Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

Reconstruction of Lattice-Like Objects on Mobile Devices

[Thomas Hunziker]

***Abstract*—Smart phones and tablet computers are typically equipped with a camera and a powerful processor, and together with the right computer vision software they can support engineers and other professionals in a broad range of tasks. Creating 3D models of real world objects is one job that can be very time consuming for humans if carried out manually, whereas with an adequate image processor the task can be completed very efficiently. In this paper we focus on the recognition of lattice-like objects made up of elements from a certain set of basic components. We propose a procedure incorporating a 2D detection of object parts, a 3D reconstruction, and a 3D model composition. The efficiency of the procedure is demonstrated by means of a prototype in the form of an Android app, accomplishing a 3D recognition and rendering of molecular models.**

***Keywords*—computer vision, pattern recognition, 3D modeling, bundle adjustment**

I. Introduction

Many applications of computer vision technology involve a recognition and reconstruction of 3D objects from 2D images. Fast and simple creation of 3D object models for further elaboration, pose estimation and target tracking, as well as object component recognition are tasks that typically require a 3D object reconstruction. The range of applications in these areas is broad and new possibilities have opened up recently with the availability of small hand-held devices – such as smart phones and tablets – combining cameras with great processing power.

A typical monocular camera performs a projection from the 3D space onto a 2D plane, thereby loosing depth information. Dealing with rigid objects the missing dimension can be recovered, in principal, provided that sufficiently many images of a target object are available. However, the problem of estimating the coordinates of object points in the 3D space is challenging for a number of reasons. First of all, the camera positions and orientations associated with the delivered 2D images are usually unknown and must be estimated along with the coordinates of selected 3D object points. Given there are m images – each of which is associated with a camera pose with six degrees of freedom – and n object points, the total number of parameters amounts to $6m+3n$. The joint estimation of these parameters on the basis of a sufficient number of corresponding 2D point observations is referred to as bundle adjustment.

Thomas Hunziker
Lucerne University of Applied Sciences and Arts
Switzerland

Adopting a mean-squared error criterion the bundle adjustment can be formulated as a minimization problem and tackled by the Levenberg-Marquardt algorithm [1], but the non-convex cost function may lead to results that do not match the real scene. For camera pose estimation on the basis of a small number of images there are also direct analytical methods from epipolar geometry [2]. Here a difficulty is to deal with non-ideal cameras (although the cameras in many smart phones and tablets conform quite well to the pinhole camera model) and point deviations in the 2D image planes. Such deviations can easily occur during the identification of point (i.e., feature) correspondences over multiple images.

These difficulties can be dealt with for rather straightforward 3D objects such as buildings. Commercial and free software solutions are available which produce rather accurate 3D models of suitable objects, mostly in off-line processing and in some cases requiring user assistance. As for the reconstruction of more complex 3D objects, such as objects comprised of many different components, only with a sufficient number of images there is a chance to capture all object parts. This motivates the use of image sequences rather than small sets of images from distinct positions. Various methods for tracking 3D features over a series of frames by a monocular camera have been proposed [3]. This topic has attracted interest particularly in robotics – for the creation of maps of unknown environments by autonomous vehicles – under the acronym SLAM (Simultaneous Localization and Mapping) [4], [5].

In this paper we focus on lattice-like objects, that is, complex objects composed of distinct object parts that are interconnected by bars or similar connecting elements. An example would be electricity pylons in the form of steel lattice towers. Our goal is to recognize the object parts, determine their 3D locations, and also the connecting elements such that a 3D model of the complete object can be composed. A motivation behind a machine-aided recognition of such objects could be the detection of faulty or missing elements for safety purposes.

Images of lattice-like objects consist of lines for the most part, and 3D reconstruction based on line segments has been considered in some recent papers [6], [7]. Line segment detection also plays a role in the procedure we propose, however, for the 3D reconstruction we rely exclusively on point features as they have turned out more suitable in the context of bundle adjustment. We represent the camera poses and the feature locations by probability distributions, the former by means of a particle filter (as in [8], for instance) and the latter in the form of Gaussian distributions. With every additional frame the 3D location of an observed feature

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

becomes more confined, which is reflected by covariance matrices with declining determinants.

Rather than dealing with large-scale compositions we use small-scale molecular models as test objects (see Fig. 1 for an example). These are handy while possessing structural similarities with the above mentioned objects. We have implemented the algorithm proposed in this paper on a tablet computer running Android. Moving the tablet around with the camera pointing towards a molecule, a 3D model of the object is composed within just a few seconds and displayed on the screen.

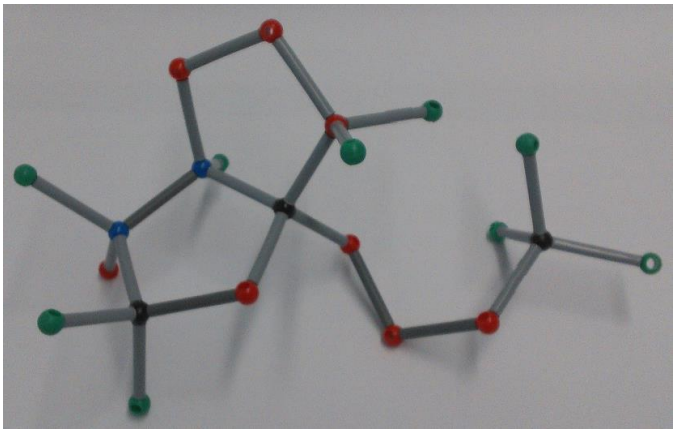

Figure 1. Example of a molecular model.

II. 3D Object Reconstruction

The principal steps of the processing chain are sketched in Fig. 2. The 3D reconstruction relies on point features detected in the 2D images. Line segments are also detected for finding the connecting elements, however, they are incorporated not before the 3D locations of the point features have been estimated. In the final processing step the 3D model is build. In the following the steps shown in the figure are discussed in more detail.

A. *2D Point Feature Detection and Tracking*

A considerable number of point feature detectors and descriptors have been proposed and benchmarked over the past years. We use feature *descriptors* to track identified object parts over consecutive frames, while for the actual feature *detection* we simply employ a circle detector based on the Hough transform because our elementary object components are atoms having the form of spheres. More specifically, a circle detection first delivers the 2D locations of possible atoms in an image, followed by a labelling of these locations by means of descriptors. As is common practice, the descriptors facilitate the matching of detected features over consecutive frames. We have found in a study that in terms of complexity and reliability we are best served by BRIEF [9], one of the descriptors implemented in the popular OpenCV library.

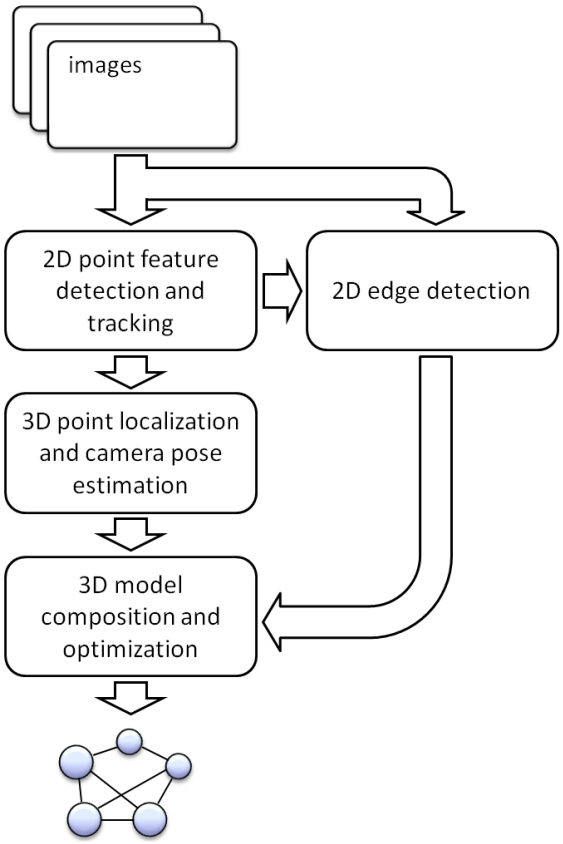

Figure 2. Proposed 3D recognition procedure.

B. *3D Point Location and Camera Pose Estimation*

The point features that have been tracked over multiple frames serve as the basis for the 3D reconstruction, which goes along with an estimation of the camera poses associated with the frames. In the following the estimation of the 3D coordinates of the features is described first, followed by a method for camera pose tracking and a scheme to incorporate the two tasks.

Let $\mathbf{x}_n$ represent the position of the nth observed feature in the world coordinate system. The coordinate transformation from the world coordinate system to the camera coordinate system associated with the mth camera can be expressed as

$$\mathbf{p}_{m,n} = \mathbf{R}_m \mathbf{x}_n + \mathbf{t}_m, \qquad (1)$$

where $\mathbf{R}_m$ and $\mathbf{t}_m$ denote a rotation matrix and a translation vector, respectively, defining the mth camera pose. The camera coordinate system, with the origin coinciding with the center of the perspective projection, may be chosen such that the x- and y-axes are parallel to the corresponding axes in the 2D image plane while the perpendicular z-axis represents the optical axis. Assuming a pinhole camera model, the 2D

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

coordinates in the image plane can then be derived from $\mathbf{p}_{m,n}$ very simply through a division of the *x* and *y* coordinates in $\mathbf{p}_{m,n}$ by the *z* coordinate.

We actually need the inverse mapping, from an observed 2D image point to a 3D point in world coordinates. Making some assumption about the distance between the camera center and the observed feature, possibly in the form of a predefined constant, an estimate $\hat{\mathbf{p}}_{m,n}$ of the 3D feature location in camera coordinates is readily obtained. The estimated coordinates can be transformed to coordinates in the world coordinate system according to

$$\hat{\mathbf{x}}_{m,n} = \mathbf{R}_m^{-1}\left(\hat{\mathbf{p}}_{m,n} - \mathbf{t}_m\right), \qquad (2)$$

the inverse mapping of (1). Clearly, because of the unknown distance from the camera the accuracy of the vector $\hat{\mathbf{x}}_{m,n}$ is poor in the direction $\mathbf{R}_m{}^{-1}\hat{\mathbf{p}}_{m,n}$ in particular, the direction of the beam from the optical center through the observed point in the image plane (represented as dashed lines in Fig. 3). But deviations also occur in the other directions as a result of nonideal camera characteristics and feature tracking. We choose to describe the actual location $\mathbf{x}_n$ of the point statistically, by means of a 3-variate normal distribution

$$\left(\mathbf{x}_n \middle| \hat{\mathbf{x}}_{m,n}\right) = \frac{1}{\sqrt{(2\pi)^3 \det \boldsymbol{\Sigma}_{m,n}}} \times \exp\left(-\frac{1}{2}(\mathbf{x}_n - \hat{\mathbf{x}}_{m,n})^{\mathrm{T}} \boldsymbol{\Sigma}_{m,n}^{-1} (\mathbf{x}_n - \hat{\mathbf{x}}_{m,n})\right) \qquad (3)$$

with $\hat{\mathbf{x}}_{m,n}$ defining the mean and $\boldsymbol{\Sigma}_{m,n}$ the covariance matrix. The latter we define as

$$\boldsymbol{\Sigma}_{m,n} = c_0 \cdot \left(\mathbf{R}_m^{-1}\hat{\mathbf{p}}_{m,n}\right)\left(\mathbf{R}_m^{-1}\hat{\mathbf{p}}_{m,n}\right)^{\mathrm{T}} + c_1 \cdot \mathbf{I}_3 \qquad (4)$$

with c_0 and c_1 two constants and $\mathbf{I}_3$ the 3x3-identity matrix. Areas containing a feature with high probability are visualized in Fig. 3 as ellipsoids. Their semi-major axes coincide with the beams from the camera center through the observed points in the image planes.

Let us now suppose the feature at $\mathbf{x}_n$ is also observed in the *k*th frame. We may regard (3) as the prior distribution of the location $\mathbf{x}_n$. As the Gaussian distribution is self-conjugate with respect to a Gaussian likelihood function $p(\hat{\mathbf{x}}_{k,n}|\mathbf{x}_n)$, the posterior distribution $p(\mathbf{x}_n|\hat{\mathbf{x}}_{k,n}, \hat{\mathbf{x}}_{m,n})$ is also Gaussian. Specifically, the mean and covariance matrix of the posterior distribution are given as

$$\hat{\mathbf{x}}_{\{m,k\},n} = \boldsymbol{\Sigma}_{\{m,k\},n}\left(\boldsymbol{\Sigma}_{m,n}^{-1}\hat{\mathbf{x}}_{m,n} + \boldsymbol{\Sigma}_{k,n}^{-1}\hat{\mathbf{x}}_{k,n}\right) \qquad (5)$$

and

$$\boldsymbol{\Sigma}_{\{m,k\},n} = \left(\boldsymbol{\Sigma}_{m,n}^{-1} + \boldsymbol{\Sigma}_{k,n}^{-1}\right)^{-1}, \qquad (6)$$

respectively. The observations in a third frame and beyond can be incorporated in a similar fashion. The more observations of a feature are available the smaller the determinant of the covariance matrix and thus the more confined the area in which the feature resides with high probability.

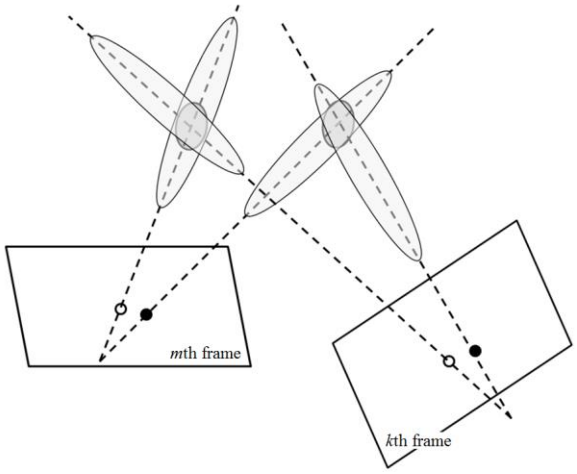

Figure 3. Illustration of the probabilistic point feature localization using Gaussian distributions. For features found in a single image, the ratio of the largest and smallest covariance matrix eigenvalues (corresponding to the ratio of the major and minor radiuses of the ellipsoid) is large. The more images with a certain feature from different angles are available, the smaller the eigenvalue ratio as well as the area in which the feature resides with high probability.

We now turn to the estimation of the camera poses associated with the sequence of frames. The first camera pose can be defined arbitrarily. The choice of the six parameters defining the pose – three coordinates and three Euler angles – establishes the world coordinate system. Subsequently it is the deviation of a camera pose with respect to the preceding cameras that needs to be estimated, with the help of the 2D coordinates of a number of features that have been found in both the actual as well as the preceding frames. On the basis of two images the problem of determining the relative pose along with the 3D locations of a number of object points may actually have multiple solutions, so at the beginning each of the possible solutions needs to be reevaluated with at least a third image. A larger number of frames is actually needed for accurate results in the case of inappropriate object points, deviations in the point observations or insufficient change of pose from frame to frame.

The equation system holding the six parameters that define a pose is nonlinear and difficult to solve in an efficient fashion. However, once the Euler angles are known, the 3D camera position can be analytically derived in a straightforward way. Regarding the pose estimation as an optimization problem, we may thus resort to a numerical method performing a search in the three dimensional space of the Euler angles. Rather than an exhaustive search we choose to carry out a Monte Carlo sampling, embedded in a particle filter. Employing a particle filter seems particulary suitable in view of the above mentioned multitude of possible solutions over several frames as well as the nonlinear dependence on the Euler angles. Each particle represents a specific camera trajectory, covering both the positions and the orientations of the cameras behind the thus far obtained frames. Once an additional frame becomes available, for each particle an eligible camera pose is found through a Monte Carlo

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

sampling. This step involves an evaluation of a larger number of samples on the basis of the reconstructed 3D object points, and in the same time a rejection of outliers in a RANSAC (Random Sample Consensus) fashion. Finally, an appropriate objective function facilitates the choice of the most suitable particle.

C. 2D Edge Detection

To find connecting elements between the point features (i.e., the atoms) we employ an image pre-processing in the form of a Canny edge detection followed by a Hough line transform. Typically a bar linking two atoms is enclosed by two parallel lines, but of course sometimes lines are missing due to insufficient contrast, or they appear as a result of shadows for instance. Moreover, a third atom in front of a bar may be seen as two bars linked to the third atom.

A rule is needed to identify probable connecting elements on the basis of the detected straight lines. The rule we have defined builds on the positions of the line endpoints relative to the atom centers. Additionally, a tentatively detected bar linking two atoms is rejected if a third atom happens to lie on the extended line between the two atoms. This way, false detections can be limited at the cost of a rather moderate detection rate, as can be seen in Fig. 4.

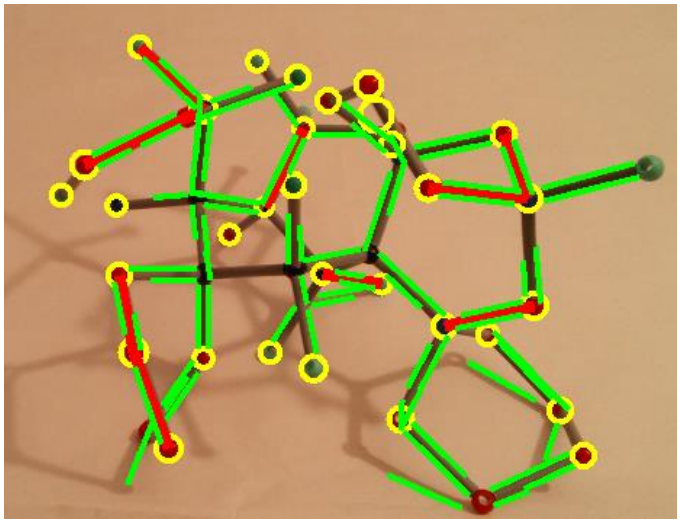

Figure 4. Detected connecting elements (marked by red lines) on the basis of detected point features (yellow circles) and straight lines (green lines).

D. 3D Model Composition and Organization

Missed object elements, false detections due to shadows or reflections, as well as incorrect feature tracking and other flaws result in inaccurate 3D object reconstruction. The defects can be mended with the help of a priori information about the object. A priori information about adjacent object elements, the lengths of certain connecting elements, or the angles between connectors may be available in deterministic or probabilistic form. Using this knowledge possible methods for improving the results from the preceding processing range from simple element elimination to sophisticated optimization procedures based on graphical models.

III. Prototype

The procedure described in Sect. II has been implemented on a tablet computer running Android on a quad-core processor and equipped with a rear camera. The app uses functions from the OpenCV library for the Hough circle and line transforms, the computation of BRIEF descriptors of the detected point features, as well as for the Canny edge detection. These operations are rather time consuming given the camera resolution of 1280 times 960 pixels, so that the frame rate is limited to only about one to three frames per second, depending on the scene complexity.

The 3D reconstruction process and the model optimization process run in parallel to the image pre-processing and feature detection process. The 3D reconstruction is also computationally intensive, in particular the pose estimation with the particle filter behind. For each particle, the computation of a new pose involves a Monte-Carlo sampling of candidate orientations, each of which needs to be evaluated on the basis of the observed features. These evaluations involve large numbers of covariance and rotation matrix inversions, an efficient implementation of which significantly reduces the computation times. Rotation matrices can efficiently be defined by quaternions, one of the concepts we make use of for ending up with a solution that offers on-line 3D model building on customary hardware.

In the prototype the numbers of particles and samples per pose update are adjusted continuously such as to keep up with the delivery of frames. The number of necessary frames for the reconstruction of a certain molecule model depends on the complexity of the latter, ranging from a couple of frames to several hundred frames in the case of complex models composed of 50 atoms or more. Molecules of moderate complexities are perfectly reconstructed under normal circumstances (see Fig. 5 for an example), whereas 3D reconstructions of larger molecules turn out to be incomplete in some cases. It is difficult to precisely assess the detection rates, because they depend on so many factors including camera motion, background, and light. The app further offers the option to display the camera poses along with the reconstructed object (see Fig. 6).

IV. CONCLUSIONS

Creating models of 3D scenes or objects from the images of a monocular camera is a challenging task. For the necessary bundle adjustment a number of methods have been proposed. The prospects depend on the characteristics of the 3D scene, but often the results are not satisfactory because of the non-convexity of the underlying optimization problem or inaccurate observations.

We have focussed on lattice-like objects, which have not yet received much attention in the context of 3D

Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

reconstruction. Having the above mentioned difficulties in mind, we have proposed a new method towards 3D reconstruction, building on well-known concepts from computer vision, such as feature tracking, and Bayesian statistics, such as particle filters. To demonstrate the efficiency of the proposed method, an app has been implemented running on a smart phone or tablet computer, able to compute and render 3D reconstructions of molecular models in an online fashion.

We believe that the described method may also facilitate a 3D modeling of more complex objects of the same type, such as electricity pylons for instance. Numerous applications are conceivable in this context, ranging from fault detection to 3D map generation.

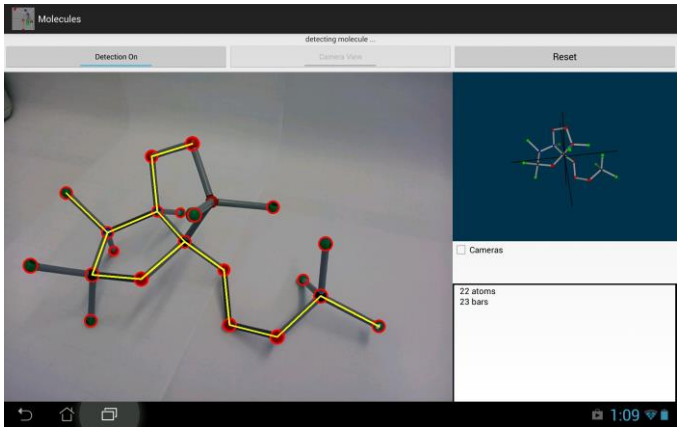

Figure 5. Example of a molecular model reconstruction: the bigger view shows what the camera sees, and the smaller view shows the rendered model using OpenGL.

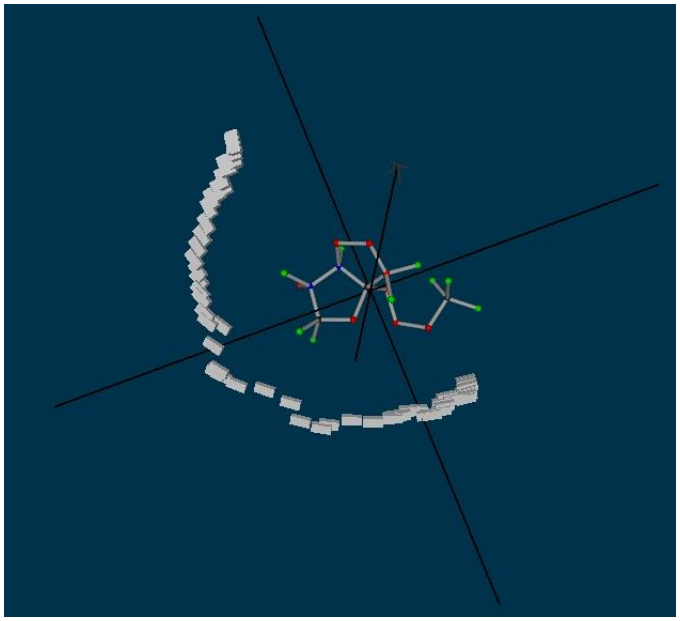

Figure 6. Reconstructed molecule including camera positions and orientations.

References

[1] M. I. A. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," ACM Transactions on Mathematical Software, vol. 36, pp. 2:1–2:30, Mar. 2009.

[2] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge, UK: Cambridge University Press, 2nd ed., 2004.

[3] V. Lepetit and P. Fua, "Monocular model-based 3D tracking of rigid objects: A survey," Foundations and Trends in Computer Graphics and Vision, vol. 1, no. 1, pp. 1–89, 2005.

[4] R. Munguia, B. Castillo-Toledo, and A. Grau, "A robust approach for a filter-based monocular simultaneous localization and mapping (SLAM) system," Sensors, vol. 13, pp. 8501–8522, July 2013.

[5] S. Riisgaard and M. R. Blas, "SLAM for dummies: A tutorial approach to simultaneous localization and mapping," tech. rep., 2005.

[6] M. Hofer, A. Wendel, and H. Bischof, "Incremental line-based 3D reconstruction using geometric constraints," in Proc. British Machine Vision Conference (BMVC 2013), Bristol, UK, Sept. 2013.

[7] A. Jain, C. Kurz, T. Thormählen, and H.-P. Seidel, "Exploiting global connectivity constraints for reconstruction of 3D line segment from images," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), San Francisco, CA, June 2010.

[8] B. Kalyan, K. W. Lee, and W. S. Wijesoma, "FISST-SLAM: Finite set statistical approach to simultaneous localization and mapping," the International Journal of Robotics Research, vol. 29, pp. 1251–1262, Sept. 2010.

[9] M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 7, pp. 1281–1298, 2012.

About Author (s):

Thomas Hunziker received the Ph.D. degree from the Swiss Federal Institute of Technology (ETH) Zurich in 2002. After a couple of years in both Japan (at ATR in Kyoto) and Germany (at Univ. of Kassel) he joined the Lucerne University of Applied Sciences and Arts, Switzerland. Besides of lecturing he is engaged in research projects in the areas of signal processing, computer vision, machine learning and digital communications.