Good Edgels To Track: Beating the Aperture Problem with Epipolar Geometry

Tommaso Piccini [1], Mikael Persson[1], Klas Nordberg[1], Michael Felsberg[1] and Rudolf Mester[1 2]

[1]CVL, ISY, Linköping University
[2] VSI Lab, C.S. Dept. Goethe University, Frankfurt

Abstract. An open issue in multiple view geometry and structure from motion, applied to real life scenarios, is the sparsity of the matched key-points and of the reconstructed point cloud. We present an approach that can significantly improve the density of measured displacement vectors in a sparse matching or tracking setting, exploiting the partial information of the motion field provided by linear oriented image patches (edgels). Our approach assumes that the epipolar geometry of an image pair already has been computed, either in an earlier feature-based matching step, or by a robustified differential tracker. We exploit key-points of a lower order, *edgels*, which cannot provide a unique 2D matching, but can be employed if a constraint on the motion is already given. We present a method to extract edgels, which can be effectively tracked given a known camera motion scenario, and show how a constrained version of the Lucas-Kanade tracking procedure can efficiently exploit epipolar geometry to reduce the classical KLT optimization to a 1D search problem. The potential of the proposed methods is shown by experiments performed on real driving sequences. [1]

1 Introduction

Most methods for finding image-to-image correspondences by tracking or matching are applied on selected *key-points*, i.e., image locations which are unique in appearance and can easily be re-identified. While providing a coarse outline of the motion structure in a scene, a drawback of this approach is that the feature points are distributed sparsely, and cannot produce a dense 3D point cloud.

A natural next step is to consider a scenario where the relative motion between the camera and the scene/object already has been determined with high accuracy, e.g., by means of sparse feature matching and windowed bundle adjustment, and is represented by the epipolar structure. After this step, it is possible to increase the density of the motion field, aiming for a dense 3D reconstruction. This can be done by stereo matching on rectified image pairs, and gives

[1] **Acknowledgments -** This work has been funded by the Swedish Excellence Center at Linköping - Lund in Information Technology (ELLIIT)

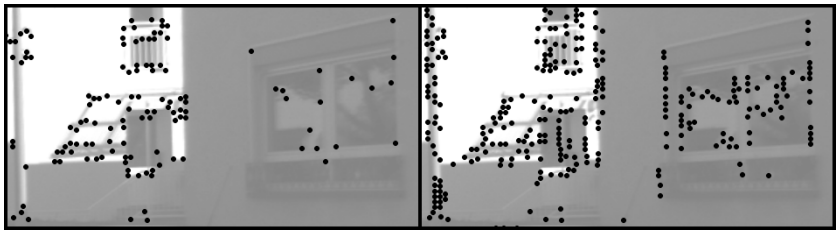

Fig. 1. Close up of an image patch showing a situation where the *GETT* keypoints strongly outnumber *GFTT*. **Left:** GFTT. **Right:** GETT

generally very good results in the case of sideways camera motion and relatively small rotations, but it is not a feasible approach for a general motion, such as forward/backward camera motion combined with significant rotations.

As an alternative, we instead extract edge pixels that are likely to track well given a known epipolar geometry, and perform a local search for the correspondence only on the physically plausible segment of the epipolar line. To do this, we modify the Kanade-Lucas-Tomasi (**KLT**) tracker to only operate along the epipolar line, thus allowing to also track several edge points (Fig. 1). Relevant applications of this method include:

- densification of the partial reconstruction for object detection in the context of autonomous robots.
- increase of the density of the reconstructed 3D point cloud as a final step of a multi-view structure from motion.

To summarize, this paper presents two novel contributions:

- We extract edgels that are likely to track well under the given relative motion. These edge points are called *Good Edgels To Track* (**GETT**).
- We densify the motion field by tracking these new edgel features, using a modified version of the KLT tracker that can deal with the aperture problem, given that the epipolar geometry is known.

2 Related Work

Many applications in computer vision, e.g., structure from motion and visual odometry, have as an initial step the computation of optical flow. This can be done in two ways: computing a global optical flow for the whole image plane [13, 7] or by sparsely computing the displacement of a limited number of (feature) points in the image. The first approach provides a dense flow field, but has a high computational complexity. The sparse approach is more computationally efficient, but the sparsity of the motion field can be a limiting factor when a dense reconstruction is desired. The sparsity is a consequence of applying these methods only on selected feature points, with a structure in their surroundings that allows them to be re-identified in subsequent images [10, 22, 24, 1]. Such

feature points can be found on a second image, depicting a slightly modified version of the same scene, either by matching or by tracking procedures. Matching procedures extract feature points in both images, and try to match the most similar pairs using a specifically designed descriptor [6, 17, 21, 3, 18]. Tracking is instead implemented generally as a search, in the second image, for a point that minimizes a cost function which measures the difference between small patches of the image surrounding the two points. The most common approach to tracking is based on the Lucas-Kanade tracker [19], which approximately minimizes a cost function calculated as the sum of the squared differences in the intensities of the pixels belonging to the patches. The KLT tracker has been shown to work reliably only on "corner points", known as *Good Features to Track* (GFTT), and although efforts have been made to improve the performance of the tracker on said points [2, 8, 23], not many papers in the literature deal with the sparsity of such features. GFTT-points can be identified by a structure tensor with large values for both its eigenvalues [22]. The 2D motion of these points can be identified, in both directions, with a gradient descent method on the matching criterion (e.g. the SSD). A natural next step would be to track points having only one large eigenvalue for their structure tensor. Intuitively, these points lie mainly on linear edges. At these points, however, a local analysis can only provide the motion component in a direction perpendicular to the edge, corresponding to the eigenvector for the largest eigenvalue of the structure tensor. This is the well-known *aperture problem*.

A framework for matching edgels is proposed in [16], and is shown to outperform matching based on, e.g., SIFT or SURF, when a planar object is tracked. The limitation to planar objects restricts, however, the application of this approach. Another attempt to solve the aperture problem is made in [4], by jointly tracking edge points and nearby corner points with a method that combines the KLT tracker and the global Horn-Schunck method. In this framework, the motion of corner points helps in solving for the second degree of freedom of nearby edge points thanks to a regularization term inserted in the optimization process. A similar, but more generalized approach is taken in [26] where a semi-global matching method is adapted to work exclusively on fractions of the epipolar line. Also in this case, a regularization term is included in the optimization process to allow the matching of weaker points (edgels or even pixels lying on planar, textureless areas). The optimization problem is, however, NP-hard and non parallelizable so the dynamic programming approach used to solve it takes several seconds to compute.
In [15] the authors propose a method to obtain a dense optical flow which does not include a regularization term allowing each point to be treated individually. In this approach a Delaunay triangulation is performed over the motion field generated by a sparse matching framework to produce a prior on the motion field for all the points in the convex hull of the sparse features. This information is then used in combination with the estimated epipolar geometry and the trifocal tensor to steer a maximum-a-posteriori estimate for the best matching candi-

date for each pixel while also cutting down the number of possible candidates significantly

In our work, a different approach is taken. Corner points are often sufficient in number to produce a useful inter-frame motion estimate, which can be made even more accurate by the means of a global or local bundle adjustment, but are just too sparse to produce a dense reconstruction. Hence, there is no need to include edgels at this early stage. In most cases, including them at this stage would just make the egomotion estimate more computationally complex without a significant gain in accuracy and robustness. Instead, if we are only interested in making the motion flow in an image pair denser, the corresponding epipolar geometry is already available to a high degree of accuracy, and it is possible to formulate the KLT tracker as a 1-dimensional problem by constraining its optimization procedure exclusively on an epipolar line. Thus, edgels that are not parallel to the corresponding epipolar line can be tracked. We refer to this modified version of the KLT tracker as *Epipolar KLT*.

The idea of epipolar KLT has recently been proposed by Trummer et al [25], applied to the scenario of a camera mounted on a robotic arm for which the motion parameters are known. That work, however, does not fully exploit this tool for the tracking of specifically extracted edgels and, in fact, they reject the idea of completely constraining the tracker to the epipolar lines due to possible uncertainty in the epipolar geometry. Instead, they formulate a biased 2-dimensional tracker that favors steps in the direction of the epipolar line and reduces the freedom in the perpendicular direction with empirically chosen weights. Not completely trusting the epipolar geometry can be reasonable in high precision applications, but there is no reason to ignore already established camera poses when just aiming at making the motion field denser.

3 Methods

3.1 Background

The KLT tracker The classical KLT tracker has been repeatedly derived in literature, and we will only present parts of the derivation which are of interest for our modifications. For a complete derivation, see [19, 2, 5] among others. Let $W'_{\mathcal{J}}$ be a small window in the second image around a feature point, $\mathcal{I}(\mathbf{x})$ and $\mathcal{J}(\mathbf{x})$, respectively intensity value of the first and second images at position $\mathbf{x}$. In the simplest case, the change in the position of a point from $\mathcal{I}$ to $\mathcal{J}$ consists in a 2D translation, represented by the 2 values of the translation vector $\mathbf{v}$. Also, in practice, the inverse warping function is often used since it makes the procedure computationally more efficient. The expression for the cost function is:

$$G(\Delta\mathbf{v}) := \sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} (\mathcal{I}(\mathbf{x}'_i - (\mathbf{v} + \Delta\mathbf{v}) - \mathcal{J}(\mathbf{x}'_i))^2 \tag{1}$$

to be minimized w.r.t. $\Delta\mathbf{v}$. To solve the equation, a first order Taylor expansion is needed. Let $x_i = x'_i - (\mathbf{v} + \Delta\mathbf{v})$, setting, then, the first order derivative of the

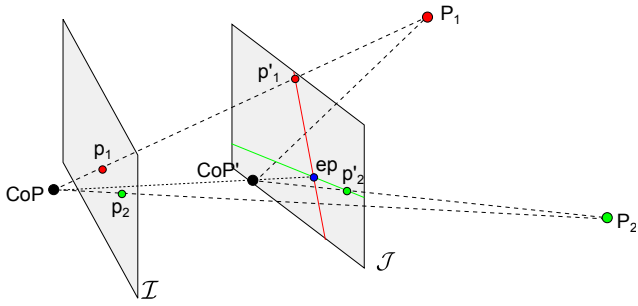

Fig. 2. Schematics of some geometric entities involved in two view geometry.

cost function to zero to find the stationary point we obtain:

$$\sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} \nabla\mathcal{I}(\mathbf{x}_i)\, \nabla\mathcal{I}(\mathbf{x}_i)^T \Delta\mathbf{v} = \sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} (\mathcal{I}(\mathbf{x}_i) - \mathcal{J}(\mathbf{x}'_i))\nabla\mathcal{I}(\mathbf{x}_i) \tag{2}$$

In matrix form we can write it as $\mathbf{A}\,\Delta\mathbf{v} = \mathbf{b}$. Solving for $\Delta\mathbf{v}$, the update rule for the iteration scheme is $\mathbf{v} = \mathbf{v} + \Delta\mathbf{v}$, and $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{v}$. The procedure is repeated until convergence or until some breaking criterion is met (for example, when the displacement is smaller than a threshold).

Epipolar Geometry This section contains a short review of the geometrical entities involved in epipolar geometry which are necessary to derive our algorithms. A more complete presentation is made, e.g., in [11]. With reference to Figure 2, for a given image pair $\mathcal{I}$ and $\mathcal{J}$ produced by two cameras, or the same camera in two different positions, having their center of projection in **CoP** and **CoP′** respectively, we can make the following observations (assuming lens distortion is absent or the images are rectified):

1. The projection of **CoP** in the image plane of $\mathcal{J}$ is the *epipole* **ep**.
2. For every 3D point $\mathbf{P}_k$ observed by both the cameras, the projection of the $\mathbf{P}_k$ in $\mathcal{J}$, $\mathbf{p}'_k$, must lie on a line in $\mathcal{J}$, given as the projection of the optic ray through $\mathbf{p}_k$ in $\mathcal{I}$, i.e., the 3D line connecting **CoP** and $\mathbf{p}_k$, onto $\mathcal{J}$. This line is the *epipolar line* generated by $\mathbf{p_k}$ in $\mathcal{J}$.
3. All the epipolar lines on an image plane, for a certain camera pair, meet in the corresponding epipole, **ep**.

Given the images $\mathcal{I}$ and $\mathcal{J}$, the corresponding epipolar geometry is represented by the fundamental matrix $\mathbf{F}$ The epipolar line in $\mathcal{J}$ corresponding to a point of pixel coordinates $\mathbf{x}$ in $\mathcal{I}$ is given by $\tilde{\mathbf{e}} = \mathbf{F}\tilde{\mathbf{x}} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^\top$, where $\tilde{\mathbf{x}}$ is the column vector of the homogeneous pixel coordinates of the point $\mathbf{x}$ and $\tilde{\mathbf{e}}$ is the dual homogeneous coordinates of the epipolar line. The 2-dimensional unity vector representing the direction of the epipolar line is then $\hat{\mathbf{e}} = \begin{bmatrix} e_2 & -e_1 \end{bmatrix}^T / \sqrt{e_1^2 + e_2^2}$.

Epipolar KLT We now have all the tools to derive the Epipolar KLT. Given an initialization on the epipolar line for the translation vector (i.e. $\mathbf{v}$ such that $\mathbf{x}+\mathbf{v}$ lies on $\tilde{\mathbf{e}}$), we can substitute $\hat{\mathbf{e}}$ in (1) to obtain a 1-dimensional cost function

$$G(\alpha) := \sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} (\mathcal{I}(\mathbf{x}'_i - (\mathbf{v} + \alpha\hat{\mathbf{e}})) - \mathcal{J}(\mathbf{x}'_i))^2 \tag{3}$$

to be minimized w.r.t. the scalar value α.

After the first order Taylor expansion and solving for the stationary point, we obtain the following expression to be iteratively minimized to converge toward the solution:

$$\alpha \sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} \nabla\mathcal{I}(\mathbf{x}_i)\, \nabla\mathcal{I}(\mathbf{x}_i)^T \hat{\mathbf{e}} = \sum_{\mathbf{x}'_i \in W'_{\mathcal{J}}} (\mathcal{I}(\mathbf{x}_i) - \mathcal{J}(\mathbf{x}'_i))\nabla\mathcal{I}(\mathbf{x}_i). \tag{4}$$

In matrix form we can write the last expression as

$$\alpha \mathbf{A}\, \hat{\mathbf{e}} = \mathbf{b}, \tag{5}$$

which is an over-determined 2 equation system. By pre-multiplying both sides by $\hat{\mathbf{e}}^T$ (i.e. projecting the 2D problem on the direction of the epipolar line), we project the system into one dimension: $\alpha\hat{\mathbf{e}}^T\mathbf{A}\,\hat{\mathbf{e}} = \hat{\mathbf{e}}^T\mathbf{b}$. Solving for α, the update rule for the iteration scheme is $\mathbf{v} = \mathbf{v} + \alpha\hat{\mathbf{e}}$ and $\mathbf{x}'_i = \mathbf{x}_i + \mathbf{v}$. The procedure is repeated until convergence or until some breaking criterion is met (for example, when the displacement is smaller than a threshold).

3.2 Good Edgels To Track

As already mentioned, assuming a static scene and a reliable estimate of the relative camera motion between two frames, we can track two different kinds of features:

- Corner features
- Edge features non parallel to the motion

Similarly to the case of corner features, edge features are identified from the eigenvalues of their structure tensor: they are associated with a structure tensor with at least one eigenvalue over a certain threshold. However, not all edge features are useful for our method: edges parallel to the their own epipolar line cannot be tracked reliably.

Let $\nabla\mathcal{I}_u(\mathbf{x})$ and $\nabla\mathcal{I}_v(\mathbf{x})$ be the horizontal and vertical gradient of the image calculated in $\mathbf{x}$ respectively. The structure tensor $T(\mathbf{x})$ for the point $\mathbf{x}$ is:

$$T(\mathbf{x}) = \begin{bmatrix} \nabla\mathcal{I}_u(\mathbf{x})^2 & \nabla\mathcal{I}_u(\mathbf{x})\nabla\mathcal{I}_v(\mathbf{x}) \\ \nabla\mathcal{I}_u(\mathbf{x})\nabla\mathcal{I}_v(\mathbf{x}) & \nabla\mathcal{I}_v(\mathbf{x})^2 \end{bmatrix}. \tag{6}$$

Given a pixel $\mathbf{x}$, its structure tensor T and its corresponding epipolar line $\tilde{\mathbf{e}}$ in the matching image, its score in the GETT sense is

$$score(\mathbf{x}) = \hat{\mathbf{e}}^T T(\mathbf{x})\, \hat{\mathbf{e}}, \tag{7}$$

where $\hat{\mathbf{e}}$ is the unit vector representing the direction of the epipolar line in $\mathcal{J}$ for $\mathbf{x}$. Note that the value of the score is bounded by the larger and smaller eigenvalues of the structure tensor and corresponds, intuitively, to measuring the structure tensor along the direction of the optimization process.

3.3 Epipolar KLT Initialization

To assure convergence to a reasonable solution it is critical to initialize the tracker on the epipolar line. In our tests we assume that no prior information on the depth of the points is available. We therefore assume that every point has an infinite depth, in this case we can find a starting point on the epipolar line that is also a hard limit for our procedure as only one of the 2 segments of the epipolar line defined by this point is physically reasonable.

Let $\mathbf{K}$ be the intrinsic parameter matrix for the camera and $\mathbf{R}$ the inter-camera rotation between the 2 views. For a point $\tilde{\mathbf{x}} \in \mathcal{I}$ in homogeneous coordinates, its infinity projection on $\mathcal{J}$ is given by

$$start(\tilde{\mathbf{x}}) = \mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}\tilde{\mathbf{x}}. \tag{8}$$

We refer to this initialization procedure as *initialization at infinity*.

Depending on the scene structure and the camera motion, this initialization can fail, in particular if the scene has large variation in depth and the translational motion component is significant relatively to the rotational component. This is the case for most driving sequences, such as those offered by the KITTI dataset [9], where this initialization is sensitive to repeating patterns and large displacements. This simple initialization, however, works in practice for the majority of the points. To increase the number of good matches, we do however perform an initialization in multiple steps as outlined in Algorithm 2. The details are given below.

4 Experiments

We tested our algorithm against the OpenCV [2] standard KLT implementation. For a fair comparison, the epipolar constraint has been imposed to the output of the standard tracker to reduce outliers. Both trackers have been fed with the same parameters in term of windows size (5×5), pyramid level (2nd level) and termination criteria (maximum 10 iterations, minimum displacement length of 0.1 pixels). The two testing schemes are outlined in Algorithms 1 and 2.

4.1 Implementation Details

In this subsection we go into the details of the testing scheme outlined in Algorithm 2.

[2] http://opencv.org

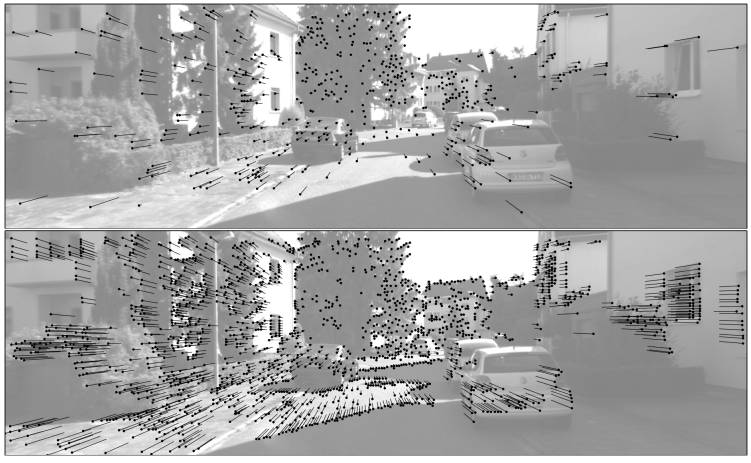

Fig. 3. Sample tracking output image from Sequence 1. **Top:** Tracking performed on GFTT with the standard KLT procedure. **Bottom: Tracking performed on GETT with the epipolar KLT.**

Algorithm 1 Testing pseudocode for standard KLT procedure

procedure TRACKGOODFEATURESTOTRACK(I, J, K, R, t)
  $p \leftarrow extractGoodFeaturesToTrack(I)$
  $q \leftarrow standardKLT(I, J, p)$
  $r \leftarrow standardKLT(J, I, q)$
  $corr \leftarrow \{(p_i, q_i) | \|p_i - r_i\| < \tau\}$
  $corr \leftarrow epipolarFilter(corr, F)$
  return *corr*
end procedure

Extraction of corners and edges We extract the corner and edge keypoints respectively according to the classical GFTT scheme and the hereby presented GETT scheme. The procedure works in the following way:

1. Extract the structure tensor for each pixel in the image.
2. Calculate the GFTT and GETT score for each pixel. The following steps are computed independently for GFTT and GETT points.
3. An Non-Maximum Suppression (**NMS**) filter is applied to the score images to obtain a better distribution of the keypoints. A sliding window is applied to each score image suppressing all the points which are not a local maximum. The window size is set at the same size of the tracking window at the lowest pyramid level.
4. A threshold τ is computed as a fraction of the highest score found in the image and the points with a score $> \tau$ are used as keypoints.

Tracker initialization The first initialization is given by the *initialization at infinity* presented in Section 3.3.

Algorithm 2 Testing pseudocode for our approach

procedure TRACKMIXEDFEATURES(I, J, K, R, t)
  $F \leftarrow K^{-T}[t]_x R K^{-1}$
  $p \leftarrow extractCornersEdges(I, F)$
  $start_p \leftarrow K R K^{-1} p$
  $q \leftarrow epipolarKLT(I, J, p, start_p, F)$
  $start_q \leftarrow K R^T K^{-1} q$
  $r \leftarrow epipolarKLT(J, I, q, start_q, F^T)$
  $corr \leftarrow \{(p_i, q_i) | \|p_i - r_i\| < \tau\}$
  $clusters \leftarrow clusterize(p)$
  $c_{MEAN} \leftarrow clustermeans(clusters, corr)$
  $start_p \leftarrow moveStartingPoints(start_p, c_{MEAN})$
  $q \leftarrow epipolarKLT(I, J, p, start_p, F)$
  $corr \leftarrow \{(p_i, q_i)\}$
  $c_{MEAN} \leftarrow clustermeans(clusters, corr)$
  $corr \leftarrow filterOutliers(corr, c_{MEAN})$
  return *corr*
end procedure

Tracking The tracking is done with the Epipolar KLT in Section 3.1, including a pyramidal *coarse-to-fine* scheme similar to the one presented in [5]. At the end of the tracking procedure, features presenting a large error (see [8]) or that moved on the wrong segment of the epipolar line are rejected as outliers.

Backtracking As suggested in [12] , a backtracking step is performed to further remove outliers. Features are tracked backwards from image $\mathcal{J}$ to image $\mathcal{I}$ and when the procedure converges to a point different from the original feature, the feature is considered an outlier. Note that this step is proven particularly useful in the case of repeating patterns due to the naivety of the initialization.

Clusterization and tracker reinitialization The tracks surviving the first two steps are in general of good quality, but are not dense due to the naive initialization step performed. Therefore, we compute a new initialization based on region-wise local mean displacement for the available tracks. The image is subdivided in macro-regions (in our tests we use a 3×7 grid to determine the regions) and the mean displacement along the epipolar line is robustly computed for each region, based on the available data of the surviving tracks belonging to the region. The displacement mean for each cluster of the image is robustly computed using the algorithm presented in [14]. This method works on 1-dimensional data, it orders the data and determines the mean giving the maximum amount of inliers, using a sliding window. The region-wise mean so computed is then used to reinitialize the tracker by moving the starting position of the tracker, away from the infinity projection of the feature, along the epipolar line.

Outlier removal The last tracking step has proven to provide a much higher inlier ratio than the initialization at infinity, making a new backtracking step

Method	Sequence 1	Sequence 2	Sequence 3	Total
GETT + EpiKLT	3542 ± 464	4439 ± 848	4603 ± 484	4196 ± 779
GFTT + KLT	2537 ± 425	3593 ± 881	3908 ± 551	3347 ± 874

Table 1. Average number of features extracted over the 3 sequences ± standard deviation

Method	Sequence 1	Sequence 2	Sequence 3	Total
GETT + EpiKLT	1700 ± 324	1656 ± 470	2121 ± 373	1818 ± 448
GFTT + KLT	509 ± 250	412 ± 238	909 ± 337	610 ± 352

Table 2. Average number of features tracked over the 3 sequences ± standard deviation

unnecessary. To remove the remaining outliers, we perform a filtering based on a re-computed region based mean, and remove points that do not behave according to the majority of the points in the region (we assume a Gaussian distribution and set the filter threshold at twice the standard deviation for the region).

Note that this complicated procedure is only necessary when no prior information on the depth structure of the scene is known. For image sequences, features can in general be tracked more than once, thus providing a depth estimate and a better initialization of the tracker. For new features, even accounting for reasonable depth discontinuities, the initialization can be performed by using the already available depth information of similar nearby points.

4.2 Results

We tested our method on sequences extracted from the KITTI dataset [9], consisting of driving scenes recorded by a calibrated stereo setup. Each of the extracted sequences consists of 199 image pairs and present different settings:

- Sequence 1 shows an urban scenario: mainly buildings on the sides of the road.
- Sequence 2 is set in a suburban scenario. Mostly trees surround the road
- Sequence 3 also has a suburban setting and shows a variety of buildings, vegetation and parked cars on the side of the road.

In our tests, we disregard the stereo information, focusing on a monocular scenario and making use of the given camera calibration. To show the robustness of the method, we do not use the ground truth egomotion provided with the KITTI dataset, we take instead the estimates produced by an implementation of [20]. The approach consists in a matching procedure performed on FAST keypoints [24] and BRIEF descriptors [6], the epipolar geometry estimation is made robust with RANSAC, and stabilized with a windowed bundle adjustment.

Tables 1 and 2 show, as expected, the higher density of GETT compared to GFTT which, combined with a better inlier ratio granted by the proposed tracking procedure, allows to sensibly increase density of the motion field. Notice

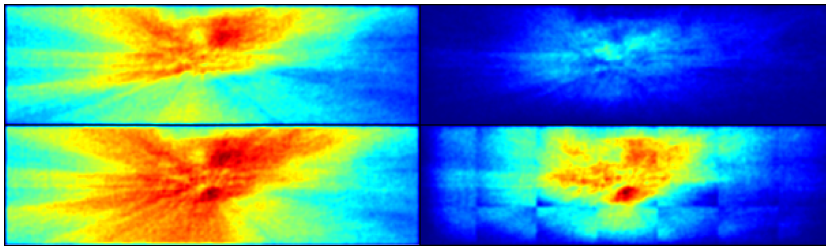

Fig. 4. Distribution of extracted GFTT (**top-left**) and GETT (**bottom-left:**). Distribution of successfully tracked GFTT (**top-right**) and GETT (**bottom-right**)

that, as shown in Figure 4 and 3 the increase in density is particularly evident in areas of the image closer to the camera and on the road plane. This is a very desirable feature for many applications since closeby objects are generally of more interest and can be measured more precisely.

5 Conclusions

In this work we presented a complete framework for the extraction and tracking of edge elements in an image pair. Such edge elements would normally suffer from the aperture problem and cannot be tracked successfully. By exploiting a known estimate of the epipolar geometry of the scene, our algorithm allows to extract and successfully track specific edgels that are likely to behave well with the given egomotion. We have shown that such keypoints (*GETT*) consistently outnumber the standard *GFTT* in different settings, have a higher inlier ratio when tracked with the procedure we presented and are also more uniformly spread on the image plane. These are all very desirable qualities that allow to produce a much denser point cloud in any structure from motion application.

References

1. Agrawal, M., Konolige, K., Blas, M.R.: Censure: Center surround extremas for realtime feature detection and matching. In: Europ. Conf. on Computer Vision (ECCV 2008). pp. 102–115 (2008)
2. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A unifying framework. International Journal of Computer Vision 56(3), 221–255 (Feb 2004)
3. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Europ. Conf. Computer Vision (ECCV) pp. 404–417 (2006)
4. Birchfield, S.T., Pundlik, S.J.: Joint tracking of features and edges. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008) (2008)
5. Bouguet, J.Y.: Pyramidal implementation of the affine Lucas Kanade feature tracker: description of the algorithm. Tech. rep., Intel Corporation (2001)
6. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. Europ. Conf. on Computer Vision (ECCV 2010) pp. 778–792 (2010)

7. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Image Analysis, Lecture Notes in Computer Science, vol. 2749, pp. 363–370. Springer (2003)
8. Fusiello, A., Trucco, E.: Improving feature tracking with robust statistics. Pattern Analysis & Applications pp. 312–320 (1999)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2012) (2012)
10. Harris, C., Stephens, M.: A combined corner and edge detector. Alvey vision conference pp. 23.1–23.6 (1988)
11. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
12. Hedborg, J., Forssén, P., Felsberg, M.: Fast and accurate structure and motion estimation. In: Advances in Visual Computing, Lecture Notes in Computer Science, vol. 5875, pp. 211–222. Springer (2009)
13. Horn, B., Schunck, B.: Determining optical flow. In: SPIE 0281, Techniques and Applications of Image Understanding. vol. 319 (1981)
14. Jonsson, E., Felsberg, M.: Efficient robust mean value computation of 1D features. In: Proceedings of Svenska Sällskapet för Automatiserad Bildanalys. SSBA-2005 (2005)
15. Kitt, B., Lategahn, H.: Trinocular optical flow estimation for intelligent vehicle applications. In: International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 300 – 306. IEEE (Sep 2012)
16. Lee, T., Soatto, S.: Fast planar object detection and tracking via edgel templates. In: IEEE Workshop on the Applications of Computer Vision (WACV). pp. 473–480. IEEE (Jan 2012)
17. Leutenegger, S.: BRISK: Binary robust invariant scalable keypoints. In: IEEE Int. Conf. on Computer Vision (ICCV), 2011. pp. 2548–2555 (Nov 2011)
18. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2), 91–110 (Nov 2004)
19. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intellicence (IJCAI) (1981)
20. Persson, M.: Online Monocular SLAM. Master's thesis, Computer Vision Laboratory, Linköping University, Sweden (December 2013)
21. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: IEEE Int. Conf. on Computer Vision (ICCV). pp. 2564–2571 (2011)
22. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conf. on Computer Vision and Pattern Recognition CVPR-94. pp. 593–600 (1994)
23. Tommasini, T., Fusiello, A.: Making good features track better. IEEE Conf. on Computer Vision and Pattern Recognition pp. 178 – 183 (1998)
24. Trajković, M., Hedley, M.: Fast corner detection. Image and Vision Computing 16(1998), 75–87 (1998)
25. Trummer, M., Denzler, J., Munkelt, C.: KLT tracking using intrinsic and extrinsic camera parameters in consideration of uncertainty. International Conference on Computer Vision Theory and Applications (VISAPP) (2008)
26. Yamaguchi, K., McAllester, D., Urtasun, R.: Robust monocular epipolar flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1862 – 1869. IEEE (Jun 2013)