https://doi.org/10.2352/ISSN.2470-1173.2019.7.IRIACV-465

Improved 3D Scene Modeling for Image Registration in Change Detection

Sjors van Riel; Eindhoven University of Technology; Eindhoven, the Netherlands
Dennis van de Wouw; Eindhoven University of Technology, ViNotion B.V.; Eindhoven, the Netherlands
Peter de With; Eindhoven University of Technology; Eindhoven, the Netherlands

Abstract

This paper presents a novel method for 3D scene modeling using stereo vision, with an application to image registration. The method constists of two steps. First, disparity estimates are refined, by filling gaps of invalid disparity and removing halos of incorrectly assigned disparity. A coarse segmentation is obtained by identifying depth slices, after which objects are clustered based on color and texture information using Gabor filters. The second step consists of reconstructing the resulting objects in 3D for scene alignment by fitting a planar region. A 2D triangle mesh is generated, and a 3D mesh model is obtained by projecting each triangle onto the fitted plane. Both of these extensions result in improved alignment quality with respect to the state of the art, and operate in near real time using multi-threading. As a bonus, the refined disparity map can also be used in combination with the existing method.

Introduction

Image registration remains an important computer vision challenge and involves the process of aligning pairs of images into the same coordinate system. Typical registration applications include remote sensing [1],[2], image stitching [3], robotics, medical domains [4] and change detection. Image alignment techniques can generally be categorized as either global or local. Global methods aim at computing a single transformation between an image pair, such as a homography transformation. These methods generally assume planar scenes, although they are also often applied in remote sensing. In such a case, the scene typically contains objects at large distances from the camera and may be considered approximately planar.

A recent development is the use of Deep Learning and Convolutional Neural Networks (CNNs) for homography estimation. Detone *et al.* [5] train a deep CNN called HomographyNet that performs end-to-end learning of homography parameters from two input images. A similar approach is used by Chang *et al.* [6], which uses a CNN consisting of a feature point extractor, cascaded with a Lucas-Kanade layer. Both approaches learn the relevant features and transformation computation simultaneously during training, but these approaches are limited to a global transformation, which is not applicable to our problem.

Local methods, such as proposed by Lou *et al.* [7] first perform segmentation into planar regions in the image and then attempt to find a transformation per segmented region. Computing a transformation for each region is typically performed by matching feature points, of which SIFT [8] is most widely used. Alternatively, pixel-based methods such as optical flow can estimate a local transformation, but generally lack illumination invariance and cannot handle dynamic changes between the images. An in-depth survey of image registration methods can be found in [9].

In this work, we investigate the alignment of video frames recorded from a moving vehicle over the course of two separate drives along the same trajectory. In the context of repetitive capturing, our objective is to detect changes that have occurred in the environment between the two recordings. For this purpose, we aim to align a previously recorded frame, which we will refer to as the 'reference' frame, onto the live frame during driving. Since the reference and live vehicle trajectories never exactly coincide and may be recorded under different recording conditions, large illumination changes and viewpoint differences may occur. The latter introduces parallax effects, where the relative position and ordering of objects change with respect to each other and the background. This invalidates the use of global alignment methods, as parallax effects cannot be compensated by a single affine transformation.

Diego *et al.* [10] propose an algorithm for video frame alignment from a moving vehicle using inference on Bayesian networks. While results are promising, this approach is aimed at driving trajectories within the same lane, corresponding to small camera displacements of up to 2.5 m. Furthermore, it does not take parallax effects into account. In order to obtain robust scene alignment, we perform alignment in 3D, where a rigid transformation between the 3D scenes can be found. To this end, we mount stereo cameras on our vehicle to capture the 3D scene geometry. Typically, scenes from two recordings can then be registered by aligning the 3D point clouds obtained from this stereo camera setup. Finally, the aligned point cloud can be projected back to 2D, resulting in an aligned image.

While alignment of 3D point clouds has been investigated in numerous studies, most approaches assume accurate and densely sampled point clouds, such as those obtained from a laser scanner. However, disparity estimates generated from stereo images are typically noisy and contain many outliers. Hence, when projecting back to 2D, noisy 3D points may be mapped to the wrong 2D coordinates. To circumvent this problem, point clouds can be approximated by a noise-robust 3D surface model, onto which texture can be projected. Such models typically outperform point-based processing, as they may remove holes in the resulting image.

Van de Wouw *et al.* [11] have proposed a registration approach using such a textured 3D model, which differentiates between a ground plane and objects, both being combined into a single model. Although this approach produces accurate ground-plane alignment, detailed object modeling remains a challenge.

In the field of 3D object modeling, many methods aim at

creating a mesh model. Creating a mesh model from noisy data is typically not feasible. Therefore, approximating a model using piecewise planar regions is a commonly used method [12], [7]. Planar models can produce promising results when using multi-view stereo data [13]. Therefore, we adopt this method of modeling 3D objects using planar regions.

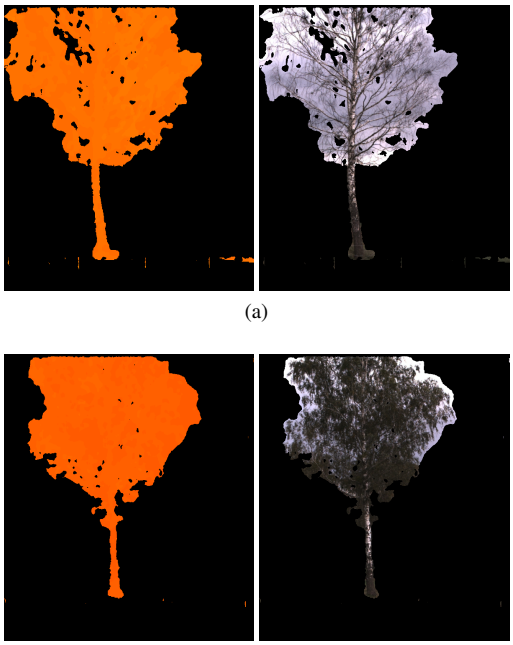
In this work, we contribute with a novel method for modeling 3D objects, which can be used to complement or even replace the modeling approach of [11]. In our approach, objects are segmented from 2D disparity maps and refined using color and texture information. We perform per-object 2D-triangulation and project the resulting triangles onto a planar region to obtain a 3D object mesh. Furthermore, we aim at near real-time execution of our reconstruction algorithm, to allow for overnight processing of recorded data.

This paper is structured in the following way: We first present the baseline system from van de Wouw *et al.* [11] and investigate its limitations. The proposed method is discussed in the next section. Several experimental results are provided using semi-automatic validation, and finally we draw conclusions and provide recommendations.

Baseline scene alignment

In this paper, we build further upon the work of Van de Wouw *et al.* [11], who have proposed a system for 3D scene modeling by introducing a diorama-box model. Objects are modeled by upright fronto-parallel rectangular super-pixels called stixels, superimposed on a model of the ground surface. These stixels are efficiently implemented by the Stixel World algorithm. *Stixels* are constructed by first sampling the image horizontally along a fixed grid. The resulting columns are then split in order to obtain super-pixels corresponding to either ground or obstacles. Using depth estimates obtained from the stereo camera, each stixel is converted to a fronto-parallel 3D planar model. Next, a number of post-processing steps are applied to improve the stixel model quality. First, stixels are slanted according to the modeled object orientation, and non-connecting stixels are interpolated in order to guarantee a watertight model. Additionally, pixels that do not satisfy the stixel orientation (such as background pixels) are rendered as black pixels to prevent distorted pixels in the projected 2D image. Although this model produces promising results, it contains several limitations. First, stixels are only slanted with respect to the viewing direction, creating an approximate projection of the scene, which may introduce alignment errors for non-upright objects. Second, the rectangular grid-like nature of stixels requires very thin stixels in order to be able to model thin (vertical) objects. This implies that a large number of stixels is required to cover the entire scene. This method of sampling is therefore not well suited for rendering non-vertically-oriented objects. In constrast, our proposed method is not restricted to vertical objects, while containing only a minor preference for fronto-parallel oriented objects. Moreover, the proposed method is better suited to capture the shape of the object, as it is no longer restricted to rectangular models.

Another issue that accompanies the use of stereo cameras is that of poor disparity estimates. Disparity is typically estimated using a block matcher. However, it often produces incorrect results around edges of objects or in poorly textured areas. The former produces areas of incorrect disparity around objects,



(a)

(b)

Figure 1: Examples of large halo effects caused by invalid disparity estimates, where background pixels are assigned disparity values of adjacent objects. This effect cause severe warping distortions in the aligned image. Left: disparity-based object segmentation, Right: corresponding image region.

which we will refer to as *halos*. Examples are shown in Fig. 1, which shows blobs where the block matcher cannot discriminate between objects and background. These halos will result in distorted pixels in the aligned image.

Our method

This section presents a novel method for 3D scene modeling for image registration. This method aims at accurate surface reconstruction of 3D objects in the presence of noisy depth data originating from a stereo camera. Our method consists of two steps. First, the scene is accurately segmented by combining depth, color and texture information, and the disparity map is refined in correspondence with the objects found. In this stage, we also fill missing data in the disparity map (gaps of invalid disparity), at points where no occlusion is present. Second, a planar region is estimated per segmented object, which is then described by a 3D mesh model. An overview of the proposed processing steps is shown in Fig. 2, where each of the blocks is further detailed in the next paragraphs.

An important advantage of this two-step approach is the possibility of using it as an extension to the existing method. When applied as extension, the refined disparity map of the first stage can be used to create an improved stixel-based model. In this case, the refined disparity map is used as an input to the stixel-based algorithm, thereby directly improving its results.

Object segmentation and disparity refinement

In this processing block, the disparity map is refined by removing halos (Fig. 1) and filling gaps, i.e. regions of invalid disparity. Each of the four processing steps will now be described in

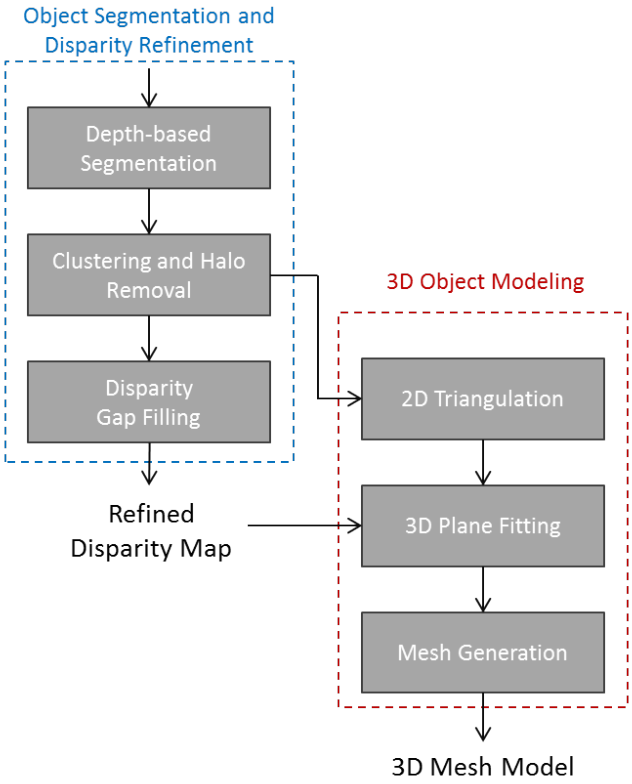


Figure 2: Block diagram of the proposed method

detail.

Depth-based segmentation

Although individual pixel disparity estimates may be unreliable, the overall disparity histogram contains important information about objects present in the scene. Therefore, we first indentify peaks in the histogram to obtain a coarse segmentation of the scene into slices of similar depth. This is obtained by smoothing the histogram, after which each slice is defined as the area around a local maximum, up to the nearest local minimum. A few of these slices are visualized in Fig. 3a. Note that a slice may contain multiple objects, such as the fence poles in the first layer of this figure. Each slice is represented by a binary mask in the corresponding image, from which individual objects are then segmented using connected components. This guarantees a full segmentation, where every pixel is classified as an object or as background.

Clustering and halo removal

In order to remove halos, background pixels should be classified. However, object or background appearances are not defined and may change between scenes or even between objects within a scene. Therefore, an unsupervised machine learning approach is adopted, where each object is clustered into two clusters using color and texture information. Color information is obtained by RGB color, while texture information is extracted by Gabor filters at four orientations. Note that this approach requires small intra-object variability compared to object-background variability. Clustering is performed using the K-means algorithm, as it has low computational complexity and allows real-time execution. Clustering results for several objects are shown in Fig. 3b, where the two clusters for each object are indicated. Note that due to the random cluster initialization of K-means for each object, it is a-priori unknown which cluster denotes the background. Determining the background cluster corresponding to a halo is not a trivial task. For humans, the clusters of Fig. 3b are easily recognized as trees or fence posts, but without a semantic interpretation, distinguishing the object cluster from the background becomes a challenging task. Assumptions cannot be imposed on size, shape or appearance of the halos, except that halos only occur at the edges of objects. A number of methods may be applied to find this outer region, such as region growing, active contours or snakes. However, we can also take prior scene information into account. In scenes recorded from a moving vehicle, we may assume that objects are connected to the ground plane. This is a robust check, independent of halo size or shape. Additionally, it is computationally less expensive than any contour-based approach, since it can be implemented by removing the least ground-connected cluster. This can be efficiently computed by dilating the object mask and performing a binary AND-operation with the ground mask. A small fraction of objects may not satisfy the constraint of being connected to the ground plane, such as small branches or overhead traffic signs. For these objects, the smallest contour cluster is classified as the object cluster, with an additional condition that the cluster must be partially included in the background cluster. By this condition, we impose a preference of no refinement over the risk of removing the object itself.

Disparity gap filling

The last disparity refinement step consists of gap filling. Candidate gaps are extracted using connected components on the missing-disparity mask (denoted by black pixels in Fig. 4). While it is possible to construct a gap-free disparity estimate, we specifically choose not to fill all gaps in the disparity map. For example, missing disparity estimates originating from occlusion do present meaningful information about the scene. Therefore, we only fill disparity gaps up to a maximum size of 50×50 pixels. This size was empirically determined. To ensure that only similar-colored regions are being filled, we impose a threshold on the median color difference between the gap and its surrounding area. This color difference is defined as the sum of individual channel differences. For robustness, thresholding was performed as a function of the standard deviation of the surrounding area:

$$\left|\text{Median}_{RGB}(I) - \text{Median}_{RGB}(J)\right| < T_{color} \cdot \sigma_{RGB}(J) \qquad (1)$$

where I denotes the gap area, J indicates the contour area surrounding the gap and σ represents the standard deviation. The subscript *RGB* indicates that the RGB color space was used. Finally, filling should be careful, to avoid filling background gaps between objects with different disparity values. Therefore, gaps outside the ground plane are only filled if the disparity around the gap is uniform, hence

$$\sigma_D(J) < T_{disparity} \cdot \mu_D(J) \qquad (2)$$

Again, index J incidates the gap contour area, and μ denotes the image mean. The subscript D indicates disparity processing. By imposing a threshold on its standard deviation dependent on the

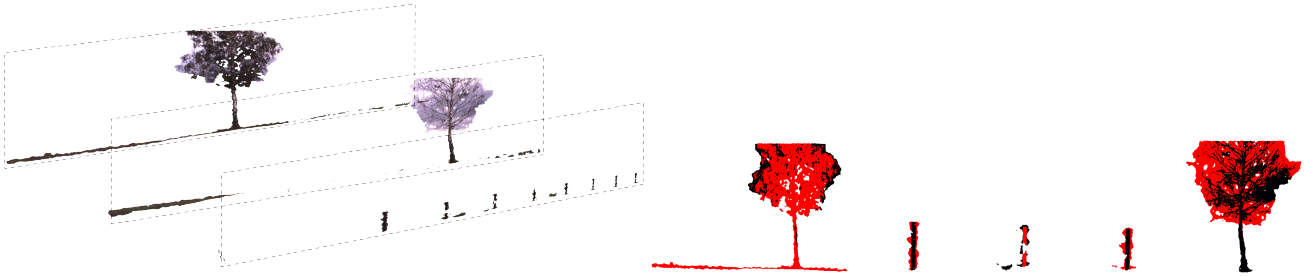


Figure 3: (a) Visualization of depth slices found from peaks in the disparity histogram. Note that for visualization purposes, slices are shown as having a single disparity, while in reality each slice represents a range of disparities. (b) Object refinement using K-means clustering. Each individual object is clustered into two clusters, shown in different colors. Note that the order of the clusters is non-deterministic, a result of the random initialization in the clustering algorithm.

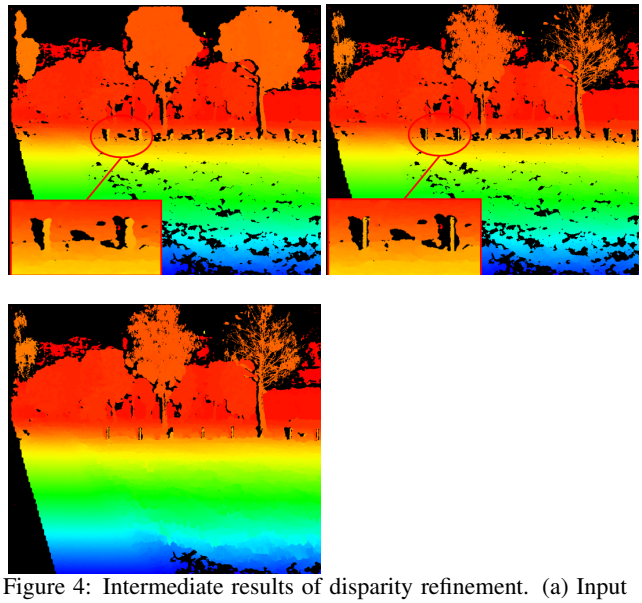


Figure 4: Intermediate results of disparity refinement. (a) Input disparity. (b) Disparity after object refinement. (c) Disparity after gap filling. Note the large refinement changes in the trees and detailed refinement of fence poles, corresponding to the clusters of Fig. 3b. After hole filling, occlusions (e.g. close to trees) remain intact.

mean, we incorporate the knowledge that disparity resolution progressively deteriorates for low disparity, as depth and disparity are inversely related. Additionally, for increasing depth, less pixels become available for matching. Therefore, the disparity uniformity constraint can be relaxed for low disparity. The remaining candidate gaps are now filled by an inpainting approach. Gap pixels are ordered into a priority queue according to an inward-facing spiral, after which each invalid pixel is replaced with a linear interpolation of the surrounding valid pixels.

Object modeling

3D scene reconstruction is performed on object basis, which can now be distinguished accurately from the object segmentation created in the disparity refinement algorithm. We would like to directly reconstruct each object with a 3D mesh model, but this is invalidated by the noisy depth estimation produced by a stereo camera, typically containing many outliers. Therefore, we choose

Figure 5: Triangulating a complex object (a tree with small and detailed branches).

to model each object by a planar structure. This approach consists of three steps: triangulation, plane fitting and mesh generation for each object, which are now briefly discussed.

Triangulation

First, a 2D polygon triangulation is constructed for each individual object from its 2D contour. We apply the ear-clipping algorithm [14] for triangulation, which operates in $O(n^2)$ time. Faster algorithms exist [15], but they are are not required for our use case, since triangulation time is negligible compared to computationally expensive vision tasks (like contour extraction).

Planar Region Fitting

Each segmented object is now approximated by a single plane, using Random Sample Consensus (RANSAC) on the noisy 3D data, which is robust to outliers. In contrast with the baseline approach, we have no assumptions on the plane orientation.

Mesh Generation

Once the 2D triangulation and the planar estimation are available, the final 3D mesh model can be obtained. This is performed by projecting each 2D triangle onto the fitted plane, by casting rays from the camera through each triangle corner, and computing the intersection with the plane. In this way, creating a mesh model enables modeling of arbitrarily shaped objects, such as tree branches, while the planar region guarantees that meshes connect in 3D. Projecting entails computing a new disparity estimate d' for each corner point (x, y), such that the resulting projec-

Method	TPR	TNR	Halos/obj
Stixel-based (Baseline)	0.515	0.990	0.738
Refined, Stixel-based	0.511	0.987	0.511
Refined, Mesh-based (ours)	0.528	0.988	0.585

Table 1: Performance of several algorithmic combinations on manually annotated data.

tion (X,Y,Z) by the camera projection matrix Q lies on the fitted plane. In homogeneous coordinates, this reduces to computing d' from

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = Q \begin{bmatrix} x \\ y \\ d' \\ 1 \end{bmatrix}, \tag{3}$$

such that

$$Z = aX + bY + cW \tag{4}$$

where a, b and c represent the fitted plane coefficients. Solving this system for each triangle corner point results in a set of 3D triangle meshes along the plane that models the object.

The final step of the algorithm consists of obtaining an aligned image from the scene model. First, the reference image texture is projected onto the mesh model. Then, a rigid transformation is estimated between the live and reference 3D point clouds, which is outside the scope of this paper. This transformation then allows the reference scene to be rendered from the live viewpoint.

Experimental results

The proposed method was evaluated on pairs of videos, recorded in a rural environment. Stereo cameras were mounted on a vehicle to record the environment while driving. The vehicle was equipped with GPS, which enables the automatic selection of a reference frame for each live frame, i.e. the reference frame with the most similar viewpoint with respect to the current viewpoint.

Evaluation

The proposed method can both be used as an extension to the existing algorithm and as a replacement of the baseline stixel-based model. Therefore, we perform experiments for three algorithmic combinations: (1) baseline stixel-based registration method; (2) disparity refinement in combination with the stixel-based model, where the refined disparity map is used as input to the baseline registration; and (3) the proposed method (including disparity refinement). An example of the aligned image for each method is shown in Fig. 6.

Validation of halo removal

Evaluation of image alignment algorithms is not a trivial task. In halo removal, we intend to find changes at pixel level, which invalidates common evaluation methods, such as keypoints. Instead, for six pairs of images all objects were manually annotated at pixel level. Corresponding objects were assigned the same color, while an additional color was used for the background. The annotated reference image was then transformed using the alignment transformation found for the original image, which produced an annotated aligned image. In this image, all objects were still represented with their respective color, while any background remaining in the image now indicates a halo.

Figure 6: Aligned images (cutout) for different algorithmic configurations. Top-l: baseline, top-r: baseline with refined disparity, bottom-l: proposed method, bottom-r: Target image.

A quantitative evaluation was now performed by comparing this image to the annotated live image. To this end, we define three performance metrics commonly used in classification and segmentation: the true-positive rate (TPR) denoting the ratio of object pixels that are correctly aligned, and the true negative rate (TNR) giving the fraction of background pixels that are correctly ignored. Finally, we denote the ratio of halos to object pixels as the halo rate, which should be minimized. By defining the halo rate in this way, the metric is not affected by the class imbalance present in these images, since there are typically many more background or ground pixels than object pixels. The measured TPR/TNR/halo results are depicted in Table 1. Clearly, the number of halos per object is sharply reduced by 15-20%, while the mesh-based and stixel-based rendering method perform on the average equally well.

Finally, it should be noted that the proposed algorithm executes in near real-time, allowing for overnight data processing.

Discussion

The proposed algorithm for halo removal by K-means clustering imposes no assumptions on appearance of objects or background. However, it is required that inter-object variability is larger than intra-object variability. While this is generally valid in rural scenes, this assumption often fails in urban scenes, resulting in holes in the model. Additionally, this problem can occur when large lighting differences are present within a single object. Therefore, future work should explore extending of the proposed

clustering algorithm using more than two clusters. Here, the extra complication again lies in finding the background cluster(s) from the object cluster(s).

Conclusions

We have proposed a novel 3D scene-modeling method for image registration based on a stereo camera. This approach consists of two parts. First, the disparity map from the stereo camera is refined by correcting for common errors in disparity estimation. Incorrect disparity estimations around objects (halos), i.e. background pixels that have been (incorrectly) assigned disparity values of the object, are identified from a coarse segmentation using machine learning. Next, gaps of missing disparity are filled, while informative invalid regions such as occlusions remain intact. This first part results in a refined disparity map thus serving as an extension, but it can also be applied as independent input to the existing baseline method, thereby directly improving model quality.

The second part consists of creating a 3D mesh model per segmented object. A robust model for each object is created by first fitting a planar region through noisy depth data, corresponding to that object. A 2D triangle mesh is then generated using an ear-clipping algorithm, and the resulting triangles are projected onto the fitted 3D plane, to obtain the final mesh model. The proposed method is better capable of modeling thin objects.

The novel algorithm has been evaluated on a set of image pairs, recorded from a moving vehicle in a rural environment. All objects present in these frames were manually annotated in order to evaluate the proposed halo-removal algorithm. This resulted in a reduction of the number of halo pixels per object of up to 20%. Finally, a similar performance improvement was observed when combined with the baseline (stixel-based) rendering method.

Both the disparity refinement and the 3D mesh model result in more accurate alignment. Finally, the algorithm operates in near real time to allow for offline overnight processing, while real-time online alignment is enabled during driving. Further improvement can be achieved by considering more clusters for halo removal and fitting multiple planes per object, whereas further speed optimization is obtained by exploiting GPU cores.

References

[1] Maoguo Gong, Shengmeng Zhao, Licheng Jiao, Dayong Tian, and Shuang Wang, "A Novel Coarse-to-Fine Scheme for Automatic Image Registration Based on SIFT and Mutual Information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 4328–4338, 7 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6619415/

[2] H. Goncalves, J. A. Goncalves, and L. Corte-Real, "HAIRIS: A Method for Automatic Image Registration Through Histogram-Based Image Segmentation," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 776–789, 3 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5570999/

[3] W.-Y. Lin, L. Liu, Y. Matsushita, K.-L. Low, and S. Liu, "Aligning Images in the Wild."

[4] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998. [Online]. Available: http://www.medicalimageanalysisjournal.com/article/S1361-8415(01)80026-8/pdf

[5] D. Detone, T. Malisiewicz, and A. Rabinovich, "Deep Image Homography Estimation." [Online]. Available: https://arxiv.org/pdf/1606.03798.pdf

[6] C.-H. Chang, C.-N. Chou, and E. Y. Chang, "CLKN: Cascaded Lucas-Kanade Networks for Image Alignment," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 7 2017, pp. 3777–3785. [Online]. Available: http://ieeexplore.ieee.org/document/8099885/

[7] Z. Lou and T. Gevers, "Image Alignment by Piecewise Planar Region Matching," *IEEE Transactions on Multimedia*, vol. 16, no. 7, pp. 2052–2061, 11 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6874530/

[8] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, 2004. [Online]. Available: https://www.cs.ubc.ca/ lowe/papers/ijcv04.pdf

[9] B. Zitová and J. Flusser, "Image registration methods: a survey."

[10] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez, "Video Alignment for Change Detection," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1858–1869, 7 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5648349/

[11] D. W. J. M. van de Wouw, W. P. Sanberg, G. Dubbelman, and P. H. N. de With, "Fast 3D Scene Alignment with Stereo Images using a Stixel-based 3D Model," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2018, pp. 250–259.

[12] A.-L. Chauve, P. Labatut, and J.-P. Pons, "Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2010, pp. 1261–1268. [Online]. Available: http://ieeexplore.ieee.org/document/5539824/

[13] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 9 2009, pp. 1881–1888. [Online]. Available: http://ieeexplore.ieee.org/document/5459417/

[14] H. ElGindy, H. Everett, and G. Toussaint, "Slicing an ear using prune-and-search," *Pattern Recognition Letters*, vol. 14, no. 9, pp. 719–722, 9 1993.

[15] R. E. Tarjan and C. J. Van Wyk, "An $O(n\log \log n)$-Time Algorithm for Triangulating a Simple Polygon," *SIAM Journal on Computing*, vol. 17, no. 1, pp. 143–178, 2 1988. [Online]. Available: http://epubs.siam.org/doi/10.1137/0217010

Author Biography

Sjors van Riel received his BSc degree from Eindhoven University of Technology in 2016, and his MSc degree in 2018. His fields of application include 3D scene analysis and medical image processing.

Dennis van de Wouw received his MSc degree from the Eindhoven University of Technology in 2011. As the change detection system architect at ViNotion, he is now finalizing his PhD thesis on depth-based extensions for change detection, in close collaboration with TU/e.

Peter H.N. de With is Full Professor of the Video Coding and Architectures group in the Department of Electrical Engineering at Eindhoven University of Technology. He is an IEEE Fellow, has (co-)authored over 400 papers on video coding, analysis, architectures, and 3D processing and has received multiple papers awards. He is a program committee member of the IEEE CES and ICIP and holds some 30 patents.

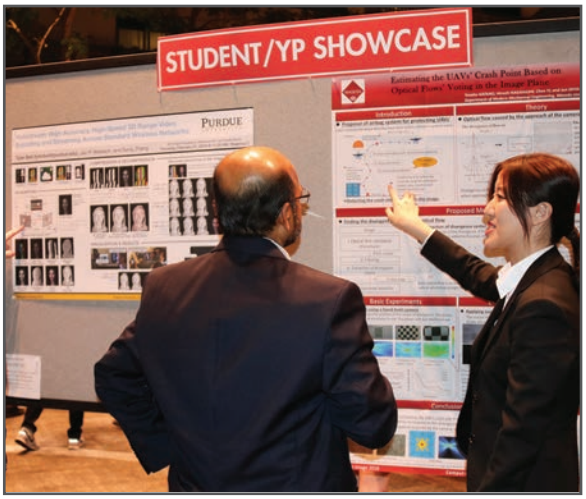
JOIN US AT THE NEXT EI!

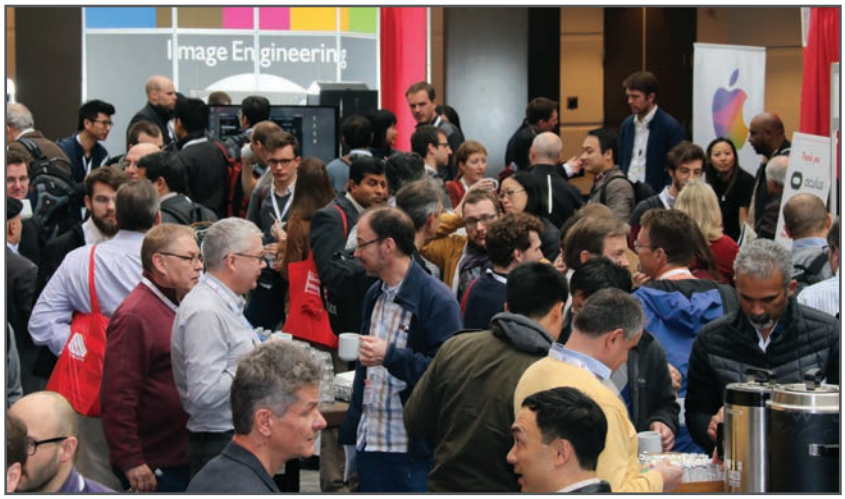
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org