Combined GKLT Feature Tracking and Reconstruction for Next Best View Planning

Michael Trummer[1], Christoph Munkelt[2], and Joachim Denzler[1]

[1] Friedrich-Schiller University of Jena, Chair for Computer Vision
Ernst-Abbe-Platz 2, 07743 Jena, Germany
{michael.trummer,joachim.denzler}@uni-jena.de
[2] Fraunhofer Society, Optical Systems
Albert-Einstein-Straße 7, 07745 Jena, Germany
christoph.munkelt@iof.fraunhofer.de

Abstract. *Guided* Kanade-Lucas-Tomasi (GKLT) tracking is a suitable way to incorporate knowledge about camera parameters into the standard KLT tracking approach for feature tracking in rigid scenes. By this means, feature tracking can benefit from additional knowledge about camera parameters as given by a controlled environment within a next-best-view (NBV) planning approach for three-dimensional (3D) reconstruction. We extend the GKLT tracking procedure for controlled environments by establishing a method for combined 2D tracking and robust 3D reconstruction. Thus we explicitly use the knowledge about the current 3D estimation of the tracked point within the tracking process. We incorporate robust 3D estimation, initialization of lost features, and an efficient detection of tracking steps not fitting the 3D model. Our experimental evaluation on real data provides a comparison of our extended GKLT tracking method, the former GKLT, and standard KLT tracking. We perform 3D reconstruction from predefined image sequences as well as within an information-theoretic approach for NBV planning. The results show that the reconstruction error using our extended GKLT tracking method can be reduced up to 71% compared to standard KLT and up to 39% compared to the former GKLT tracker.

1 Introduction and Literature Review

Three-dimensional reconstruction from digital images requires a solution to the correspondence problem. Feature tracking, especially KLT tracking [1] in an image sequence is a commonly accepted approach to establish point correspondences between images of the input sequence. A point correspondence between two images consists of the two image points that are mappings of the same 3D world point. Together with calibration data, in particular the intrinsic and extrinsic camera parameters, these point correspondences are used to estimate the position of the respective 3D world point.

The original formulation of KLT tracking by Lucas and Kanade in [1] entailed a rich variety of extensions, lots of them reviewed by Baker and Matthews in [2]. Fusiello et al. [3] remove spurious correspondences by an outlier detection based

J. Denzler, G. Notni, and H. Süße (Eds.): DAGM 2009, LNCS 5748, pp. 161–170, 2009.
© Springer-Verlag Berlin Heidelberg 2009

on the image residuals. Zinsser et al. [4] propose a separated tracking process by inter-frame translation estimation using block matching followed by estimating the affine motion with respect to the template image.

Recent research [5,6] deals with purposive 3D reconstruction within a controlled environment (e.g. Fig. 1) by planning camera positions that most support the respective task. Such planning methods calculate camera positions that, for instance, allow the most complete reconstruction of an object with a certain number of views or that optimize the accuracy of reconstructed points. This field of application is an example, where additional knowledge about camera parameters as available and should be used to improve feature tracking. Heigl [7] uses an estimation of camera parameters to move features along their epipolar line, but he does not consider the uncertainty of the estimation. Trummer et al. [8] give a formulation of KLT tracking with known camera parameters regarding uncertainty, called *Guided KLT tracking (GKLT)*, but still use the traditional optimization error function. In [9] the authors extend the error function and the optimization algorithm of GKLT to handle uncertainty estimation together with the estimation of transformation parameters.

Fig. 1. Robotic arm *Stäubli RX90L* as an example of a controlled environment

In this paper we present an extension of GKLT tracking resulting in combined tracking and reconstruction. We perform the reconstruction by robustly estimating the position of the respective 3D point. This step endows efficient detection of spurious tracking steps not fitting the current 3D model during the tracking process as well as reinitialization of lost features. We further compare our extended GKLT tracking method with standard KLT and previous GKLT tracking methods in the context of NBV planning using the NBV benchmark object proposed in [10].

The remainder of this paper is organized as follows. Section 2 gives a review of standard KLT tracking and the previous versions of GKLT tracking. In Sect. 3 we present our extended GKLT tracking for combined tracking and reconstruction. A comparison of the considered tracking methods within a NBV planning approach is carried out in Sect. 4. The conclusion of this paper and the outlook to future work is given in Sect. 5.

2 Review of KLT and GKLT Tracking

This section briefly reviews the relevant tracking methods as seen from literature [1,2,8,9]. Thus the notation is defined and the previous extensions of KLT tracking for the usage of camera parameters are described.

2.1 KLT Tracking

Given a feature position in the initial frame, KLT feature tracking aims at finding the corresponding feature position in the consecutive input frame with intensity function $I(\mathbf{x})$. The initial frame is the template image with intensity function $T(\mathbf{x})$, $\mathbf{x} = (x, y)^T$. A small image region and the intensity values inside describe a feature. This descriptor is called feature patch P. Tracking a feature means that the parameters $\mathbf{p} = (p_1, ..., p_n)^T$ of a warping function $W(\mathbf{x}, \mathbf{p})$ are estimated iteratively, trying to minimize the squared intensity error over all pixels in the feature patch. A common choice is affine warping by

$$W^a(\mathbf{x}, \mathbf{p}^a) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \tag{1}$$

with $\mathbf{p}^a = (\Delta x, \Delta y, a_{11}, a_{12}, a_{21}, a_{22})^T$. Following the additive approach (cf. [2]), the error function of the optimization problem can be written as

$$\epsilon(\Delta\mathbf{p}) = \sum_{\mathbf{x} \in P} (I(W(\mathbf{x}, \mathbf{p} + \Delta\mathbf{p})) - T(\mathbf{x}))^2, \tag{2}$$

where the goal is to find $\arg\min_{\mathbf{p}} \epsilon(\Delta\mathbf{p})$. An iterative update rule for $\Delta\mathbf{p}$ is found by first-order Taylor approximations of the error function (2).

2.2 Guided KLT Tracking

In comparison to standard KLT tracking, GKLT uses knowledge about intrinsic and extrinsic camera parameters to alter the translational part of the warping function. Features are moved along their respective *epipolar* line, but allowing for translations perpendicular to the epipolar line caused by the *uncertainty* in the estimate of the epipolar geometry. The affine warping function (1) is changed to

$$W^a_{EU}(\mathbf{x}, \mathbf{p}^a_{EU}, \mathbf{m}) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{-l_3}{l_1} - \lambda_1 l_2 + \lambda_2 l_1 \\ \lambda_1 l_1 + \lambda_2 l_2 \end{pmatrix} \tag{3}$$

with $\mathbf{p}^a_{EU} = (\lambda_1, \lambda_2, a_{11}, a_{12}, a_{21}, a_{22})^T$ and $l_1 \neq 0$; the respective epipolar line $\mathbf{l} = (l_1, l_2, l_3)^T = \mathbf{F}\tilde{\mathbf{m}}$ is computed using the fundamental matrix $\mathbf{F}$ and the feature position (center of feature patch) $\tilde{\mathbf{m}} = (x_m, y_m, 1)^T$. The first version of GKLT [8] uses a weighting matrix in the parameter update rule to control the feature's translation along and perpendicular to the respective epipolar line. In [9] a new optimization error function for GKLT is proposed. The weighting matrix $\mathbf{A}_{w,\Delta w}$ and thus the uncertainty parameter w is included in the modified error function

$$\epsilon(\Delta\mathbf{p}_{EU}, \Delta w) = \sum_{\mathbf{x} \in P} (I(W_{EU}(\mathbf{x}, \mathbf{p}_{EU} + \mathbf{A}_{w,\Delta w}\Delta\mathbf{p}_{EU}, \mathbf{m})) - T(\mathbf{x}))^2. \tag{4}$$

This results in an EM-like approach for a combined estimation of the uncertainty and the warping parameters.

By this means, Guided KLT tracking uses additional knowledge about camera parameters to optimize the tracking process in the 2D image space.

3 Combining GKLT Tracking and Robust 3D Reconstruction

In this section we present a combined approach for GKLT feature tracking and 3D reconstruction of the respective 3D world point. We show how 2D tracking can benefit from an online 3D estimation using robust statistics. For a compact formulation, we denote our extended GKLT tracking method as GKLT_{3D}.

Table 1. Comparing flowcharts of KLT and GKLT_{3D} methods. Steps describe actions for tracking one feature in one frame. Further explanations are given in Sect. 3.

KLT method	**GKLT_{3D} method**
init feature position: detect feature (new) OR init feature from last position (tracked) OR *stop* without tracking (lost)	init feature position: detect feature (new) OR init feature from last position (tracked) OR *reinit* from back-projection (lost)
↓	↓
if initialized: *KLT* tracking	*always*: *GKLT* tracking
	↓
	if tracking successful: init weights for 3D estimation
	↓
	robust estimation of 3D position
	↓
	check tracking step for acceptance

The GKLT_{3D} tracking method consists of the following steps for tracking one feature in one frame, cf. Table 1.

Initialize feature position. Since tracking in the KLT sense is an iterative optimization of feature transformation parameters, an initial solution is required. If the feature was tracked in the previous frame, it is straightforward to use the last parameter estimation as the initialization for the current frame, which corresponds to the condition of small baselines between consecutive frames. We also use this initialization technique for GKLT_{3D}. In addition, GKLT_{3D} reinitializes features that were lost in the previous frame or earlier and that were tracked in at least one frame. Thus a 3D estimation from at least two frames exists, in particular from the frame where the feature was detected and from at least one frame of successful tracking. For lost features we use the back-projection of the estimated 3D point to reinitialize the feature position for GKLT_{3D} tracking.

GKLT tracking. Having initialized the feature transformation, we perform 2D feature tracking by the GKLT method elaborated in [8,9]. In fact, this step of the GKLT_{3D} method can be performed by any other tracking method including standard KLT tracking. However, we find it natural to further extend the existing GKLT tracking method that already uses knowledge about camera parameters.

Initialize weights for 3D estimation. After successful feature tracking we include the additional information about the actual feature position in the 3D estimation. Since we use an iterative estimation and robust statistics, we need to initialize each weight $w_i \in [0,1]$ for the feature position $\mathbf{x}_i$ in frame i. The only w_i we can know for sure is $w_0 = 1$; frame 0 is the initial frame where the feature is detected. The feature positions tracked in the following frames are afflicted with increasing uncertainty. It is more likely for them to be outliers. Thus we propose a strictly decreasing sequence $(w_i^{(\mathrm{init})})_{i=0,1,\ldots,n}$ with

$$w_0^{(\mathrm{init})} = 1 \ \text{ and } \ \forall i > 0 : w_i^{(\mathrm{init})} < w_{i-1}^{(\mathrm{init})} \tag{5}$$

as initialization for the weights w_i. In the presence of output weights from a previous 3D estimation, we initialize the position weights with

$$w_i^{(\mathrm{init})} = \begin{cases} 1 & , i = 0 \\ w_i^{(\mathrm{prev})} & , 1 \leq i \leq n-1 \\ 0.5 & , i = n \end{cases} \tag{6}$$

and hence ensure that $w_0^{(\mathrm{init})} = 1$, initialize the weight regarding the latest tracked position as $w_n^{(\mathrm{init})} = 0.5$ and use the previously adapted weights $w_i^{(\mathrm{prev})}$, $i = 1, ..., n-1$.

Robust estimation of 3D position. For 3D reconstruction we use the known camera parameters and a robust adaptation of the standard direct linear transform (DLT) algorithm for 3D triangulation [11] to perform an estimation following the idea of iteratively reweighted least squares (IRLS) estimation [12]. Since the DLT algorithm endows rather an algebraically optimal than a least squares estimation, we use robust iteratively reweighted DLT (IRDLT) estimation of the 3D position. We apply the error norm proposed by Huber [13] as robust estimator,

$$\rho(e) = \begin{cases} \frac{1}{2}e^2 & , |e| < t \\ t|e| - \frac{1}{2}t^2 & , |e| \geq t \end{cases} \tag{7}$$

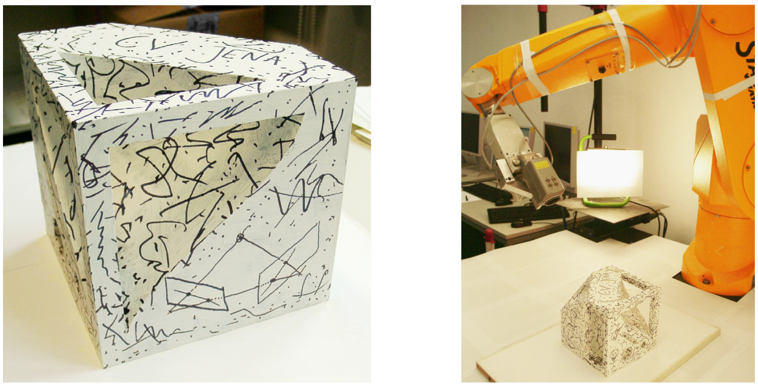
which yields the weight function

$$w(e) = \frac{1}{e}\frac{\partial \rho(e)}{\partial e} = \begin{cases} 1 & , |e| < t \\ -\frac{t}{e} & , |e| \geq t \wedge e < 0 \\ \frac{t}{e} & , |e| \geq t \wedge e \geq 0 \end{cases} \tag{8}$$

for error e and outlier boundary t. The IRDLT estimation algorithm performs the following steps to compute an estimation $\hat{\mathbf{X}}$ of 3D point $\mathbf{X}$ from image points $\mathbf{x}_i$ and projection matrices $\mathbf{P}_i$ using weights w_i, $i = 0, 1, ..., n$:

<u>preparation:</u> init weights w_i for 3D reconstruction according to (6), if previously estimated weights available, else according to (5)
1) perform 3D reconstruction using weighted DLT algorithm
2) recompute weights w_i following (8)
3) if changes of w_i are small, stop; else go to 1)

These steps endow a costly-inexpensive and robust 3D estimation $\hat{\mathbf{X}}$ of the world point $\mathbf{X}$.



(a) View of the NBV test object ...

(b) ... inside a controlled environment.

Fig. 2. All-aluminium NBV test object proposed in [10]. Outstanding artistic design to provide optical surface structure and hence features for tracking.

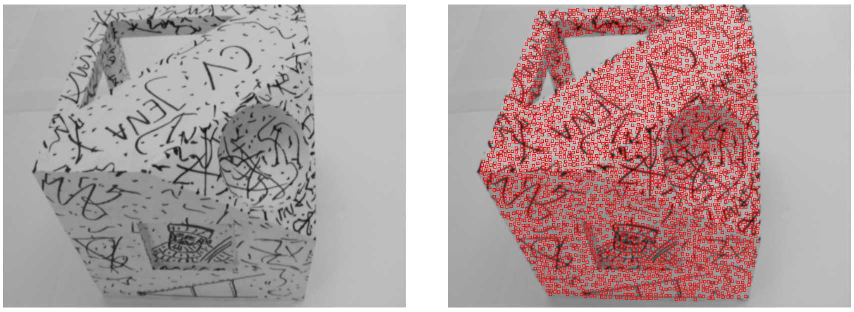
Check tracking step for acceptance. Besides a robust estimation of the tracked point's 3D position, the IRDLT procedure yields weights $w_i \in [0, 1]$. These weights indicate how likely it is for a position $\mathbf{x}_i$ to be an outlier, whereat $w_i = 1$ states that position $\mathbf{x}_i$ in image i perfectly supports the estimated 3D position $\hat{\mathbf{X}}$. We use the weight w_n of the last tracked position $\mathbf{x}_n$ to decide for acceptance of the whole tracking step. If $w_n < t_w$, e.g. $t_w = 0.5$, we roll back the whole tracking step of GKLT$_{3D}$, i.e. we restore the previous 3D estimation and delete position $\mathbf{x}_n$. In this case the current feature position is reinitialized from the 3D estimate instead of the outlying tracked position for the consecutive frame.

All the steps described above form the GKLT$_{3D}$ tracking method, cf. Table 1. The cycle for tracking one feature in one frame performs 2D GKLT tracking, robust estimation of the 3D position, and usage of the estimated 3D information. The outlier rejection is based on the coherence of the latest tracked image position and the robustly estimated 3D position of the respective world point. Thus the 2D tracking process benefits from the concurrent robust 3D estimation in terms of reinitialization of lost features, outlier detection regarding the robust 3D estimate, and, of course, in terms of the robustly estimated 3D position itself.

4 Experimental Comparison of KLT, GKLT and GKLT$_{3D}$ Tracking

We compare our GKLT$_{3D}$ to the standard KLT and GKLT tracking methods. As input data we use predefined image sequences as well as a planned sequence produced by the information-theoretic NBV planning approach described in [5].

Figure 2 shows the experimental setup. All image sequences are taken with a calibrated camera Sony DFW-VL 500 mounted on a robotic arm Stäubli RX90L providing position parameters. Figures 2(a) and 2(b) show the NBV test object proposed in [10]. The image sequences are taken from camera positions on a



(a) Initial frame. (b) 3227 features (red boxes) selected.

Fig. 3. Initial frame and selected features

half sphere over the object. The test object itself is manufactured from its CAD model with an accuracy of 30μm. From this CAD model we derive a very dense point cover of the object surface, in particular 10^6 points equally distributed on the object surface. After transformation to the robot coordinate frame this point set provides ground-truth reference data for the 3D reconstruction.

For quantitative evaluation of the tracking and reconstruction results, we use the following criteria. We measure the tracking performance by noting the mean trail length μ_L and the standard deviation σ_L in frames. The reconstruction accuracy is measured by mean error μ_E and standard deviation σ_E in mm. For this we calculate the distances between each reconstructed point and the respectively closest point from the reference point set. For a meaningful comparison of the reconstruction – and hence tracking – accuracies, we just use the trails in the 2D image space produced by each tracker to perform 3D reconstruction with the standard DLT triangulation algorithm. Thus we do not evaluate the robust 3D estimates from $\mathrm{GKLT}_{\mathrm{3D}}$. Each 3D point available is included in the evaluation, i.e. each point that has been seen in at least two frames. Figure 3 shows the initial frame of all test sequences and the set of 3227 features selected along image edges for tracking within the predefined sequences. For NBV planning we reduce the set of features considering the planning runtime.

4.1 Comparison Using a Short Image Sequence

We perform feature tracking and 3D point reconstruction using as few as ten frames for tracking. The 11 frames, one for feature detection and ten for tracking, are taken moving the camera on a meridian of the half sphere over the object with the camera directed to the center of the corresponding sphere. Since the baseline between consecutive frames on the meridian is 0.375 and thus very small, the whole sequence covers a small baseline only. Considering the fact that small 2D position errors cause large 3D errors in the presence of a small baseline, the reconstruction results emphasize the tracking accuracy.

The results in Table 2 show that the mean trail length compared to the standard KLT is increased about 11% with both the GKLT and the $\mathrm{GKLT}_{\mathrm{3D}}$

Table 2. Comparison of trail lengths ($_L$) and reconstruction errors ($_E$) for tracking the features from Fig. 3(b) in a short sequence of 11 frames, one frame for feature detection. GKLT$_{3D}$ offers best accuracy

	μ_L(frames)	σ_L(frames)	μ_E(mm)	σ_E(mm)
KLT	9.56	2.60	7.62	25.27
GKLT	10.67 (+11.61%)	1.29 (−50.38%)	3.46 (−54.59%)	6.30 (−75.07%)
GKLT$_{3D}$	10.64 (+11.30%)	1.36 (−47.70%)	2.75 (−63.91%)	2.74 (−89.16%)

tracker. More considerably GKLT$_{3D}$ reduces the mean reconstruction error by about 64% and the standard deviation by about 89% compared to the standard KLT tracker. With respect to GKLT, GKLT$_{3D}$ reduces μ_E by 20.52% and σ_E by 56.51%. GKLT$_{3D}$ benefits from the removal of spurious tracking steps not fitting the 3D estimation.

4.2 Comparison Using a Long Image Sequence

In addition to the tracking evaluation using a short image sequence we further apply the tracking methods to the same image features within a long sequence of 201 frames, one for feature detection and 200 for tracking. In this sequence the camera positioning covers a large baseline and change of the viewing direction. By this means, we achieve a meaningful evaluation of the tracking durations.

As shown in Table 3, tracking features in the long image sequence points out the benefits of reinitializing lost features, which requires an estimation of the respective 3D world point. Considering the average case, GKLT$_{3D}$ can track features for nearly four times more frames than standard KLT and nearly three times more than GKLT in the test sequence. This also entails a larger standard daviation σ_L. The difference of the mean reconstruction errors is even larger than for the short test sequence. The mean error produced by GKLT$_{3D}$ is about 71% smaller compared to standard KLT and about 39% smaller compared to GKLT. In comparison with the results using the short test sequence, only GKLT$_{3D}$ can improve the reconstruction accuracy; standard KLT and GKLT produce larger mean errors. This seems contradictory, since the longer test sequence covers a larger baseline and features are tracked in more frames. Actually, standard KLT and GKLT suffer from tracking inaccuracies due to difficult imput images that superimpose the effect of the larger baseline. By robustly estimating the current

Table 3. Comparison of trail lengths ($_L$) and reconstruction errors ($_E$) for tracking the features from Fig. 3(b) in a long sequence of 201 frames, one frame for feature detection. GKLT$_{3D}$ shows by far the best tracking duration and reconstruction accuracy in the comparison.

	μ_L(frames)	σ_L(frames)	μ_E(mm)	σ_E(mm)
KLT	23.47	21.22	9.10	27.26
GKLT	33.88 (+44.35%)	21.70 (+2.36%)	4.34 (−52.31%)	6.69 (−75.46%)
GKLT$_{3D}$	91.06 (+287.98%)	41.90 (+97.46%)	2.65 (−70.88%)	2.38 (−91.27%)

3D position and removing spurious tracking steps, only GKLT_{3D} allows a more accurate 3D reconstruction using the long test sequence.

4.3 Comparison within an Information-Theoretic Approach for Next Best View Planning

Compared to the standard structure-from-motion approach, 3D reconstruction within a controlled environment offers additional information and allows purposive actions to improve the reconstruction procedure and the result. NBV planning uses these additional possibilities to achieve defined reconstruction goals. The NBV planning method in [5] uses an extended Kalman filter to compute 3D reconstructions of tracked features and determines the next best view by applying an information-theoretic quality criterion and visibility constraints. We track 496 features and use the short test sequence as the initial sequence of the planning procedure.

Table 4. Comparison of reconstruction errors after the n-th planned view NBV_n for respective tracking method. GKLT_{3D} allows more planned views and yields best accuracy.

	init		NBV_1		NBV_2		NBV_3	
	μ_E	σ_E	μ_E	σ_E	μ_E	σ_E	μ_E	σ_E
KLT	5.20	21.47	5.42	21.46	/	/	/	/
GKLT	2.62	3.14	2.70	3.51	/	/	/	/
GKLT_{3D}	2.10	1.43	1.74	1.26	1.79	1.16	1.78	1.16

Table 4 lists the reconstruction errors after each iteration of the NBV planning procedure. The feature trails provided by KLT and GKLT tracking allow only one planned view after the initial sequence. Afterwards, the respective planning result repeats itself, since both trackers cannot keep the features through the longer sequence. Only GKLT_{3D} can gather enough information in the elongated sequence to provide new information for the next planning step, which allows a new planned position. We stopped planning with GKLT_{3D} after the third planned view. The mean reconstruction error of about 1.75mm using GKLT_{3D} clearly outperforms the results reached with KLT and GKLT tracking.

5 Conclusion and Future Work

We presented an extension to GKLT feature tracking within a controlled environment. Following the idea of using the additional knowledge about camera parameters within the tracking process, we described concurrent robust estimation of the 3D position from 2D feature positions by the IRDLT algorithm. We used the robustly estimated 3D position to reinitialize lost features during the tracking process and to detect and remove spurious tracking steps not supporting the current 3D estimation. Further we performed an experimental evaluation using defined image sequences as well as within an information-theoretic approach for next-best-view planning.

The experimental evaluation outlined a clear performance gain using our extended GKLT tracking method – considering tracking duration as well as tracking accuracy. In comparison to the standard KLT and the former GKLT trackers, the mean reconstruction error in the experiments was reduced by up to 71% and 39%, respectively. The gain in the tracking duration increased with longer image sequences. We noted an increase of about 290% with a long test sequence.

Future work should deal with the bottleneck of constant feature templates. The reinitialization of lost features yields no positive effect if the current view shows the feature through a completely different perspective projection than seen in the initial frame. A solution to this problem also should use the additional knowledge about camera parameters.

References

1. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of 7th International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
2. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. International Journal of Computer Vision 56, 221–255 (2004)
3. Fusiello, A., Trucco, E., Tommasini, T., Roberto, V.: Improving feature tracking with robust statistics. Pattern Analysis and Applications 2, 312–320 (1999)
4. Zinsser, T., Graessl, C., Niemann, H.: High-speed feature point tracking. In: Proceedings of Conference on Vision, Modeling and Visualization (2005)
5. Wenhardt, S., Deutsch, B., Angelopoulou, E., Niemann, H.: Active Visual Object Reconstruction using D-, E-, and T-Optimal Next Best Views. In: Computer Vision and Pattern Recognition. CVPR 2007, June 2007, pp. 1–7 (2007)
6. Chen, S.Y., Li, Y.F.: Vision Sensor Planning for 3D Model Acquisition. IEEE Transactions on Systems, Man and Cybernetics – B 35(4), 1–12 (2005)
7. Heigl, B.: Plenoptic Scene Modelling from Uncalibrated Image Sequences. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2003)
8. Trummer, M., Denzler, J., Munkelt, C.: KLT Tracking Using Intrinsic and Extrinsic Camera Parameters in Consideration of Uncertainty. In: Proceedings of 3rd International Conference on Computer Vision Theory and Applications (VISAPP), vol. 2, pp. 346–351 (2008)
9. Trummer, M., Munkelt, C., Denzler, J.: Extending GKLT Tracking – Feature Tracking for Controlled Environments with Integrated Uncertainty Estimation. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 460–469. Springer, Heidelberg (2009)
10. Munkelt, C., Trummer, M., Wenhardt, S., Denzler, J.: Benchmarking 3D Reconstructions from Next Best View Planning. In: Proceedings of IAPR Conference on Machine Vision Applications (MVA), pp. 552–555 (2007)
11. Hartley, R., Zisserman, A.: Multiple View Geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge (2003)
12. Maronna, R., Martin, R., Yohai, V.: Robust Statistics. Wiley Series in Probability and Statistics (2006)
13. Huber, P.: Robust Estimation of a Location Parameter. Annals of Mathematical Statistics 35(1), 73–101 (1964)