloud) features.

Theoretically, 3D features are very powerful for generating strong correspondences at salient 3D structures [9][24], especially when color is incorporated [22]. However, such features typically use point orientations [26], which are derived from locally estimated surfaces in the point cloud. In the case of our mobile stereo setup, the point cloud is sparse and noisy over the viewing direction, which means that surface normals cannot be computed reliably (at the scale of interest). The resulting 3D features are therefore not found at consistent locations between scenes, which makes them unsuitable for estimating the 3D transformation in the considered mobile application.

Alternatively, 2D image features can be used, such as the well-known Scale Invariant Feature Points (SIFT) [12] and Local Binary Patterns (LBP)[16] descriptors. In this work, Binary Robust Independent Elementary Features (BRIEF) [4] descriptors are used together with the Accelerated Segment Test (FAST) [18][19], where the latter is a corner detection heuristic. This combination has shown to perform particularly well in our mobile application and can be computed efficiently. By projecting the resulting 2D features to 3D using the disparity map, the 3D transformation can be estimated.

However, neither of the aforementioned features contain structural information, nor are they robust to occlusions or appearance changes that result from severe viewpoint differences. To tackle such appearance issues, Morel *et al.* [14] introduced synthesized views that simulate different viewpoints in Affine-SIFT (ASIFT). An affine transformation applied to the original image yields a synthesized view, approximating the scene as seen from a different viewpoint. In ASIFT, the full range of synthetic viewpoints is simulated, which has shown attractive performance in registering planar scenes from different viewpoints. However, the execution time is excessive because each view is matched separately. A recent variant to ASIFT, Matching On Demand with view Synthesis (MODS) [13], iteratively applies more time-consuming feature detectors in a progressive way, which employ a robust variant of SIFT, called RootSIFT [1]. Additionally, each iteration applies a predefined amount of viewpoint simulations. Compared to ASIFT or 2D features, MODS shows a huge performance increase on non-planar scenes, but its iterative nature is undesirable for real-time application.

Contrary to the above-mentioned methods in which no a-priori knowledge about the viewpoint difference is assumed, we exploit the GPS/IMU position and previous vehicle displacements to synthesize a single relevant view. This synthesized view neutralizes image appearance differences of the ground plane due to a different viewpoint, which allows the use of the efficient FAST and BRIEF features.

Baseline system

The baseline image registration method [25] can be summarized in four stages, as visualized in Figure 2. Prior to these stages, live and historic images featuring the same scene from different viewpoints, are paired using GPS and vehicle heading. Next, a depth value is obtained for every pixel in the historic image through disparity estimation.

Figure 3: Prototype vehicle used for our experiments, featuring a stereocamera with 1920×1440 pixel resolution and GPS/IMU positioning, achieving decimeter accuracy.

In the first stage of the image registration, the historic scene is modeled by projecting texture onto a 3D model of the historic scene, yielding a textured 3D polygon model. The implementation of this model can be found in [25]. In the second stage, a 3D transformation between the live and historic scene is calculated. This transformation is subsequently used in the third stage to transform the historic 3D model, as if it was viewed by the live camera. The transformed model is then projected to 2D, resulting in an historic image that is aligned with the live image. This initial registration may have local misalignments, e.g. shifts of several pixels, due to small inaccuracies in either the 3D scene reconstruction or the 3D transformation. In the last stage, the registration is refined using an optical flow method to achieve a pixel-precise registration.

The baseline system estimates the 3D transformation through 2D feature matching. First, the FAST corner detector together with the descriptor of Oriented FAST, Rotated Brief (ORB) [20] are applied to find features. Next, 2D correspondences are generated between the live and historic features using a full-search bi-directional matcher, where only one-to-one matches are accepted. Additionally, false correspondences are filtered with the Second Nearest Neighbor (SNN) [12] similarity constraint. The resulting set of feature correspondences is projected to 3D using world coordinates obtained from the depth map. The geometrical constraint from Hirschmuller [7] further removes outliers from the set of 3D correspondences. From this improved set of feature correspondences, a rigid 3D transformation between the two scenes is estimated using the 3D RANdom SAmple Consensus (RANSAC) algorithm. This baseline 3D transformation estimation method performs well for lateral displacements up to 2.5 m, but performance declines rapidly for larger displacements.

Approach

In this work we propose an improved 3D transformation estimation that is more robust to inaccuracies in the depth map, occlusions of objects and ambiguities in the scene, such as repeating patterns. For this reason, we introduce a single synthesized view to cope with large perspective distortions. Furthermore, we minimize the influence of inaccuracies in the depth map, by applying an additional refinement to the set of 2D-3D correspondences.

Figure 4 shows the flow of our method in four stages. First, view synthesis is applied. To reduce the computational load of this stage, we choose to create a single synthesized view of the live ground surface, as appearance changes due to viewpoint variations are most apparent in this area, e.g. the ground surface in Figure 1 is significantly more distorted than the lighting pole. To this end, the historic and live image are first split into separate ground surface and obstacle views. These views can be interpreted as images containing only the ground surface or obstacle pixels, where the ground/ obstacle division is made using the 3D historic scene model[1]. Second, features are computed for each view separately. Third, point correspondences are obtained and false correspondences are rejected. The remaining correspondences are projected to 3D using the depth map. Fourth, the 3D correspondences are used to estimate an initial transformation, which is later refined using a set of 2D-3D correspondences, thereby minimizing the registration error in 2D. The following paragraphs provide a detailed description of these four steps.

View synthesis

The proposed pipeline starts with an initial estimate of the lateral displacement of the vehicle w.r.t. the previous driving trajectory, which is based on a temporal filtering of the GPS/IMU data and the previously estimated transformation. This estimate is used to synthesize a single relevant view.

View syhthesis increases appearance similarity between images acquired from different viewpoints. Performing view synthesis on an image involves applying a 2D transformation to that image, which is known to be valid only for a single world plane. In this work, the live ground plane is synthesized to increase the appearance similarity with respect to the historic ground plane (see Figure 5). The focus lies on canceling appearance changes due to lateral displacements of the vehicle, as these are most common in practice, e.g. driving in a different lane.

It was shown by Ranft [17] that the 2D transformation for simulating a slanted world plane as seen by another camera (in stereo setup), simplifies to a shearing transformation. However, this only holds when the pair of images are rectified. Rectification implies that the camera centers are displaced in the x-direction only and there is no angular difference between the optical axis of the cameras. As we are interested in approximating image appearances, we can neglect small differences in the y-direction or in image scale and only rectify the angular differences of the optical axis.

The process of view synthesis by angular rectification and shearing the live ground surface is shown in Algorithm 1. Rectification of the live optical axis with respect to the historic axis,

[1] In reality regions of interest (ROIs) are used. However, for ease of explanation, we refer to these as separate ground surface and obstacle views instead.

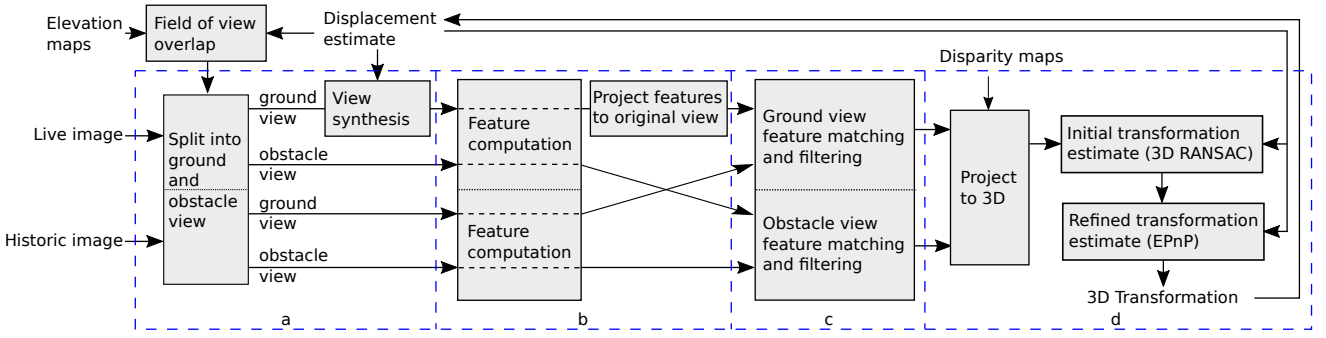
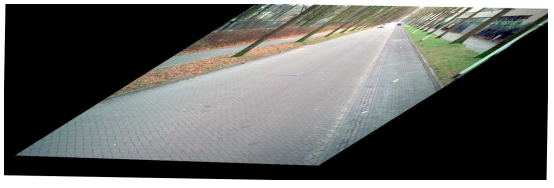


Figure 4: Block diagram of the proposed method. First, the live and historic images are split into separate ground surface and obstacle views. Second, features are computed for each view separately. Third, the features are matched, false matches are filtered from the correspondences and the resulting correspondences are projected to 3D using the depth map. Finally, a transformation is estimated from the 3D correspondences and later refined.

(a) Historic image

(b) Live image at 3.5m displacement



(c) Synthesized view of the live image

Figure 5: An example of a synthesized view of the live image. The image is transformed such that the ground surface is depicted as viewed from a different angle. This transformation is clearly not applicable to pixels not belonging to the ground surface.

is achieved using a rotational homography $H_{\Delta R}$ [5] (Line 9 of Algorithm 1), computed from the angular difference ΔR between the optical axes. Due to vibrations of the vehicle disturbing the onboard IMU, the roll and pitch angles supplied by the IMU are inaccurate. Instead, the roll and pitch angles are derived from an estimation of the ground surface normal and the yaw angle is derived directly from the IMU. This is shown in Lines 2–6 of Algorithm 1, where α, β and γ are the pitch, roll and yaw angles towards the ground surface, respectively. The actual shearing transformation A only requires a shearing gradient g (Line 10, Algorithm 1). This gradient is calculated from the initial lateral displacement δ between camera poses, the measured height h_v of the cameras atop the vehicle and the absolute roll angle of the historic ground surface β^H. The synthesized view is finally created by applying the rectification ($H_{\Delta R}$) and shearing (A) transformations to the live image.

Feature matching

The FAST corner detector is combined with the BRIEF descriptor, which have shown good performance under perspective transformations [6]. From Figure 4 it can be observed that features are matched independently for the ground and obstacle views. This reduces the amount of false matches and is more efficient than jointly matching all features. A full-search matcher is used to match the features based on the Hamming distance between their descriptor vectors. False matches are filtered from the initial correspondences using the Second Nearest Neighbor (SNN) ratio, where a match is rejected if the first and second nearest matches are too similar.

Transformation estimation

A set of 3D feature correspondences is obtained by projecting the 2D correspondences from the previous section to 3D, using the depth values obtained from the disparity map. These 3D correspondences are then used to estimate an initial rigid 3D transformation, by minimizing the Euclidean distance in 3D.

Algorithm 1 View synthesis of the live ground surface

Input: I^L - Live image, δ - displacement estimate between live and historic camera pose, (S^L, S^H) - Live and historic 3D ground surface
Output: I^S - Synthesized live image
Parameters: K - Intrinsic camera matrix, h_v - Height of camera

1: **procedure** SYNTHESIZEVIEW(I^L, δ, S^L, S^H)
2: **for** $i \in \{L_{(ive)}, H_{(istoric)}\}$ **do**
3: N^i = **EstimateSurfaceNormal**(S^i)
4: $\alpha^i = \text{atan}(N^i_y / N^i_z)$
5: $\beta^i = \text{atan}(N^i_y / N^i_x)$
6: γ^i = **RetrieveYawFromIMU(i)**
7: **end for**
8: $\Delta R = R_z(\beta^L - \beta^H) \cdot R_y(\gamma^L - \gamma^H) \cdot R_x(\alpha^L - \alpha^H)$
9: $H_{\Delta R} = K \cdot \Delta R \cdot K^{-1}$
10: $g = \delta / h_v \cos\beta^H$
11: $I^S(x) = I^L(A \cdot H_{\Delta R} \cdot x)$, $A = \begin{bmatrix} 1 & g & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
12: **return** I^S
13: **end procedure**

However, due to inaccuracies in the depth map, which typically involve a high uncertainty in the localization of the point on the ray through the camera center (pinhole camera model), the initial transformation in 3D may not correspond to an optimal registration after back projection to 2D. To reduce the effect of such 3D inconsistencies, an additional transformation refinement is employed that minimizes the reprojection error in 2D. This is accomplished using the Efficient Perspective-n-Point camera-pose estimation algorithm, EPnP [10]. Since the 2D projection of a 3D point is unaffected by the localization of this point on the ray through the camera center, the resulting transformation is less affected by 3D inconsistencies in the viewing direction.

Although an initial 3D transformation estimate is strictly not necessary for the EPnP algorithm, the accuracy of EPnP has shown to improve significantly when less spatial outliers are present [10]. Both the estimation of the initial transformation and the EPnP algorithm are wrapped in a RANSAC scheme to improve the robustness of the solution.

Efficiency optimizations

Several efficiency optimizations are included in our implementation to ensure a transformation can be estimated in real-time. These optimizations are as follows:

- Only image regions withing the overlapping Field of View (FoV) of both cameras are processed.
- Time-critical elements are executed on a GPU. For this reason, we have created our own GPU implementation of BRIEF and used GPU algorithms from OpenCV [3] for FAST, full-search feature matching and image warping.
- Memory overhead is reduced by cropping the synthesized view to the relevant ground-surface region. This also reduces the overhead from GPU data transfers.

The Section '*Timing evaluation*' near the end of this paper, shows the execution time with and without these optimizations.

Experiments & Results

Both the transformation accuracy and the execution time of the proposed method are evaluated for several realistic scenarios. Results are compared to the baseline system as well as the state-of-the-art MODS algorithm. The experiments focus on challenges that typically occur for moving vehicles: different driving trajectories, different viewpoints and illumination changes. Furthermore, the contribution of the synthesized view to the alignment accuracy is explicitly addressed.

Evaluation framework

This work focuses on estimating an accurate 3D transformation, which is used to register images acquired from different viewpoints. Therefore, the registration error between the live and aligned-historic frame can be used as an evaluation metric as follows. Corresponding points in both historic and live images are manually annotated. Similar to the techniques described in Section '*Approach*', the historic annotations can be converted to 3D points using the depth map. By applying the estimated 3D transformation, the historic annotations are transformed to the (3D) coordinate frame of the live camera. After re-projecting the transformed historic annotations to 2D, the historic annotations should now be aligned to the live annotations.

Throughout this paper, we define the *transformation accuracy* as the percentage of annotations with an alignment error below 5 pixels. It has been observed that the manual annotation process itself causes alignment errors that are on the average below 2 pixels.

Results

This section evaluates the transformation accuracy of the proposed, the baseline and the state-of-the-art MODS system, where MODS replaces stage a-c in Figure 4. Table 1 summarizes the results of all experiments, which are now discussed separately. Figure 6 shows examples of the challenges encountered in our datasets, which are evaluated in this section.

Figure 6: Illustration of the challenges encountered in our datasets. The left and right images show the same scene under a different: lateral displacement (top), driving orientation (center) and with dynamic changes in the scene (bottom).

Lateral displacements: This experiment contains parallel driving trajectories with different lateral displacements between the historic and live trajectories (of 0 cm, 160 cm, 350 cm, 530 cm and 700 cm). This simulates the effect of driving in a different lane.

Figure 7, shows the alignment-error histograms for different alignment algorithms, including the proposed method. For such histograms, the energy is ideally located at the left side of the histogram. From Table 1 (first 5 columns), it can be observed that the proposed method achieves over 95% accuracy up to 350-cm lateral displacement and still shows 85% accuracy at 530-cm displacement. The MODS algorithm, when embedded in the proposed evaluation framework, shows similar accuracy as the proposed method, where it should be noted that its execution time is approximately 30 to 80 times slower. The baseline system consistently fails to find accurate transformations at displacements of 350 cm and larger and already drops to 49% at 160 cm.

Table 1: Transformation accuracy and execution time results for the baseline method, the proposed method and the state-of-the-art MODS algorithm integrated in our evaluation framework. *method does not apply EPnP refinement to cope with 3D model inconsistencies.

	lateral 0cm		**lateral 160cm**		**lateral 350cm**		**lateral 530cm**		**lateral 700cm**		**illumination**		**orientations**	
	acc. [%]	time [ms]	acc. [%]	time [ms]	acc. [%]	time [ms]	acc. [%]	time [ms]	acc. [%]	time [ms]	acc. [%]	time [ms]	acc. [%]	time [ms]
Baseline [25]*	99	256±27	49	242±13	27	229±13	19	253±12	6	269±17	86	377±20	5	101±97
Proposed	100	100±26	98	72±9	95	65±5	85	68±7	64	64±14	99	108±10	90	110±37
MODS [13]	100	8312±194	99	7742±160	88	7548±124	80	7586±189	64	20539±7207	97	19021±10587	66	12251±6769

These results agree with the initial transformation accuracy reported in [25], where the accuracy was significantly improved by the 2D optical flow refinement (not applied here). It should be noted that the baseline is evaluated without using the EPnP algorithm.

Varying orientations: This experiment consists of images where the vehicle orientation between the live and historic trajectory deviates more than 15°. This limits the overlap in Field Of View (FOV) and introduces perspective distortion between the reference and live images. The transformation accuracy is shown in Table 1, Column 7. The proposed method achieves a high accuracy, around 90%. Although MODS shows promising results in the previous experiment, its performance degrades for different viewing angles. This can be explained by the limited overlap in the FoV of the live and reference images, which is only taken into account by the proposed system. When this view overlap is small, processing more than the overlapping region results only in more false correspondences. Similar to the lateral displacements experiment, the baseline system cannot handle the viewpoint differences and shows poor accuracy.

Illumination changes: This experiment focuses on images with both global and local illumination changes, e.g. shadows. The transformation accuracy is shown in Table 1, Column 6. It can be directly observed that all methods perform exceptionally well, which can be explained as all employed features are invariant to small illumination changes.

Limitations of the Synthesized view: To evaluate the practical limitations of the synthesized view, the lateral displacement experiments are repeated, however, this time with the obstacle views omitted, i.e. only the ground surface is used for estimating the 3D transformation. The proposed method is then evaluated with and without using a synthesized view, to clearly validate the advantage of using such a synthesized view. Furthermore, we employ the system in an environment that violates some of the assumptions behind the synthesized views, i.e. a non-flat ground surface. Similar to the previous experiment, feature matching is limited to the ground surface and the system is evaluated with and without the synthesized view.

Figure 8 shows the alignment-error histograms with and without the synthesized views. The 0-cm and 160-cm datasets are not shown, because the transformation accuracy for those datasets is comparable to the results on the 350-cm dataset, i.e. over 95% accuracy, both with and without synthesized views. It can be concluded that the synthesized view significantly improves performance for a lateral displacement of 530 cm, but is not strictly necessary for estimating a transformation at smaller displacements. Additionally, at 700-cm lateral displacement, the transformation for generating a synthesized view induces such significant distortion, that hardly any correspondences are found, thus resulting in large transformation errors. The result of the 700-cm histogram also implies that the accuracy for the 700-cm test reported in Table 1 is mainly based on feature correspondences from the obstacle views.

Finally, the histograms in Figure 9 show that even though the synthetic view is applied to a terrain with a non-flat surface, the registration accuracy is only slightly affected.

Timing evaluation

All implementations are evaluated on a system comprised of an i7-3960X 3.3-GHz hexacore processor with 16-GB RAM, a GeForce GTX Titan X GPU and a 256-GB SSD, running Ubuntu 14.04. Table 2 reports the improvement in execution time from the efficiency optimizations introduced in the equally named section. Table 1 shows the execution time of the proposed, the baseline and the state-of-the-art MODS system for the various experiments. From this table it is clear that the proposed method executes significantly faster than the other methods, in the order of 100 milliseconds versus seconds for the state-of-the-art algorithm.

Discussion & Recommendations

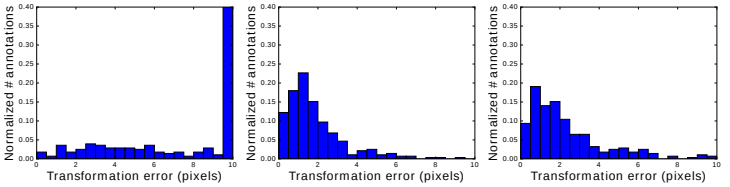
It was shown that the alignment accuracy is comparable for displacements up to 350 cm, regardless whether the synthesized view is applied or not. Furthermore, it was shown that the synthesized view is no longer beneficial when the displacement becomes too large, e.g. 700 cm. Only in specific scenarios, where the lateral displacement is between 400 and 600 cm, the synthesized view significantly improves the alignment accuracy. We therefore argue to apply it only in those situations where the displacement estimate is between 400 and 600 cm.

The reader may have noticed the absence of a registration refinement stage, as employed in [25]. The method proposed in this paper would also benefit from this additional refinement, which was beyond the scope of this work.

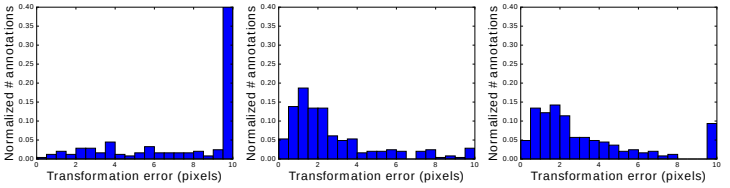
At this point, we also list a recommendation for the near future. The estimate of the displacement is currently an approximation using a weighted moving average filter. Although our method is robust to small errors of the estimate, the displacement estimate is always lagging behind on the actual displacement. Therefore, a prediction of the displacement that would exploit both visual information as well GPS/IMU data, would be more robust to displacement changes. Prediction or localization algorithms, such as an Extended Kalman Filter (EKF) [8], or a visual-inertial SLAM [11] algorithm, could be applied to reduce the difference between our estimated and the actual displacement.

Conclusions

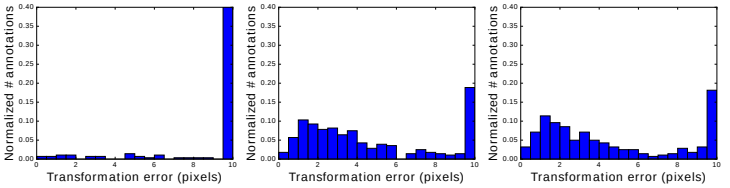
The contributions of this paper are twofold. First, we have introduced an improved method for estimating a robust 3D transformation in cluttered environments, which facilitates the registration of images captured under large viewpoint differences.



(a) Baseline, 350 cm (b) Proposed, 350 cm (c) MODS, 350 cm

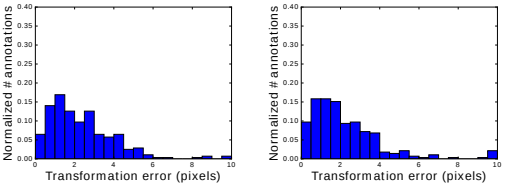


(d) Baseline, 530 cm (e) Proposed, 530 cm (f) MODS, 530 cm

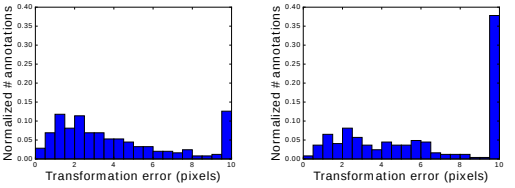


(g) Baseline, 700 cm (h) Proposed, 700 cm (i) MODS, 700 cm

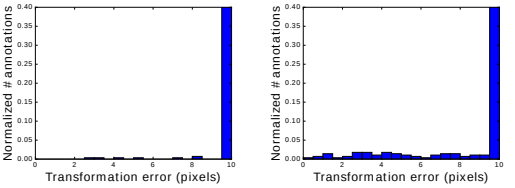
Figure 7: Alignment error histograms for the baseline method, the proposed method, and the state-of-the-art MODS algorithm integrated in our evaluation framework, evaluated for different lateral displacements. Alignment errors above 10 pixels are included in the rightmost bin.



(a) 350 cm, with synth. view (b) 350 cm, no synth. view



(c) 530 cm, with synth. view (d) 530 cm, no synth. view

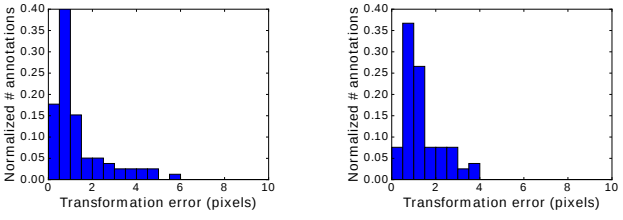


(e) 700 cm, with synth. view (f) 700 cm, no synth. view

Figure 8: Alignment-error histograms of the 350-, 530-, 700-cm lateral displacement for our proposed method with and without the synthesized view, using the ground surface only.

Table 2: Improvement in execution time resulting from the efficiency optimizations introduced in Section '*Efficiency optimizations*'

	Pre-processing	**View synthesis**	**Feature computation**	**Feature matching**	**Initial transformation**	**Transformation refinement**	**Total**
Proposed	2±1	63±2	136±14	259±43	10±5	7±13	474±59
Proposed optimized	2±1	5±0	19±2	46±10	10±5	18±29	100±26



(a) irregular, no synth. view (b) irregular, w. synth. view

Figure 9: Alignment-error histograms of the irregular terrain dataset, which violates the synthesized view assumptions, for our proposed method with and without the synthesized view. The results are based on processing the ground surface only.

The proposed method has a significantly higher accuracy than the baseline system and is comparable with, or even outperforms the computationally more expensive state-of-the-art MODS method. Second, we have demonstrated real-time execution resulting from efficiency optimizations and a GPU implementation of the time-critical elements.

The proposed method achieves over 95% accuracy for lateral displacements up to 350 cm, while still achieving over 85% transformation accuracy at a displacement of 530 cm. Moreover, the method has shown to be robust to different viewing directions, illumination changes and irregular terrain. It was observed that it is sufficient to apply the synthesized view in specific scenarios only, i.e. when the lateral displacement estimate is between 400 and 600 cm.

We have significantly improved the operational range of the mobile registration system by increasing the robustness to lateral displacements from 2 m in the baseline system, to more than 5 m in the novel system.

References

[1] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2911–2918. IEEE, jun 2012.

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[3] Gary Bradski et al. The opencv library. *Doctor Dobbs Journal*,

25(11):120–126, 2000.

[4] M Calonder, V Lepetit, M Ozuysal, T Trzcinski, C Strecha, and P Fua. Brief: Computing a Local Binary Descriptor Very Fast. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1281–98, jul 2012.

[5] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[6] Jared Heinly, Enrique Dunn, and Jan-Michael Frahm. *Comparative Evaluation of Binary Features*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, oct 2012.

[7] H. Hirschmuller, P.R. Innocent, and J.M. Garibaldi. Fast, unconstrained camera motion estimation from stereo without tracking and robust statistics. In *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002.*, volume 2, pages 1099–1104. Nanyang Technological Univ, 2002.

[8] Gregory Hitz, François Pomerleau, Marie-Eve Garneau, Cédric Pradalier, Thomas Posch, Jakob Pernthaler, and Roland Y Siegwart. Autonomous inland water monitoring: Design and application of a surface vessel. *Robotics & Automation Magazine, IEEE*, 19(1):62–72, 2012.

[9] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[10] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International journal of computer vision*, 81(2):155–166, 2009.

[11] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[12] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[13] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141:81–93, 2015.

[14] Jean-Michel Morel and Guoshen Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[15] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275, 2010.

[16] Timo Ojala, Matti Pietikinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[17] Benjamin Ranft and Tobias Strauß. Modeling arbitrarily oriented slanted planes for efficient stereo vision based on block matching. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 1941–1947. IEEE, 2014.

[18] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[19] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: a machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–19, jan 2010.

[20] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[21] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152. IEEE Comput. Soc, 2001.

[22] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[23] Aleksandr V Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. *Proc. of Robotics: Science and Systems*, 2:4, 2009.

[24] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010*, pages 356–369. Springer, 2010.

[25] Dennis W. J. M. van de Wouw, Gijs Dubbelman, and Peter H. N. de With. Hierarchical 2.5-d scene alignment for change detection with large viewpoint differences. *IEEE Robotics and Automation Letters*, 1(1):361–368, Jan 2016.

[26] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. IEEE, 2009.

Author Biography

Dennis van de Wouw received his MSc in Electrical Engineering at Eindhoven University of Technology (TU/e) in 2011. He joined ViNotion B.V. as an R&D engineer, where he worked on a change detection system for countering Improvised Explosive Devices. In March 2013 he continued his research as a PhD student of the Video Coding and Architectures research group at the TU/e, where he focuses on change detection for intelligent vehicles.

Martin Pieck graduated cum laude from Eindhoven University of Technology (TU/e) in Electrical Engineering in 2016. He joined ViNotion B.V. as an R&D engineer, where he works on the subject of automatic pedestrian and traffic analysis.

Dr. Gijs Dubbelman is an assistant professor with the Eindhoven University of Technology and focuses on signal processing technologies that allow mobile sensor platforms to perceive the world around them. He obtained his PhD. research in 2011 on the topic of Visual SLAM. In 2011 and 2012 he was a member of the Field Robotics Center of Carnegie Mellon's Robotics Institute, where he performed research on 3-D computer vision systems for autonomous robots and vehicles.

Peter H. N. de With (MSc. EE) received his PhD degree from University of Technology Delft, The Netherlands. After positions at Philips Research, University Mannheim, LogicaCMG and CycloMedia, he became full professor at Eindhoven University of Technology. He is an (inter-)national expert in surveillance for safety/security and was involved in multiple EU projects on video surveillance analysis with the Harbor of Rotterdam, Dutch Defense, Bosch Security, TKH-Security, ViNotion, etc. He is board member of DITSS and R&D advisor to multiple companies. He is IEEE Fellow, has (co-)authored over 300 papers on video analysis, systems and architectures, with multiple awards of the IEEE, VCIP, and EURASIP.