ROBUST LOCAL OPTICAL FLOW ESTIMATION USING BILINEAR EQUATIONS FOR SPARSE MOTION ESTIMATION

Tobias Senst, Jonas Geistert, Ivo Keller and Thomas Sikora

Technische Universität Berlin
Communication Systems Group
EN 1, Einsteinufer 17, 10587 Berlin, Germany

ABSTRACT

This article presents a theoretical framework to decrease the computation effort of the Robust Local Optical Flow method which is based on the Lucas Kanade method. We show mathematically, how to transform the iterative scheme of the feature tracker into a system of bilinear equations and thus estimate the motion vectors directly by analyzing its zeros. Furthermore, we show that it is possible to parallelise our approach efficiently on a GPU, thus, outperforming the current OpenCV-OpenCL implementation of the pyramidal Lucas Kanade method in terms of runtime and accuracy. Finally, an evaluation is given for the Middlebury Optical Flow and the KITTI datasets.

Index Terms— Optical flow, KLT, feature tracking, RLOF, OpenCL, GPU

1. INTRODUCTION

Motion information has become an important cue in many video-based computer vision applications, not least because the accuracy and efficiency of motion estimation techniques have been substantially improved in recent years.

The most common motion estimation methods are based on the concept of optical flow [1] and can be classified as global and local approaches. Global approaches are based on a system of equations in which the resulting motion of each point depends on the data of the whole image through coupled energy terms as, e.g., the smoothness constraint proposed by Horn and Schunck [2]. In this way, these methods are able to estimate the optical flow very accurately [1] but only scalable related to the image size.

In contrast, local approaches are based on a system of equations in which a motion vector depends on the textural information of a limited image region. Local approaches are scalable related to the number of motion vectors and are very efficient in estimating sparse motion information. There

The research leading to these results has received funding from the European Communitys FP7 under grants agreement number 261743 (NoE VideoSense) and number 261776 (MOSAIC).

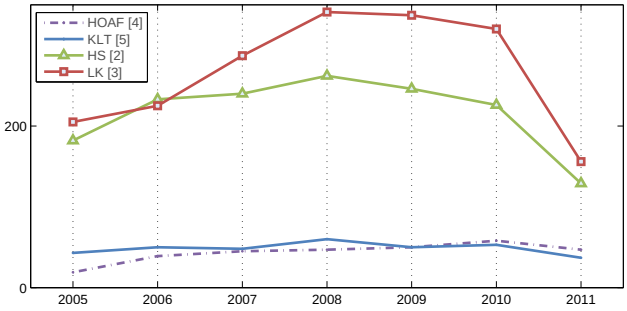

Fig. 1. Citations per year of gradient based optical flow methods for the origin global [2] and local [3] approach and the HOAF [4] and KLT [5] as common global and local methods (source: Microsoft Academic Search).

are still misunderstandings in comparing the accuracy of local and global optical flow methods since the evaluation is based on dense vector fields. For that reason local approaches are barely represented in the Middlebury evaluation [1] although they are still used in many applications such as robot navigation, augmented reality or Lagrangian based crowd and pedestrian analysis which could be deduced from the number of citations, see figure 1. The KITTI benchmark [6] provides the opportunity of comparing sparse and dense motion information by choosing different evaluation criteria. Comparing only the estimated pixels the RLOF (Robust Local Optical Flow) [7] outperforms the global ones in terms of accuracy and runtime, see Figure 3.

In the recent past, research on local optical flow was based on the KLT (Kanade Lucas Tomasi) tracker [5] and motivated by improving the runtime performance through parallelisation, e.g. GPU implementations were proposed by Sinha *et al.* [8] and Zach *et al.* [9] or by reducing the computational complexity through additional approximations, e.g. integral projections [10]. In [11] the authors proposed a non-iterative warping approach for the KLT method that avoids the recomputation of the mismatch vector between the pixel borders and thus achieves a reduced runtime without losses in accuracy. To enhance the accuracy of the motion vectors,

Kim *et al.* [12], Odebez *et al.* [13] and Senst *et al.* [7] investigated into norms that are robust against outliers.

In this paper we will introduce a fast and accurate local optical flow method for sparse motion estimation that is based on the RLOF and motivated by the work of Rav-Acha and Peleg [11]. We will mathematically show how to transform the iterative solution scheme of the RLOF into a system of bilinear equation and thus derive a scheme to directly solve a defined set of motion vectors. In contrast to [11], the proposed method does not only avoid the re-computation of the mismatch vector, but it also includes a strategy to compute the respective motion vector directly, which results in an additional runtime gain.

2. RLOF BY MEANS OF BILINEAR EQUATIONS

In order to introduce the mathematical notation used in this paper, we will first briefly review the Lucas Kanade method. The computation of a motion vector $\mathbf{d} = (u, v)^T$ is given by minimizing the following generalized gradient-based optical flow equation [7]:

$$\underset{\mathbf{d}}{min} \sum_{\Omega} w(\mathbf{x}) \cdot \rho \left(\nabla I(\mathbf{x})^T \cdot \mathbf{d} + I_t(\mathbf{x}), \boldsymbol{\sigma} \right) \tag{1}$$

The displacement $\mathbf{d}$ for a small region Ω at time t is estimated depending on the spatial derivatives $\nabla I(\mathbf{x})$ and the temporal derivative $I_t(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x}, t+1)$ of a grayscale image $I(\mathbf{x}, t)$ for $\mathbf{x} \in \Omega$, where $w(\mathbf{x})$ is a weighting function and ρ a norm with its scale parameters $\boldsymbol{\sigma}$. To solve equation (1) Lucas and Kanade applied the least square estimator i.e. $\rho(y) = y^2$. In current applications [5] a pyramidal implementation and for each level an iterative scheme in a Newton-Raphson fashion is applied, so that:

$$\Delta \mathbf{d}^i = \underbrace{\left[\sum_{\Omega} \nabla I(\mathbf{x}) \cdot \nabla I(\mathbf{x})^T \right]^{-1}}_{\mathbf{G^{-1}}\text{(inv. gradient matrix)}} \cdot \underbrace{\left[\sum_{\Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}) \right]}_{\mathbf{b}^{i-1}\text{(mismatch vector)}} \tag{2}$$

denotes the incremental motion vector and:

$$\mathbf{d}^i \leftarrow \mathbf{d}^{i-1} + \Delta \mathbf{d}^i \tag{3}$$

denotes the iterative update or alignment for each motion vector $\mathbf{d}$ and is applied to each pyramidal level. While the gradient matrix is constant for each iteration, the mismatch vector has to be updated at each step by warping the second frame, so that $I_t^{i-1}(\mathbf{x}) = I(\mathbf{x}, t) - I(\mathbf{x} + \mathbf{d}^{i-1}, t+1)$. The computational cost of the Lucas Kanade method for each level is than given by $O(n \cdot i \cdot N^2)$, with N^2 the number of pixels in Ω, n the number of motion vectors to compute and i the number of iterations per motion vector.

To achieve subpixel accuracy an interpolation kernel is applied to estimate the subpixel values of the mismatch vector and the gradient matrix. As described by Rav-Acha and Peleg [11], it is possible to update the mismatch vector without warping the second image in the subpixel domain for each iteration, because the subpixel values only depend on the interpolation kernel and its adjacent pixels.

In this article the interpolation kernel is to be constrained as bilinear which is the base of the following lemma:

Lemma 1. *Let ϵ_x and ϵ_y be the decimal fraction with $\lfloor \mathbf{d}^{i-1} \rfloor + (\epsilon_x, \epsilon_y)^T = \mathbf{d}^{i-1}$ and $\mathbf{x}_{00} \in \mathbb{Z}^2$ the respective integer value of the position $\lfloor \mathbf{x} + \lfloor \mathbf{d}^{i-1} \rfloor \rfloor$ with its adjacent pixel positions $\mathbf{x}_{01}, \mathbf{x}_{10}, \mathbf{x}_{11} \in \mathbb{Z}^2$. Assuming that $\epsilon_x, \epsilon_y \in [0, 1)$ and the subpixel intensity values $I(\mathbf{x}, t), I(\mathbf{x}, t+1)$ are estimated with the bilinear interpolation, than equation* (2) *could be formulated as the following system of bilinear equations:*

$$\boldsymbol{\delta}^i(\epsilon_x, \epsilon_y) = \epsilon_x \epsilon_y \mathbf{a}_1 + \epsilon_x \mathbf{a}_2 + \epsilon_y \mathbf{a}_3 + \mathbf{a}_4, \tag{4}$$

with $\mathbf{a}_k, \boldsymbol{\delta}^i \in \mathbb{R}^2$.

Proof. To solve equation (2), note that the gradient matrix is fixed and the bilinear interpolation is applied to the mismatch vector $\mathbf{b}^{i-1}$. Using the following auxiliary variables:

$$\begin{aligned}
\mathbf{c}_1 &= \mathbf{c}_3 + \sum_{\Omega} \nabla I(\mathbf{x}) \cdot (I_t^{i-1}(\mathbf{x}_{11}) - I_t^{i-1}(\mathbf{x}_{10})) \\
\mathbf{c}_2 &= \mathbf{c}_4 - \sum_{\Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}_{10}) \\
\mathbf{c}_3 &= \mathbf{c}_4 - \sum_{\Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}_{01}) \\
\mathbf{c}_4 &= \sum_{\Omega} \nabla I(\mathbf{x}) \cdot I_t^{i-1}(\mathbf{x}_{00})
\end{aligned} \tag{5}$$

we can insert the inverse gradient matrix that is a coupled term between the two components of the mismatch vector:

$$\mathbf{a}_k = \mathbf{G^{-1}} \cdot \mathbf{c}_k. \tag{6}$$

Following equation (5) and (6) it is obvious that equation (2) could be formulated with equation (4) and $\Delta \mathbf{d}^i = \boldsymbol{\delta}^i$. □

Lemma 1 proves that the convergence behavior of the iterative Lucas Kanade method at the subpixel range is only depended on four adjacent intensity values that are constant for $\epsilon_x, \epsilon_y \in [0, 1)$. This methodology could be extended for the RLOF by the following theorem since the shrinked Hampel norm used by the RLOF is composed of quadratic functions which are important since the derivative of the norm determines the form of the mismatch vector.

Theorem 1. *Let y of $\rho(y, \boldsymbol{\sigma})$ be fixed for $\epsilon_x, \epsilon_y \in [0, 1)$ and the subpixel intensity values $I(\mathbf{x}, t), I(\mathbf{x}, t+1)$ to be estimated by bilinear interpolation, then $\Delta \mathbf{d}^i \in \mathbb{R}^2$ estimated by the RLOF method depends on the following system of bilinear equations:*

$$\boldsymbol{\delta}^i(\epsilon_x, \epsilon_y) = \epsilon_x \epsilon_y \mathbf{a}_1 + \epsilon_x \mathbf{a}_2 + \epsilon_y \mathbf{a}_3 + \mathbf{a}_4, \tag{7}$$

with $\mathbf{a}_k, \boldsymbol{\delta}^i \in \mathbb{R}^2$.

	Dimetrodon		Grove2		Grove3		Hydrangea		RubberWhale		Urban2		Urban3		Venus	
	AEE	η	AEE	η	AEE	η	AEE	η	AEE	η	AEE	η	AEE	η	AEE	η
RLOF	0.11	99.3	0.17	95.6	0.52	86.0	0.24	92.8	0.19	94.8	0.30	88.3	0.42	83.0	0.30	92.4
BERLOF	0.13	99.4	0.20	94.6	0.63	82.6	0.29	93.6	0.24	96.9	0.40	89.2	0.49	85.1	0.37	93.9
PLK	0.13	96.7	0.24	96.1	0.72	88.0	0.34	92.5	0.27	86.3	0.43	88.8	0.54	86.1	0.40	91.5

Table 1. Results of the Middlebury training sequences for sparse motion estimation.

Proof. According to [7] the incremental motion vector $\Delta\mathbf{d}^i$ is defined as:

$$\Delta\mathbf{d}^i = \mathbf{G}_{RLOF}^{-1} \cdot \mathbf{b}_{RLOF}^{i-1} \tag{8}$$

where $\mathbf{G}_{RLOF}^{-1}$ is fixed and the bilinear interpolation is applied to the mismatch vector.

$$\begin{aligned}
\mathbf{b}_{RLOF}^{i-1} &= \sum_{\Omega_1 \in \Omega} \nabla I(\mathbf{x}) \cdot \Big((1-\epsilon_x)(1-\epsilon_y) I_t^{i-1}(\mathbf{x}_{00}) \\
&+ \epsilon_x \epsilon_y I_t^{i-1}(\mathbf{x}_{11}) + (1-\epsilon_x)\epsilon_y \cdot I_t^{i-1}(\mathbf{x}_{01}) \\
&+ \epsilon_x (1-\epsilon_y) \cdot I_t^{i-1}(\mathbf{x}_{10}) \Big) \\
&+ \sum_{\Omega_2 \in \Omega} \frac{\sigma_1}{\sigma_1 - \sigma_2} \nabla I(\mathbf{x}) \cdot \Big(\epsilon_x \epsilon_y I_t^{i-1}(\mathbf{x}_{11}) \\
&+ \dots \\
&- sign(I_t^{i-1}(\mathbf{x}))\sigma_2 \Big)
\end{aligned} \tag{9}$$

with Ω_1 denoting the subset Ω fulfilling $|I_t^{i-1}(\mathbf{x})| \leq \sigma_1$ and Ω_2 the subset of pixels fulfilling $\sigma_1 < |I_t^{i-1}(\mathbf{x})| < \sigma_2$. It is obvious that equation 9 could be transformed such as equation 5 and 6. □

Note, the approximation of setting $y = I_t^{i-1}(\mathbf{x})$ of $\rho(y, \boldsymbol{\sigma})$ to be constant is needed to transfer equation 8 into a system of bilinear equations. The solution of the iterative scheme (3) is determined by one of the two zeros $\epsilon_{x0_{[1,2]}}, \epsilon_{y0_{[1,2]}}$ of the system of bilinear equations (4). A zero is a valid candidate for the limit of equation (3), since it fulfills the necessary break condition:

$$\Delta\mathbf{d}^i = \boldsymbol{\delta}^i(\epsilon_{x0_{[1,2]}}, \epsilon_{y0_{[1,2]}}) \stackrel{!}{=} (0,0)^T \tag{10}$$

$\forall \epsilon_{x0_{[1,2]}}, \epsilon_{y0_{[1,2]}} \in [0,1)$. To ensure equation (3) to converges to a zero, $\boldsymbol{\delta}^i$ has to be strictly monotonically decreasing at the candidate position. That implies the following sufficient conditions:

$$\frac{\partial \delta_u^i(\epsilon_x, \epsilon_y)}{\partial \epsilon_x} < 0 \,, \frac{\partial \delta_v^i(\epsilon_x, \epsilon_y)}{\partial \epsilon_y} < 0 \tag{11}$$

with $\boldsymbol{\delta}^i = (\delta_u^i, \delta_v^i)^T$. If only one zero fulfills the necessary and the sufficient condition, the iterative scheme will be solved directly by the following analytical solution:

$$\mathbf{d}^i = \left\lfloor \mathbf{d}^{i-1} \right\rfloor + (\epsilon_{x0}, \epsilon_{y0})^T. \tag{12}$$

Otherwise the motion vector will be updated with the accelerated iterative displacement $\boldsymbol{\delta}^i(\epsilon_x, \epsilon_y)$ similar to [11].

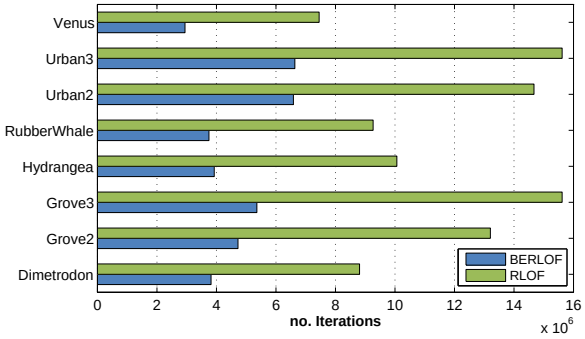

Fig. 2. Performance evaluation of the RLOF and BERLOF by comparing the total number of iterations for the Middleburry dataset.

3. EVALUATION

The evaluation of the proposed Robust Local Optical Flow by means of bilinear equations (BERLOF) has been performed with the Middlebury [1] and the KITTI [6] optical flow datasets. The performance has been measured in terms of runtime and accuracy for sparse motion vector fields. Therefore, we integrate each algorithm into a feature tracking framework, i.e. features are selected with the FAST [14] detector, tracked and validated by means of the forward and backward confidence. We compare the BERLOF with the RLOF available at `http://www.nue.tu-berlin.de/menue/forschung/projekte/rlof/` and the OpenCL OpenCV 2.4.3 implementation of the pyramidal Lucas Kanade (PLK) method available at `http://opencv.org/`. All methods are implemented by using OpenCL and run on a NVIDIA 480 GTX GPU. For each method we use the same basic configuration parameters, i.e. 3 pyramid levels, 15×15 region size Ω, the convergence criteria are set to 20 maximal iterations, $\epsilon = 0.1$ and the confidence threshold to 0.5. The norm parameter of the BERLOF and the RLOF are set to $\boldsymbol{\sigma} = (8, 50)^T$ and the small region size of the RLOF was set to 7×7.

As stated in Section 2 the motion vector $\mathbf{d}^i$ can be computed directly, if only one zero fulfills the necessary (10) and sufficient (11) condition. Figure 2 indicates in how many cases these conditions are valid. The plot illustrates the total number of iterations. On average the BERLOF computed 40% of the total iterations used by the RLOF. However, these measures are taken from the dense motion estimates for the

Rank	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	RLOF		code	**3.18 %**	**3.43 %**	1.0 px	1.2 px	14.76 %	0.488 s	GPU @ 700 Mhz (C/C++) GeForce GTX 680	
T. Senst, V. Eiselein and T. Sikora: Robust Local Optical Flow for Feature Tracking. TCSVT 2012.											
2	BERLOF			3.37 %	3.66 %	1.0 px	1.2 px	15.26 %	0.231 s	GPU @ 700 Mhz (C/C++) GeForce GTX 680	
Anonymous submission											
3	PR-Sf+E			4.08 %	7.79 %	**0.9 px**	1.7 px	100.00 %	200 s	4 cores @ 3.0 Ghz (Matlab + C/C++)	
Anonymous submission											
4	PCBP-Flow			4.08 %	8.70 %	0.9 px	2.2 px	100.00 %	3 min	4 cores @ 2.5 Ghz (Matlab + C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Robust Monocular Epipolar Flow Estimation. CVPR 2013.											

Fig. 3. Screenshots of the KITTI benchmark for sparse evaluation by time of submission (05.2013); More details are available at `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=flow&eval=est`.

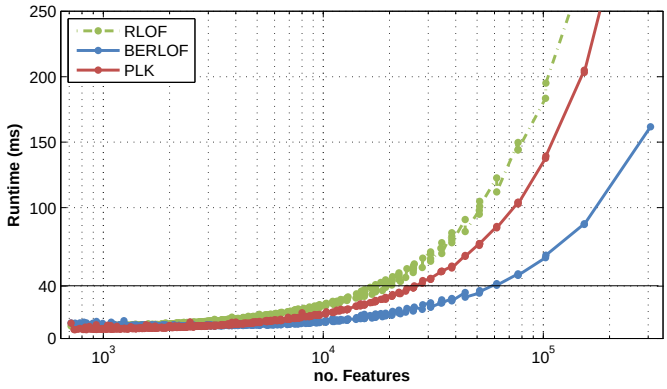

Fig. 4. Runtime comparison of RLOF, PLK and the BERLOF method for the Grove3 sequence of the Middleburry dataset (GPU OpenCL implementations). The solid line denote the 25 fps border by which the BERLOF is able to estimate 61.440, the PLK 25.920 and the RLOF 20.540 features.

Middlebury training sequences. As stated in Section 1 the scope of applications using local optical flow tracker is in computing sparse motion information in real-time. To cover a wide range of possible applications, we measured the runtime of each algorithm for varying number of features to track. Figure 4 shows the runtime comparison for the Grove3 sequence (resolution 640×480). In terms of runtime the BERLOF outperforms the RLOF and the OpenCV's PLK implementations. The BERLOF is able to compute 137.04% more motion vectors than the OpenCV PLK implementation in real-time (25fps). The results show that the theoretical improvement of the proposed partial analytical solutions could be implemented into an efficient feature tracking method that is beneficial for a huge set of applications.

The accuracy of the BERLOF is evaluated in Table 1. In addition to the average endpoint error (AEE), we provide the tracking efficiency (η) [7] to identify the rejected motion estimates. Table 1 shows that the BERLOF is less accurate than the RLOF but more precise than the PLK method. Unlike RLOF the BERLOF is not implementing the region adaption in order improve the performance the parallelised BERLOF. Thus the error rate at motion boundaries has been increased. However the difference in accuracy related to the improved runtime is marginal.

In addition, the BERLOF method has been submitted to the KITTI benchmark. The dataset consist of high-resolution (1241×376) grayscale image sequences captured form a car driving around the mid-size city of Karlsruhe. The benchmark is able to deal with dense and sparse motion estimation methods. If, as in our case, a sparse motion vector field has been submitted, then the dense evaluation is done by interpolating the dense motion vector field. A full description of the dataset and the ranking methodologies can be found at [6]. We adapt the parameter configuration, since the dataset provides a higher resolution. Details are published on the dataset website. By the time of submission of this paper, the proposed method is being ranked in the second position behind the RLOF method by evaluating the estimated pixels only. Figure 3 shows a snapshot for the sparse mode of the overall performance of the top 4 algorithms. As shown by the evaluation with the Middleburry dataset, the BERLOF does not obtaining the full accuracy of the RLOF, but it is able to reduce the runtime from 600ms to 380ms for a sparse implementation.

4. CONCLUSION

In this paper we presented a mathematical formulation of the iterative solution of the Robust Local Optical Flow method which allows us to analytically and directly estimate a subset of motion vectors. We proved that the computation of incremental motion vectors can be reformulated as a system of bilinear equations in the subpixel domain. Furthermore we show that a determined set of zeroes correspond to the solution of the baseline iterative scheme. The experimental evaluation supported our argumentation. For the Grove3 sequence of the Middlebury dataset the BERLOF is able to compute 137.04% more motion vectors than the OpenCV PLK on a GPU at 25fps. The sparse evaluation of the KITTI dataset shows that the runtime of the BERLOF tracker is 63.3% in relation to the RLOF tracker and a similar accuracy.

5. REFERENCES

[1] Simon Baker, Daniel Scharstein, J.P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski, "A database and evaluation methodology for optical flow," in *International Conference on Computer Vision (ICCV 2007)*, 2007, pp. 1–8.

[2] Berthold K.P. Horn and Brian G. Schunck, "Determining optical flow," *Artifical Intelligence (AI 1981)*, vol. 17, pp. 185–203, 1981.

[3] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI 1981)*, 1981, pp. 674–679.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV 2004)*, 2004, pp. 25–36.

[5] Carlo Tomasi and Takeo Kanade, "Detection and tracking of point features," Tech. Rep., CMU-CS-91-132, CMU, 1991.

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR 2012)*, 2012, pp. 3354–3361.

[7] Tobias Senst, Volker Eiselein, and Thomas Sikora, "Robust local optical flow for feature tracking," *Transactions on Circuits and Systems for Video Technology (TCSVT 2012)*, vol. 22, no. 9, pp. 1377–1387, 2012.

[8] Sudipta N. Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc, "Gpu-based video feature tracking and matching," Tech. Rep., 06-012, UNC Chapel Hill, 2006.

[9] Christopher Zach, David Gallup, and Jan-Michael Frahm, "Fast gain-adaptive klt tracking on the gpu," in *Visual Computer Vision on GPUs Workshop (CVGPU 2008)*, 2008.

[10] Tobias Senst, Volker Eiselein, Michael Pätzold, and Thomas Sikora, "Efficient real-time local optical flow estimation by means of integral projections," in *International Conference on Image Processing (ICIP 2011)*, 2011, pp. 2393–2396.

[11] Alex Rav-Acha and Shmuel Peleg, "Lucas-kanade without iterative warping," in *International Conference on Image Processing (ICIP 2006)*, 2006, pp. 1097–1100.

[12] Yeon ho Kim, Aleix M. Martinez, and Avi C. Kak, "A local approach for robust optical flow estimation under varying illumination," in *British Machine Vision Conference (BMVC 2004)*, 2004.

[13] Jean-Marc Odobez and Patrick Bouthemy, "Robust multiresolution estimation of parametric motion models," *Visual Communication and Image Representation (JVCIR 1995)*, vol. 6, pp. 348–365, 1995.

[14] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV 2006)*, 2006, pp. 430–443.