Semi-Dense Visual Odometry for a Monocular Camera*

Jakob Engel, Jürgen Sturm, Daniel Cremers
TU München, Germany

Abstract

We propose a fundamentally novel approach to real-time visual odometry for a monocular camera. It allows to benefit from the simplicity and accuracy of dense tracking – which does not depend on visual features – while running in real-time on a CPU. The key idea is to continuously estimate a semi-dense inverse depth map for the current frame, which in turn is used to track the motion of the camera using dense image alignment. More specifically, we estimate the depth of all pixels which have a non-negligible image gradient. Each estimate is represented as a Gaussian probability distribution over the inverse depth. We propagate this information over time, and update it with new measurements as new images arrive. In terms of tracking accuracy and computational speed, the proposed method compares favorably to both state-of-the-art dense and feature-based visual odometry and SLAM algorithms. As our method runs in real-time on a CPU, it is of large practical value for robotics and augmented reality applications.

1. Towards Dense Monocular Visual Odometry

Tracking a hand-held camera and recovering the three-dimensional structure of the environment in real-time is among the most prominent challenges in computer vision. In the last years, dense approaches to these challenges have become increasingly popular: Instead of operating solely on visual feature positions, they reconstruct and track on the whole image using a surface-based map and thereby are fundamentally different from feature-based approaches. Yet, these methods are to date either not real-time capable on standard CPUs [11, 15, 17] or require direct depth measurements from the sensor [7], making them unsuitable for many practical applications.

In this paper, we propose a novel semi-dense visual odometry approach for a monocular camera, which combines the accuracy and robustness of dense approaches with the efficiency of feature-based methods. Further, it computes highly accurate semi-dense depth maps from the monocular images, providing rich information about the 3D structure of the environment. We use the term visual odometry as supposed to SLAM, as – for simplicity – we deliberately maintain only information about the currently visible scene, instead of building a global world-model.

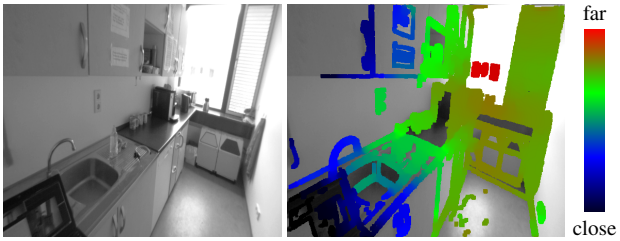

Figure 1. Semi-Dense Monocular Visual Odometry: Our approach works on a semi-dense inverse depth map and combines the accuracy and robustness of dense visual SLAM methods with the efficiency of feature-based techniques. Left: video frame, Right: color-coded semi-dense depth map, which consists of depth estimates in all image regions with sufficient structure.

1.1. Related Work

Feature-based monocular SLAM. In all feature-based methods (such as [4, 8]), tracking and mapping consists of two separate steps: First, discrete feature observations (i.e., their locations in the image) are extracted and matched to each other. Second, the camera and the full feature poses are calculated from a set of such observations – disregarding the images themselves. While this preliminary abstraction step greatly reduces the complexity of the overall problem and allows it to be tackled in real time, it inherently comes with two significant drawbacks: First, only image information conforming to the respective feature type and parametrization – typically image corners and blobs [6] or line segments [9] – is utilized. Second, features have to be matched to each other, which often requires the costly computation of scale- and rotation-invariant descriptors and robust outlier estimation methods like RANSAC.

Dense monocular SLAM. To overcome these limitations and to better exploit the available image information, dense monocular SLAM methods [11, 17] have recently been proposed. The fundamental difference to keypoint-based approaches is that these methods directly work on the images

* This work was supported by the ERC Starting Grant ConvexVision and the DFG project Mapping on Demand

instead of a set of extracted features, for both mapping and tracking: The world is modeled as dense surface while in turn new frames are tracked using whole-image alignment. This concept removes the need for discrete features, and allows to exploit all information present in the image, increasing tracking accuracy and robustness. To date however, doing this in real-time is only possible using modern, powerful GPU processors.

Similar methods are broadly used in combination with RGB-D cameras [7], which directly measure the depth of each pixel, or stereo camera rigs [3] – greatly reducing the complexity of the problem.

Dense multi-view stereo. Significant prior work exists on multi-view dense reconstruction, both in a real-time setting [13, 11, 15], as well as off-line [5, 14]. In particular for off-line reconstruction, there is a long history of using different baselines to steer the stereo-inherent trade-off between accuracy and precision [12]. Most similar to our approach is the early work of Matthies et al., who proposed probabilistic depth map fusion and propagation for image sequences [10], however only for structure from motion, i.e., not coupled with subsequent dense tracking.

1.2. Contributions

In this paper, we propose a novel semi-dense approach to monocular visual odometry, which does not require feature points. The key concepts are

- a probabilistic depth map representation,
- tracking based on whole-image alignment,
- the reduction on image-regions which carry information (semi-dense), and
- the full incorporation of stereo measurement uncertainty.

To the best of our knowledge, this is the first featureless, real-time monocular visual odometry approach, which runs in real-time on a CPU.

1.3. Method Outline

Our approach is partially motivated by the basic principle that for most real-time applications, video information is abundant and cheap to come by. Therefore, the computational budget should be spent such that the expected information gain is maximized. Instead of reducing the images to a sparse set of feature observations however, our method continuously estimates a *semi-dense inverse depth map* for the current frame, i.e., a dense depth map covering all image regions with non-negligible gradient (see Fig. 2). It is comprised of one inverse depth hypothesis per pixel modeled by a Gaussian probability distribution. This representation still allows to use whole-image alignment [7] to track new

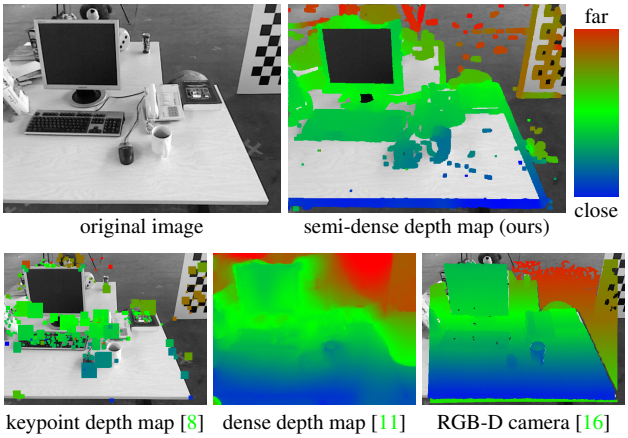

Figure 2. Semi-Dense Approach: Our approach reconstructs and tracks on a *semi-dense inverse depth map*, which is dense in all image regions carrying information (top-right). For comparison, the bottom row shows the respective result from a keypoint-based approach, a fully dense approach and the ground truth from an RGB-D camera.

frames, while at the same time greatly reducing computational complexity compared to volumetric methods. The estimated depth map is propagated from frame to frame, and updated with variable-baseline stereo comparisons. We explicitly use prior knowledge about a pixel's depth to select a suitable reference frame on a per-pixel basis, and to limit the disparity search range.

The remainder of this paper is organized as follows: Section 2 describes the semi-dense mapping part of the proposed method, including the derivation of the observation accuracy as well as the probabilistic data fusion, propagation and regularization steps. Section 3 describes how new frames are tracked using whole-image alignment, and Sec. 4 summarizes the complete visual odometry method. A qualitative as well as a quantitative evaluation is presented in Sec. 5. We then give a brief conclusion in Sec. 6.

2. Semi-Dense Depth Map Estimation

One of the key ideas proposed in this paper is to estimate a semi-dense inverse depth map for the current camera image, which in turn can be used for estimating the camera pose of the next frame. This depth map is continuously propagated from frame to frame, and refined with new stereo depth measurements, which are obtained by performing per-pixel, adaptive-baseline stereo comparisons. This allows us to accurately estimate the depth both of close-by and far-away image regions. In contrast to previous work that accumulates the photometric cost over a sequence of several frames [11, 15], we keep exactly one inverse depth hypothesis per pixel that we represent as Gaussian probability distribution.

This section is comprised of three main parts: Sec-

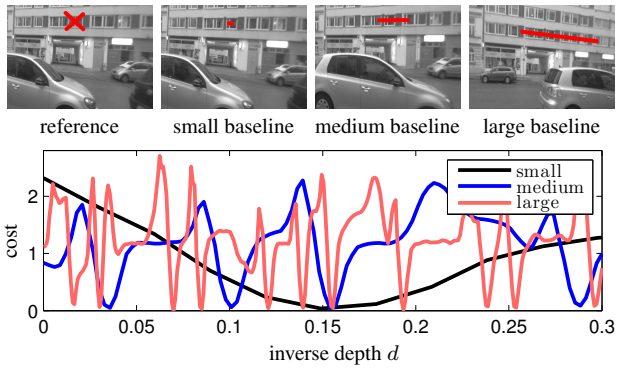

Figure 3. Variable Baseline Stereo: Reference image (left), three stereo images at different baselines (right), and the respective matching cost functions. While a small baseline (black) gives a unique, but imprecise minimum, a large baseline (red) allows for a very precise estimate, but has many false minima.

tion 2.1 describes the stereo method used to extract new depth measurements from previous frames, and how they are incorporated into the prior depth map. In Sec. 2.2, we describe how the depth map is propagated from frame to frame. In Sec. 2.3, we detail how we partially regularize the obtained depth map in each iteration, and how outliers are handled. Throughout this section, d denotes the *inverse* depth of a pixel.

2.1. Stereo-Based Depth Map Update

It is well known [12] that for stereo, there is a trade-off between precision and accuracy (see Fig. 3). While many multiple-baseline stereo approaches resolve this by accumulating the respective cost functions over many frames [5, 13], we propose a probabilistic approach which explicitly takes advantage of the fact that in a video, small-baseline frames are available before large-baseline frames.

The full depth map update (performed once for each new frame) consists of the following steps: First, a subset of pixels is selected for which the accuracy of a disparity search is sufficiently large. For this we use three intuitive and very efficiently computable criteria, which will be derived in Sec. 2.1.3. For each selected pixel, we then individually select a suitable reference frame, and perform a one-dimensional disparity search. Propagated prior knowledge is used to reduce the disparity search range when possible, decreasing computational cost and eliminating false minima. The obtained inverse depth estimate is then fused into the depth map.

2.1.1 Reference Frame Selection

Ideally, the reference frame is chosen such that it maximizes the stereo accuracy, while keeping the disparity search range as well as the observation angle sufficiently small. As the stereo accuracy depends on many factors and because this selection is done for each pixel independently, we employ the following heuristic: We use the oldest frame the pixel was observed in, where the disparity search range and the observation angle do not exceed a certain threshold (see Fig. 4). If a disparity search is unsuccessful (i.e., no good match is found), the pixel's "age" is increased, such that subsequent disparity searches use newer frames where the pixel is likely to be still visible.

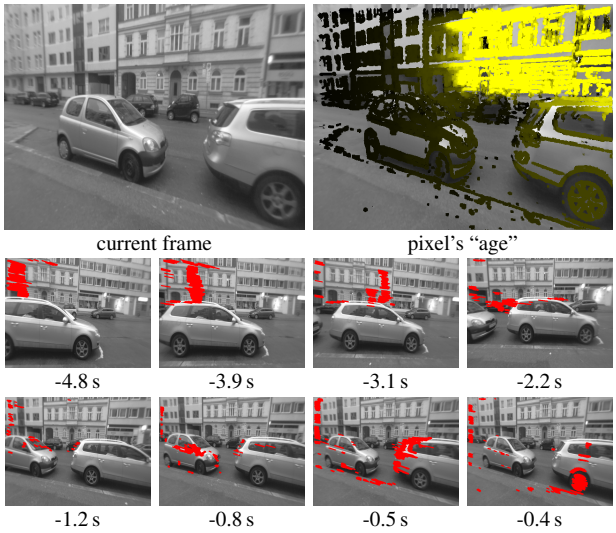

Figure 4. Adaptive Baseline Selection: For each pixel in the new frame (top left), a different stereo-reference frame is selected, based on how long the pixel was visible (top right: the more yellow, the older the pixel.). Some of the reference frames are displayed below, the red regions were used for stereo comparisons.

2.1.2 Stereo Matching Method

We perform an exhaustive search for the pixel's intensity along the epipolar line in the selected reference frame, and then perform a sub-pixel accurate localization of the matching disparity. If a prior inverse depth hypothesis is available, the search interval is limited by $d \pm 2\sigma_d$, where d and σ_d denote the mean and standard deviation of the prior hypothesis. Otherwise, the full disparity range is searched.

In our implementation, we use the SSD error over five equidistant points on the epipolar line: While this significantly increases robustness in high-frequent image regions, it does not change the purely one-dimensional nature of this search. Furthermore, it is computationally efficient, as 4 out of 5 interpolated image values can be re-used for each SSD evaluation.

2.1.3 Uncertainty Estimation

In this section, we use uncertainty propagation to derive an expression for the error variance σ_d^2 on the inverse depth d.

In general this can be done by expressing the optimal inverse depth d^* as a function of the noisy inputs – here we consider the images I_0, I_1 themselves, their relative orientation ξ and the camera calibration in terms of a projection function π[1]

$$d^* = d(I_0, I_1, \xi, \pi). \tag{1}$$

The error-variance of d^* is then given by

$$\sigma_d^2 = J_d \Sigma J_d^T, \tag{2}$$

where J_d is the Jacobian of d, and Σ the covariance of the input-error. For more details on covariance propagation, including the derivation of this formula, we refer to [2]. For simplicity, the following analysis is performed for patch-free stereo, i.e., we consider only a point-wise search for a *single intensity value* along the epipolar line.

For this analysis, we split the computation into three steps: First, the epipolar line in the reference frame is computed. Second, the best matching position $\lambda^* \in \mathbb{R}$ along it (i.e., the disparity) is determined. Third, the inverse depth d^* is computed from the disparity λ^*. The first two steps involve two independent error sources: the geometric error, which originates from noise on ξ and π and affects the first step, and the photometric error, which originates from noise in the images I_0, I_1 and affects the second step. The third step scales these errors by a factor, which depends on the baseline.

Geometric disparity error. The geometric error is the error ϵ_λ on the disparity λ^* caused by noise on ξ and π. While it would be possible to model, propagate, and estimate the complete covariance on ξ and π, we found that the gain in accuracy does not justify the increase in computational complexity. We therefore use an intuitive approximation: Let the considered epipolar line segment $L \subset \mathbb{R}^2$ be defined by

$$L := \left\{ l_0 + \lambda \begin{pmatrix} l_x \\ l_y \end{pmatrix} \mid \lambda \in S \right\}, \tag{3}$$

where λ is the disparity with search interval S, $(l_x, l_y)^T$ the *normalized* epipolar line direction and l_0 the point corresponding to infinite depth. We now assume that only the absolute position of this line segment, i.e., l_0 is subject to isotropic Gaussian noise ϵ_l. As in practice we keep the searched epipolar line segments short, the influence of rotational error is small, making this a good approximation.

Intuitively, a positioning error ϵ_l on the epipolar line causes a small disparity error ϵ_λ if the epipolar line is parallel to the image gradient, and a large one otherwise (see Fig. 5). This can be mathematically derived as follows: The image constrains the optimal disparity λ^* to lie on a certain isocurve, i.e. a curve of equal intensity. We approximate

[1] In the linear case, this is the camera matrix K – in practice however, nonlinear distortion and other (unmodeled) effects also play a role.

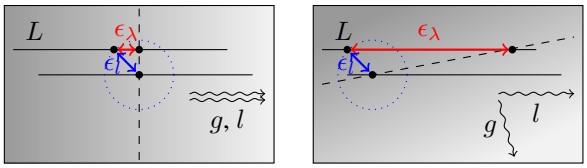

Figure 5. Geometric Disparity Error: Influence of a small positioning error ϵ_l of the epipolar line on the disparity error ϵ_λ. The dashed line represents the isocurve on which the matching point has to lie. ϵ_λ is small if the epipolar line is parallel to the image gradient (left), and a large otherwise (right).

this isocurve to be locally linear, i.e. the gradient direction to be locally constant. This gives

$$l_0 + \lambda^* \begin{pmatrix} l_x \\ l_y \end{pmatrix} \overset{!}{=} g_0 + \gamma \begin{pmatrix} -g_y \\ g_x \end{pmatrix}, \qquad \gamma \in \mathbb{R} \tag{4}$$

where $g := (g_x, g_y)$ is the image gradient and g_0 a point on the isoline. The influence of noise on the image values will be derived in the next paragraph, hence at this point g and g_0 are assumed noise-free. Solving for λ gives the optimal disparity λ^* in terms of the noisy input l_0:

$$\lambda^*(l_0) = \frac{\langle g, g_0 - l_0 \rangle}{\langle g, l \rangle} \tag{5}$$

Analogously to (2), the variance of the geometric disparity error can then be expressed as

$$\sigma^2_{\lambda(\xi,\pi)} = J_{\lambda^*(l_0)} \begin{pmatrix} \sigma_l^2 & 0 \\ 0 & \sigma_l^2 \end{pmatrix} J^T_{\lambda^*(l_0)} = \frac{\sigma_l^2}{\langle g, l \rangle^2}, \tag{6}$$

where g is the *normalized* image gradient, l the *normalized* epipolar line direction and σ_l^2 the variance of ϵ_l. Note that this error term solely originates from noise on the relative camera orientation ξ and the camera calibration π, i.e., it is independent of image intensity noise.

Photometric disparity error. Intuitively, this error encodes that small image intensity errors have a large effect on the estimated disparity if the image gradient is small, and a small effect otherwise (see Fig. 6). Mathematically, this relation can be derived as follows. We seek the disparity λ^* that minimizes the difference in intensities, i.e.,

$$\lambda^* = \min_\lambda \left(i_{\text{ref}} - I_p(\lambda)\right)^2, \tag{7}$$

where i_{ref} is the reference intensity, and $I_p(\lambda)$ the image intensity on the epipolar line at disparity λ. We assume a good initialization λ_0 to be available from the exhaustive search. Using a first-order Taylor approximation for I_p gives

$$\lambda^*(I) = \lambda_0 + \left(i_{\text{ref}} - I_p(\lambda_0)\right) g_p^{-1}, \tag{8}$$

where g_p is the gradient of I_p, that is image gradient along the epipolar line. For clarity we only consider noise on i_{ref} and $I_p(\lambda_0)$; equivalent results are obtained in the general case when taking into account noise on the image values involved in the computation of g_p. The variance of the pho-

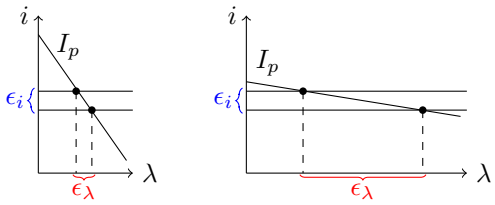

Figure 6. Photometric Disparity Error: Noise ϵ_i on the image intensity values causes a small disparity error ϵ_λ if the image gradient along the epipolar line is large (left). If the gradient is small, the disparity error is magnified (right).

tometric disparity error is given by

$$\sigma^2_{\lambda(I)} = J_{\lambda^*(I)} \begin{pmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{pmatrix} J_{\lambda^*(I)} = \frac{2\sigma_i^2}{g_p^2}, \tag{9}$$

where σ_i^2 is the variance of the image intensity noise. The respective error originates solely from noisy image intensity values, and hence is independent of the geometric disparity error.

Pixel to inverse depth conversion. Using that, for small camera rotation, the inverse depth d is approximately proportional to the disparity λ, the observation variance of the inverse depth $\sigma^2_{d,\text{obs}}$ can be calculated using

$$\sigma^2_{d,\text{obs}} = \alpha^2 \left(\sigma^2_{\lambda(\xi,\pi)} + \sigma^2_{\lambda(I)} \right), \tag{10}$$

where the proportionality constant α – in the general, non-rectified case – is different for each pixel, and can be calculated from

$$\alpha := \frac{\delta_d}{\delta_\lambda}, \tag{11}$$

where δ_d is the length of the searched inverse depth interval, and δ_λ the length of the searched epipolar line segment. While α is inversely linear in the length of the camera translation, it also depends on the translation direction and the pixel's location in the image.

When using an SSD error over multiple points along the epipolar line – as our implementation does – a good upper bound for the matching uncertainty is then given by

$$\sigma^2_{d,\text{obs-SSD}} \leq \alpha^2 \left(\min\{\sigma^2_{\lambda(\xi,\pi)}\} + \min\{\sigma^2_{\lambda(I)}\} \right), \tag{12}$$

where the $\min$ goes over all points included in the SSD error.

2.1.4 Depth Observation Fusion

After a depth observation for a pixel in the current image has been obtained, we integrate it into the depth map as follows: If no prior hypothesis for a pixel exists, we initialize it directly with the observation. Otherwise, the new observation is incorporated into the prior, i.e., the two distributions are multiplied (corresponding to the update step in a Kalman filter): Given a prior distribution $\mathcal{N}(d_p, \sigma_p^2)$ and a noisy observation $\mathcal{N}(d_o, \sigma_o^2)$, the posterior is given by

$$\mathcal{N}\left(\frac{\sigma_p^2 d_o + \sigma_o^2 d_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \sigma_o^2}{\sigma_p^2 + \sigma_o^2} \right). \tag{13}$$

2.1.5 Summary of Uncertainty-Aware Stereo

New stereo observations are obtained on a per-pixel basis, adaptively selecting for each pixel a suitable reference frame and performing a one-dimensional search along the epipolar line. We identified the three major factors which determine the accuracy of such a stereo observation, i.e.,

- the **photometric disparity error** $\sigma^2_{\lambda(\xi,\pi)}$, depending on the *magnitude* of the image gradient along the epipolar line,
- the **geometric disparity error** $\sigma^2_{\lambda(I)}$, depending on the *angle* between the image gradient and the epipolar line (independent of the gradient magnitude), and
- the **pixel to inverse depth ratio** α, depending on the camera translation, the focal length and the pixel's position.

These three simple-to-compute and purely local criteria are used to determine for which pixel a stereo update is worth the computational cost. Further, the computed observation variance is then used to integrate the new measurements into the existing depth map.

2.2. Depth Map Propagation

We continuously propagate the estimated inverse depth map from frame to frame, once the camera position of the next frame has been estimated. Based on the inverse depth estimate d_0 for a pixel, the corresponding 3D point is calculated and projected into the new frame, providing an inverse depth estimate d_1 in the new frame. The hypothesis is then assigned to the closest integer pixel position – to eliminate discretization errors, the sub-pixel accurate image location of the projected point is kept, and re-used for the next propagation step.

For propagating the inverse depth variance, we assume the camera rotation to be small. The new inverse depth d_1 can then be approximated by

$$d_1(d_0) = (d_0^{-1} - t_z)^{-1}, \tag{14}$$

where t_z is the camera translation along the optical axis. The variance of d_1 is hence given by

$$\sigma^2_{d_1} = J_{d_1} \sigma^2_{d_0} J^T_{d_1} + \sigma_p^2 = \left(\frac{d_1}{d_0} \right)^4 \sigma^2_{d_0} + \sigma_p^2, \tag{15}$$

where σ_p^2 is the prediction uncertainty, which directly corresponds to the prediction step in an extended Kalman filter. It can also be interpreted as keeping the variance on

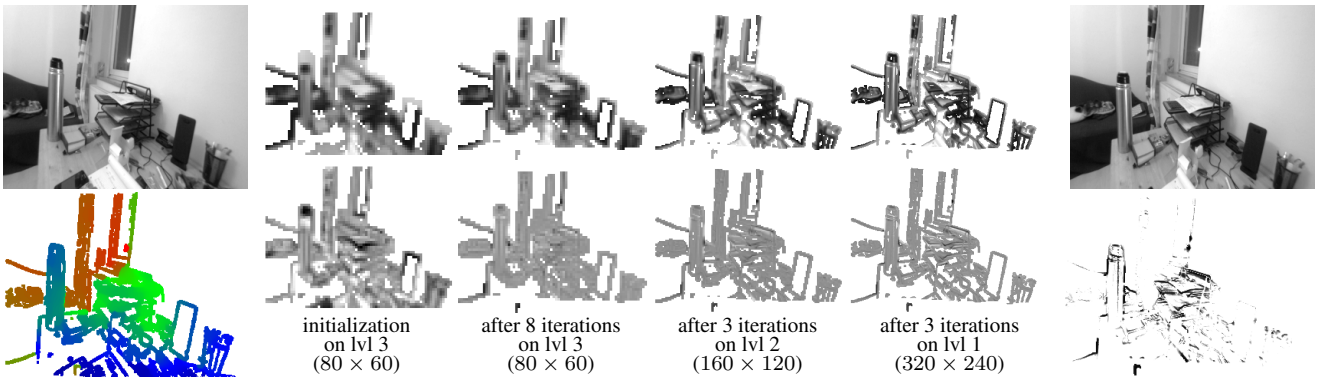

Figure 7. Dense Tracking: Reference image $I_1(\mathbf{x})$ (top left) with associated semi-dense inverse depth map (bottom left). The image in the top right shows the new frame $I_2(\mathbf{x})$ without depth information. Middle: Intermediate steps while minimizing $E(\xi)$ on different pyramid levels. The top row shows the back-warped new frame $I_2(w(\mathbf{x}, d, \xi))$, the bottom row shows the respective residual image $I_2(w(\mathbf{x}, d_i, \xi)) - I_1(\mathbf{x})$. The bottom right image shows the final pixel-weights (black = small weight). Small weights mainly correspond to newly occluded or disoccluded pixel.

the z-coordinate of a point fixed, i.e., setting $\sigma^2_{z_0} = \sigma^2_{z_1}$. We found that using small values for σ^2_p decreases drift, as it causes the estimated geometry to gradually "lock" into place.

Collision handling. At all times, we allow at most one inverse depth hypothesis per pixel: If two inverse depth hypothesis are propagated to the same pixel in the new frame, we distinguish between two cases:

1. if they are statistically similar, i.e., lie within 2σ bounds, they are treated as two independent observations of the pixel's depth and fused according to (13).
2. otherwise, the point that is further away from the camera is assumed to be occluded, and is removed.

2.3. Depth Map Regularization

For each frame – after all observations have been incorporated – we perform one regularization iteration by assigning each inverse depth value the average of the surrounding inverse depths, weighted by their respective inverse variance. To preserve sharp edges, if two adjacent inverse depth values are statistically different, i.e., are further away than 2σ, they do not contribute to one another. Note that the respective variances are not changed during regularization to account for the high correlation between neighboring hypotheses. Instead we use the minimal variance of all neighboring pixel when defining the stereo search range, and as a weighting factor for tracking (see Sec. 3).

Outlier removal. To handle outliers, we continuously keep track of the *validity* of each inverse depth hypothesis in terms of the probability that it is an outlier, or has become invalid (e.g., due to occlusion or a moving object). For each successful stereo observation, this probability is decreased. It is increased for each failed stereo search, if the respective intensity changes significantly on propagation, or when the absolute image gradient falls below a given threshold.

If, during regularization, the probability that all contributing neighbors are outliers – i.e., the product of their individual outlier-probabilities – rises above a given threshold, the hypothesis is removed. Equally, if for an "empty" pixel this product drops below a given threshold, a new hypothesis is created from the neighbors. This fills holes arising from the forward-warping nature of the propagation step, and dilates the semi-dense depth map to a small neighborhood around sharp image intensity edges, which significantly increases tracking and mapping robustness.

3. Dense Tracking

Based on the inverse depth map of the previous frame, we estimate the camera pose of the current frame using dense image alignment. Such methods have previously been applied successfully (in real-time on a CPU) for tracking RGB-D cameras [7], which directly provide dense depth measurements along with the color image. It is based on the direct minimization of the photometric error

$$r_i(\xi) := \left(I_2(w(\mathbf{x}_i, d_i, \xi)) - I_1(\mathbf{x}_i)\right)^2, \qquad (16)$$

where the warp function $w\colon \Omega_1 \times \mathbb{R} \times \mathbb{R}^6 \to \Omega_2$ maps each point $\mathbf{x}_i \in \Omega_1$ in the reference image I_1 to the respective point $w(\mathbf{x}_i, d_i, \xi) \in \Omega_2$ in the new image I_2. As input it requires the 3D pose of the camera $\xi \in \mathbb{R}^6$ and uses the estimated inverse depth $d_i \in \mathbb{R}$ for the pixel in I_1. Note that no depth information with respect to I_2 is required.

To increase robustness to self-occlusion and moving objects, we apply a weighting scheme as proposed in [7]. Further, we add the variance of the inverse depth $\sigma^2_{d_i}$ as an additional weighting term, making the tracking resistant to recently initialized and still inaccurate depth estimates from

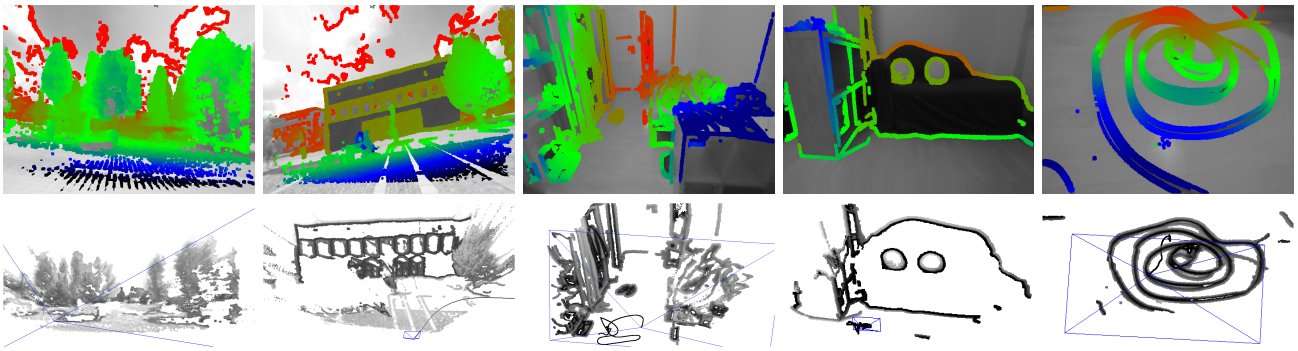

Figure 8. Examples: Top: Camera images overlaid with the respective estimated semi-dense inverse depth map. Bottom: 3D view of tracked scene. Note the versatility of our approach: It accurately reconstructs and tracks through (outside) scenes with a large depth-variance, including far-away objects like clouds , as well as (indoor) scenes with little structure and close to no image corners / keypoints. More examples are shown in the attached video.

the mapping process. The final energy that is minimized is hence given by

$$E(\xi) := \sum_i \frac{\alpha(r_i(\xi))}{\sigma^2_{d_i}} r_i(\xi), \tag{17}$$

where $\alpha\colon \mathbb{R} \to \mathbb{R}$ defines the weight for a given residual. Minimizing this error can be interpreted as computing the maximum likelihood estimator for ξ, assuming independent noise on the image intensity values. The resulting weighted least-squares problem is solved efficiently using an iteratively reweighted Gauss-Newton algorithm coupled with a coarse-to-fine approach, using four pyramid levels. Figure 7 shows an example of the tracking process. For further details on the minimization we refer to [1].

4. System Overview

Tracking and depth estimation is split into two separate threads: One continuously propagates the inverse depth map to the most recent tracked frame, updates it with stereo-comparisons and partially regularizes it. The other simultaneously tracks each incoming frame on the most recent available depth map. While tracking is performed in real-time at 30Hz, one complete mapping iteration takes longer and is hence done at roughly 15Hz – if the map is heavily populated, we adaptively reduce the number of stereo comparisons to maintain a constant frame-rate. For stereo observations, a buffer of up to 100 past frames is kept, automatically removing those that are used least.

We use a standard, keypoint-based method to obtain the relative camera pose between two initial frames, which are then used to initialize the inverse depth map needed for tracking successive frames. From this point onward, our method is entirely self-contained. In preliminary experiments, we found that in most cases our approach is even able to recover from random or extremely inaccurate initial depth maps, indicating that the keypoint-based initialization might become superfluous in the future.

Table 1. Results on RGB-D Benchmark

	position drift ($^{cm}/_{s}$)			**rotation drift** ($^{deg}/_{s}$)		
	ours	[7]	[8]	ours	[7]	[8]
fr2/xyz	0.6	0.6	8.2	0.33	0.34	3.27
fr2/desk	2.1	2.0	-	0.65	0.70	-

5. Results

We have tested our approach on both publicly available benchmark sequences, as well as live, using a hand-held camera. Some examples are shown in Fig. 8. Note that our method does not attempt to build a global map, i.e., once a point leaves the field of view of the camera or becomes occluded, the respective depth value is deleted. All experiments are performed on a standard consumer laptop with Intel i7 quad-core CPU. In a preprocessing step, we rectify all images such that a pinhole camera-model can be applied.

5.1. RGB-D Benchmark Sequences

As basis for a quantitative evaluation and to facilitate reproducibility and easy comparison with other methods, we use the TUM RGB-D benchmark [16]. For tracking and mapping we only use the gray-scale images; for the very first frame however the provided depth image is used as initialization.

Our method (like any monocular visual odometry method) fails in case of pure camera rotation, as the depth of new regions cannot be determined. The achieved tracking accuracy for two feasible sequences – that is, sequences which do not contain strong camera rotation without simultaneous translation – is given in Table 1. For comparison we also list the accuracy from (1) a state-of-the-art, dense RGB-D odometry [7], and (2) a state-of-the-art, keypoint-based monocular SLAM system (PTAM, [8]). We initialize PTAM using the built-in stereo initializer, and perform a 7DoF (rigid body plus scale) alignment to the ground truth trajectory. Figure 9 shows the tracked camera trajectory for fr2/desk. We found that our method achieves similar accu-

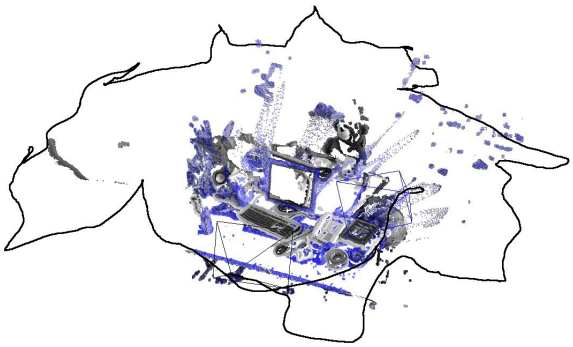

Figure 9. RGB-D Benchmark Sequence fr2/desk: Tracked camera trajectory (black), the depth map of the first frame (blue), and the estimated depth map (gray-scale) after a complete loop around the table. Note how well certain details such as the keyboard and the monitor align.

racy as [7] which uses the same dense tracking algorithm but relies on the Kinect depth images. The keypoint-based approach [8] proves to be significantly less accurate and robust; it consistently failed after a few seconds for the second sequence.

5.2. Additional Test Sequences

To analyze our approach in more detail, we recorded additional challenging sequences with the corresponding ground truth trajectory in a motion capture studio. Figure 10 shows an extract from the video, as well as the tracked and the ground-truth camera position over time. As can be seen from the figure, our approach is able to maintain a reasonably dense depth map at all times and the estimated camera trajectory matches closely the ground truth.

6. Conclusion

In this paper we proposed a novel visual odometry method for a monocular camera, which does not require discrete features. In contrast to previous work on dense tracking and mapping, our approach is based on probabilistic depth map estimation and fusion over time. Depth measurements are obtained from patch-free stereo matching in different reference frames at a suitable baseline, which are selected on a per-pixel basis. To our knowledge, this is the first featureless monocular visual odometry method which runs in real-time on a CPU. In our experiments, we showed that the tracking performance of our approach is comparable to that of fully dense methods without requiring a depth sensor.

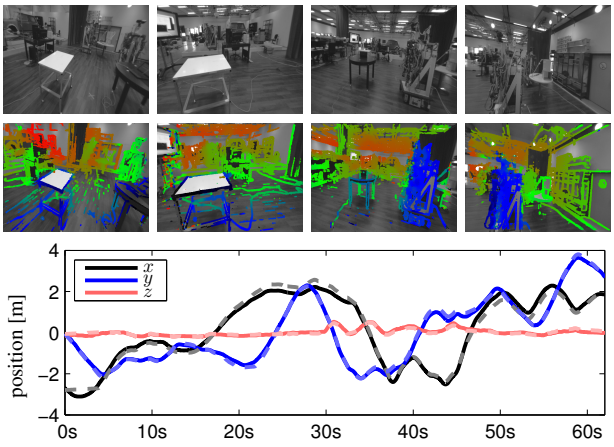

Figure 10. Additional Sequence: Estimated camera trajectory and ground truth (dashed) for a long and challenging sequence. The complete sequence is shown in the attached video.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. Technical report, Carnegie Mellon Univ., 2002. 7

[2] A. Clifford. *Multivariate Error Analysis*. John Wiley & Sons, 1973. 4

[3] A. Comport, E. Malis, and P. Rives. Accurate quadri-focal tracking for robust 3d visual odometry. In *ICRA*, 2007. 2

[4] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 2007. 1

[5] D. Gallup, J. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *CVPR*, 2008. 2, 3

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 1

[7] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *ICRA*, 2013. 1, 2, 6, 7, 8

[8] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality (ISMAR)*, 2007. 1, 2, 7, 8

[9] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *ECCV*, 2008. 1

[10] M. Pollefes et al. Detailed real-time urban 3d reconstruction from video. *IJCV*, 78(2-3):143–167, 2008. 2, 3

[11] L. Matthies, R. Szeliski, and T. Kanade. Incremental estimation of dense depth maps from image image sequences. In *CVPR*, 1988. 2

[12] R. Newcombe, S. Lovegrove, and A. Davison. DTAM: Dense tracking and mapping in real-time. In *ICCV*, 2011. 1, 2

[13] M. Okutomi and T. Kanade. A multiple-baseline stereo. *Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 15(4):353–363, 1993. 2, 3

[14] T. Sato, M. Kanbara, N. Yokoya, and H. Takemura. Dense 3-d reconstruction of an outdoor scene by hundreds-baseline stereo using a hand-held camera. *IJCV*, 47:1–3, 2002. 2

[15] J. Stuehmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition (DAGM)*, 2010. 1, 2

[16] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Intelligent Robot Systems (IROS)*, 2012. 2, 7

[17] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *ECCV*, 2012. 1