MARKERLESS CLOSED-LOOP PROJECTION PLANE TRACKING FOR MOBILE PROJECTOR-CAMERA SYSTEMS

Niklas Gard, Peter Eisert

Humboldt University and Fraunhofer HHI, Berlin, Germany

ABSTRACT

The recent trend towards miniaturization of mobile projectors is allowing new forms of information presentation and interaction. Projectors can easily be moved freely in space either by humans or by mobile robots. This paper presents a technique to dynamically track the orientation and position of the projection plane only by analyzing the distortion of the projection by itself, independent of the presented content. It allows distortion-free projection with a fixed metric size for moving projector-camera systems. To do this, an optical flow-based model is extended to the geometry of a projector-camera unit. Solving an overdetermined system of equations on pixel level leads to the pose offset between two images. In order to reach a high invariance to illumination changes we use adaptive edge images. Image pyramids allow a fast pose estimation. Because of the global optimization, there is no dependence of the availability of local feature points.

***Index Terms*—** Augmented reality, Optical flow, Projector-camera systems, Closed-loop registration

1. INTRODUCTION

A combined system of a projector and a camera allows to compensate captured distortions of a projected image to project it geometrically undistorted on arbitrary geometries. Information about the scene can be acquired by determining the geometric relation between the projected image and the image captured by the camera, as it is done for example in structured light systems. In a so called closed-loop approach [1] the projection is used to present information, e.g. to give a new appearance to an object in an augmented reality scenario. At the same time, a self-correcting mechanism is ensuring that the projection stays in place by estimating how emerging distortions can be compensated and by creating a compensating image to project next. Recently, a pose estimation system for geometric objects has been demonstrated using this principle [2].

The demand for these kind of systems is especially high today because projectors can be built as lightweight handheld devices and even an integration in smartphones is possible. Furthermore laser scanning projectors or projectors with LCoS panel and laser light source are focus free [3] and a sharp image can be projected independent of the distance to the projection plane.

In this paper, we present an automatic tracking system for distance and orientation of planar projection planes in a closed-loop approach, which can be seen as a preliminary stage for more complex tracking tasks with arbitrary geometry following the same principle. It differs from other approaches in a way that neither the four corner points of the image [4] nor discrete feature points are used to match between projected image and the image of its reflection in order to estimate a homography. The outer corner points are not visible if the projected image has a dark background (e.g. if only text is projected). A drawback of feature points as used in [5, 2] is that the projected image has to provide enough points to match and that lighting conditions affect the quality of corresponding points. Furthermore, a radiometric calibration of the projector and the camera response is needed. In contrast to other publications [6], we do not align the projection to a printed element in the projection plane but only to its own reflection resulting in a high flexibility without additional markers.

Target applications include information presentation techniques for mobile robots similar to [7, 8] allowing a robot to project an interactive user interface or information with a fixed metric size on an arbitrary plane without keystone effects, even while the plane or the robot is moving. Also, for interactive applications with a moving mobile projector as in [9] or as stabilization for wearable devices [10] our approach is promising.

In the next section, the mathematical foundations of the motion model and its relation to the optical flow equation is outlined. Section 3 covers implementation details as for example the usage of edge images or the integration of a robust estimator. In section 4, the approach is evaluated using real and synthetic images.

2. PROJECTOR-BASED OPTICAL FLOW

It is assumed that the projector is oriented towards a planar surface and that the intrinsic parameters of the projector and the camera are known as well as the extrinsic parameters between the devices. The pinhole camera model is used to describe the camera as well as the projector, which is seen as an inverted camera. A plane is defined by the normal vector $\mathbf{n}$ and a vector $\mathbf{t} = [0, 0, t_z]^T$ from the camera center to the intersection point of the camera's optical axis with the plane. This results in three degrees of freedom to be estimated, the distance to the camera t_z and two angles of rotation r_x and r_y.

2.1. Motion model

For each pixel (X_1, Y_1) in the camera image, a viewing ray is defined by

$$\mathbf{v} = [-\frac{X_1 - u_x}{f_x}, \frac{Y_1 - u_y}{f_y}, -1]^T \tag{1}$$

with f_x, f_y being the scaled horizontal and vertical focal length and u_x, u_y the pixel position of the principal point. The intersection point p between a given plane and the viewing ray is then

$$\mathbf{p} = \mathbf{v}\frac{\mathbf{t}^T\mathbf{n}}{\mathbf{v}^T\mathbf{n}} \quad . \tag{2}$$

Using the extrinsic parameters of the camera-projector system allows to transform the point to the projector coordinate system, where it can be back projected to the image plane of the projector leading to the viewing ray $\mathbf{v}'$ of the projector.

The final version is available at https://doi.org/10.1109/ICIP.2018.8451038

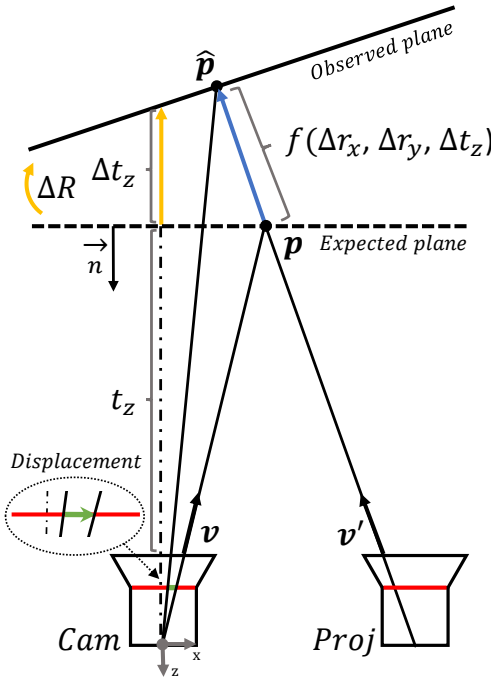

Fig. 1: Geometry of a projector-camera system. The expected plane differs from the observed plane by a certain transformation ($\Delta\mathbf{R}$, $\Delta\mathbf{t_z}$). The displacement in image space is related to $\mathbf{v}'$ and the result of f.

If we assume that the movement of the projection plane is small, the change of orientation can be approximated by a linearized rotation matrix

$$\mathbf{\Delta R} = \begin{pmatrix} 1 & 0 & \Delta r_y \\ 0 & 1 & -\Delta r_x \\ -\Delta r_y & \Delta r_x & 1 \end{pmatrix} \tag{3}$$

with two degrees of freedom. The translation offset Δt_z is the difference added to t_z to hit the new plane. This results in a movement of p by $f(\Delta t_z, \Delta r_x, \Delta r_y)$ in direction of the projectors light beam. It will intersect with the new plane at

$$\hat{\mathbf{p}} = \mathbf{p} + \mathbf{v}' f(\Delta t_z, \Delta r_x, \Delta r_y) \tag{4}$$

with $\mathbf{v}'$ being the view vector of the projector in the camera coordinate system. The function f follows the general formula of a line-plane intersection and is

$$f(\Delta t_z, \Delta r_x, \Delta r_y) = \frac{(\mathbf{\Delta t} - (\mathbf{p} - \mathbf{t}))^T \mathbf{\Delta R n}}{\mathbf{v'}^T \mathbf{\Delta R n}} \quad . \tag{5}$$

In conclusion, this means that the new pixel coordinate (X_2, Y_2) of the point in the camera image is calculated by

$$X_2 = -f_x \frac{\hat{p_x}}{\hat{p_z}} + u_x \quad Y_2 = f_y \frac{\hat{p_y}}{\hat{p_z}} + u_y \tag{6}$$

if the position of the projection plane is rotated or translated with a given offset. Fig. 1 shows how the movement from (X_1, Y_1) to (X_2, Y_2) is related to v' and to the result of f.

2.2. Extension of optical flow

In this section, the motion of the projection plane will be estimated by evaluating spatial and temporal image gradients [11]. This leads to a set of linear equations [12] to calculate the pose offset values $\Delta t_z, \Delta r_x, \Delta r_y$ from a known initial pose. The image I_1 of the actual projection is captured by the camera. Image $\hat{I}_1$ of the projection on a known initial plane is either the previous image in a tracking sequence or synthetically generated. A motion compensation minimizes the displacement error $(u_m, v_m) = (X_1 - X_2, Y_1 - Y_2)$ between $\hat{I}_1$ and I_1. By using the first order Taylor expansion the projection of a point to the sensor can be expressed as

$$\begin{aligned} \frac{\hat{p_x}}{\hat{p_z}} &\approx \frac{p_x}{p_z} + \frac{b}{p_z(\mathbf{n}^T\mathbf{v}')}(\Delta r_x o_y - \Delta r_y o_x + \Delta t_z n_z) \\ \frac{\hat{p_y}}{\hat{p_z}} &\approx \frac{p_y}{p_z} + \frac{c}{p_z(\mathbf{n}^T\mathbf{v}')}(\Delta r_x o_y - \Delta r_y o_x + \Delta t_z n_z) \end{aligned} \tag{7}$$

with

$$b = v'_x - v'_z \frac{p_x}{p_z} \quad c = v'_y - v'_z \frac{p_y}{p_z}$$

and

$$o_x = p_x n_z - (p_z - t_z) n_x \quad o_y = p_y n_z - (p_z - t_z) n_y$$

The general optical flow constraint equation [11] involving the partial derivatives of $\hat{I}_1$ in horizontal and vertical direction and the temporal difference is

$$\frac{\partial \hat{I}_1}{\partial \hat{X}_1} u_m + \frac{\partial \hat{I}_1}{\partial \hat{Y}_1} v_m \approx \hat{I}_1 - I_1 \quad . \tag{8}$$

Combining Eq. (8) and (7) results in a linear equation with three unknown motion parameters

$$a_0 \Delta r_x + a_1 \Delta r_y + a_2 \Delta t_z = \hat{I}_1 - I_1 \tag{9}$$

which is formulated for each pixel corresponding to the projection. The set of equations forms an overdetermined system which can be solved in a least squares sense or with a robust estimator. The three coefficients are defined as

$$a_0 = -o_y d \quad a_1 = o_x d \quad a_2 = -n_z d \tag{10}$$

with

$$d = \frac{1}{p_z(\mathbf{n}^T\mathbf{v}')} \left(\frac{\partial \hat{I}_1}{\partial \hat{X}_1} f_x b + \frac{\partial \hat{I}_1}{\partial \hat{Y}_1} f_y c \right) \quad .$$

When the system of equations is solved, the linearized rotation $\mathbf{\Delta R}$ is concatenated with the previous rotation. The translation offset Δt_z can simply be added to the previous value. Additionally, a smoothness term is incorporated to reduce jittering between frames following the same principle as in other optical flow applications.

3. CLOSED-LOOP PLANE TRACKING

In our test setup a *Basler acA2040-90um* camera is used and the images are downsampled to a resolution of 1024x1024 pixel. The employed projector is a *Laser Beam Pro C200* with a resolution of 1366x768 pixel and 200 lumen, which is used because of its ability to be focus free in a large range (from about 90cm to 4.5m). Both devices are mounted on a rigid holding with a baseline of approximately 8 cm. The system is calibrated based on the toolbox presented in [13]. For the projector we assume that the distortion coefficients are zero.

(a) No alignment

(b) Corrected result

Fig. 2: Model-based pose estimation. In (a) the observed image is not aligned with the expected image, highlighted by the green frame. After our correction (b), the expectation and the observation are aligned.

3.1. Initialization

A coarse initial estimation of the plane pose is needed to ensure that the optimization algorithm converges to the right pose. We achieve this initial estimation by projecting a known binary pattern twice, as a positive and as an inverted image. The pattern shows five points and four lines. Applying a threshold on the difference image is masking the points. The center of a binary dot in the camera image is set in correspondence to the center of a dot in the projected image. A triangulation leads to 3D-points as it is done in a stereo-camera setup and a plane fitting through these points is performed. Initialization is canceled if the points can not be matched in the correct order or if the reconstructed points do not lie on a plane. In this case, the projector-camera system is either not pointing towards a planar surface or the current lighting conditions/camera settings would not allow a tracking.

3.2. Algorithm description

The algorithm is based on a frame-to-frame approach using the last pose as start value for the estimation of a new pose. Following the principle of analysis-by-synthesis, a synthetic image is created and compared to the camera image. This allows the simulation of appearance of the projected image with the last parameters and prevents a cumulative drifting error. Fig. 2a shows a misalignment between expectation and observation. Accordingly the projection is geometrically distorted. In Fig. 2b expectation and observation are aligned and the distortion is eliminated.

The three derivative images in horizontal direction, in vertical direction and in time domain are computed from the synthetic and the camera image. As there is no simulation of illumination in the rendered images, a direct comparison to the camera image will introduce problems due to different pixel intensities in the images. Instead, first a binary edge image is built by thresholding the gradient magnitude of each pixel, found by using the horizontal and vertical Sobel operator [14]. While the threshold for the synthetic image remains fixed, the threshold for the camera image is adjusted dynamically to compensate for different lighting conditions. In the image area, where the projection is assumed, the proportion of white and black pixels should be about the same for both images. By subdividing the image into $n \cdot m$ tiles and by calculating separate adaptive thresholds for each tile, the method becomes more robust against noise and occlusions. Secondly, blurring the binary images with a Gaussian filter produces smooth gradients which are needed for optical flow estimation.

Image pyramids are used to compensate big movements between frames. We found four pyramid levels with a resolution of 128x128 pixels in the smallest level to be a reasonable choice. After each iteration, a warped image is generated and used as synthetic input for the next iteration. This is done by simulating the projection of the image in the same orientation on a moved target. For each pixel in the synthesized image, the viewing ray from the projector through the new plane is calculated (see Eq. (2)) and intersected with the previous plane. Back projecting this 3d point to the camera image provides the lookup position to read out the grayscale value for the new image from the old image.

The linear equations in the system are weighted following the iteratively reweighted least squares scheme [15] to cope with outliers in the data. As proposed in [16], we use the robust Charbonnier penalty $p(x) = \sqrt{x^2 + \epsilon^2}$ with a small ϵ (e.g. $\epsilon = 0.001$) to calculate the weights.

4. RESULTS

This section covers tests with simulated synthetic images and with real images in a live application. For tests with synthetic data, objective results are provided, proving the general viability of the algorithm. In tests with real image data, the projected images are judged subjectively. The experiments show how external influences affect the tracking.

4.1. Synthetic data

In the test with synthetic data, a moving projection plane respectively a moving projector-camera system is simulated. For each frame an observed distorted image is generated by warping the expectation image to a specific pose with the method described in section 3.2. We use three test images for the evaluation (see Fig. 3), first (**I1**) a regular photography, second (**I2**) a self repetitive pattern and third (**I3**) a simple user interface providing sharp edges. At the start of a sequence the simulated projection plane is placed in a distance of 1.8m from the camera and is rotated by -45° around its y-axis (pointing up) and by -22.5° around its x-axis, relative to the observing camera. The projected image is square-shaped with a resolution of $1024x1024$ and the simulated projection has a side length of 50cm. In total six test sequences (**T1**-**T6**) are evaluated. In **T1** the plane is rotated by one degree around the its x-axis, by half a degree around its y-axis and is moved 1cm towards the camera each frame until the distance to the camera is 90cm. In **T2** the step size between the frames is five times bigger, in test **T3** the step size is ten times bigger. In **T4** all gray scale values in the observation are multiplied with a factor of 1.25 to simulate additional ambient lightning. In **T5**

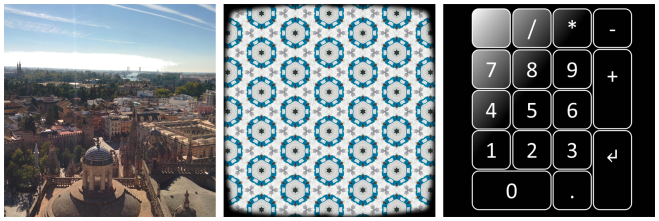

Fig. 3: Test images used for evaluation: **I1** a photography, **I2** a self repetitive pattern and **I3** a simple user interface (LTR).

the observed image is filtered with a Gaussian kernel of size $7x7$ to simulate the projection plane to be out of focus in the camera image. Lastly, in **T6** about a quarter of the observed image is covered by two white circles to simulate an occlusion. The last three tests use the step size of **T2**.

In table 1 the mean absolute error values are given over all frames for each test sequence with each image. The results of test **T1** - **T3** show that large movements between frames can be compensated. Also the ambiguous structure of image **I2** does not lead to wrong estimations. Global changes of the image brightness in **T4** slightly enlarge the error for image **I1** and **I2** most likely because of clipping artifacts. Still an error in this magnitude would not be visible in practice. If the image is blurred as in **T5**, the error increases and also a slight jittering between frames arises. Reason for this behavior is that the Sobel edges become wider in image space which leads to a small misalignment resulting in the observed pose error. Due to the model-based optimization a drift of the pose is prevented, even if the estimated pose is erroneous. The introduced occlusion in **T6** has a distinctively smaller influence on the estimation error.

	I 1			**I 2**			**I 3**		
Test	r_x	r_y	t_z	r_x	r_y	t_z	r_x	r_y	t_z
T 1	0.27	0.50	0.20	0.08	0.15	0.08	0.30	0.23	0.09
T 2	0.21	0.13	0.13	0.06	0.14	0.08	0.21	0.18	0.08
T 3	0.35	0.15	0.20	0.06	0.11	0.10	0.22	0.20	0.09
T 4	0.80	0.13	0.39	0.40	0.32	0.07	0.19	0.21	0.08
T 5	2.67	4.04	0.80	0.43	0.80	0.30	1.57	6.95	0.85
T 6	0.57	0.40	0.20	0.34	0.19	0.26	0.34	0.21	0.12

Table 1: Results of six tests with three images. The values under r_x, r_y, t_z are the mean absolute errors over a sequence. Rotation errors are given in degrees, translation errors are given in cm.

4.2. Real data

The presented procedure has been tested with real data in an interactive software, which implements the closed-loop approach. A fullscreen OpenGL render window displays predistorted images on the projector. To create an image, the intrinsic parameters of the projector, the pose parameters of the plane and the extrinsic parameters of the system are used in a shader.

Currently the correction is calculated with about five to ten frames per second. The processing time can be speeded up by forming the system of equations directly on the GPU, which is feasible because each pixel is analyzed separately. Also the warping and the edge filtering can be parallelized. Special effort has to be put in the synchronization of the projector and the camera. A new orientation

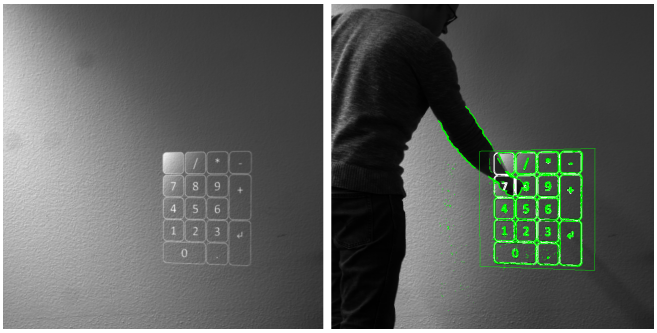

Fig. 4: The projection is not visibly reorientated if a partial occlusion appears in an AR-scenario while the tracking is active. In the left image the green frame is the bounding box of the projected image. The alignment between the edges in the camera image and the edges of the rendered image are highlighted (right).

can only be estimated if the projection with the last parameters is visible in the camera image. A possible solution is to use a HDMI-to-VGA adapter and to tap the V-Sync of the projector to trigger the camera [2].

During tests with various images and different projector orientations, distortions could be corrected even for steep angles to the projection plane. The edge length of the projection remains constant while the projector moves in front of the plane. The local thresholding turns out to compensate even big illumination variations. Even if the projection is hardly visible in the camera image, the algorithm converges to a rectified image in most cases. In practice, the proposed initialization can be skipped if the initially assumed plane distance is not completely different from the real distance. For a true distance of one and a half meter variations of more than half a meter could be corrected. If different images are projected in a loop, small variations of the pose occur resulting in visible jittering. Here an additional temporal stabilization is needed to prevent the algorithm from converging to slightly different orientations.

In Fig. 4, the tracking runs in a simulated AR scenario in which a touch interaction is necessary. Occlusions of the projection do not have a visible influence on the pose estimation because of the stabilization due to the usage of a robust estimator and the local thresholding of the edge images.

5. CONCLUSION AND FUTURE WORK

In this paper, a closed-loop model-based tracking method for the orientation of a projection plane has been defined. The general principle has been tested with synthetic and real data and can be used in projector-camera systems to display e.g. user interfaces on arbitrary planes even while the device is moving. The results are integrated in an interactive software implementing the closed-loop principle and running with multiple frames per second. Since the equations are defined on a pixel level, the algorithm can easily be parallelized and ported to the GPU to allow true real-time performance. In future work, we plan to enhance the approach for pose estimation of three dimensional objects with arbitrary geometry.

6. ACKNOWLEDGMENT

This work is supported by the German Federal Ministry of Education and Research (EASY-COHMO project, grant no. 03ZZ0443E).

7. REFERENCES

[1] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay, "Shader lamps: Animating real objects with image-based illumination," in *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, London, UK, 2001, pp. 89–102.

[2] C. Resch, P. Keitler, and G. Klinker, "Sticky projections-a model-based approach to interactive shader lamps tracking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1291–1301, March 2016.

[3] K. M. Guttag and S. Hurley, "Laser + lcos projection-technology revolution," *Journal of the Society for Information Display*, vol. 20, no. 5, pp. 279–285, 2012.

[4] R. Sukthankar, R. G. Stockton, and M. D. Mullin, "Smarter presentations: Exploiting homography in camera-projector systems," in *ICCV*, 2001, pp. 247–253.

[5] H. Fuchs and T. Johnson, "Real-time projector tracking on complex geometry using ordinary imagery," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, June 2007.

[6] S. Audet, M. Okutomi, and M. Tanaka, "Direct image alignment of projector-camera systems with planar surfaces," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 303–310.

[7] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space," in *European Conference on Mobile Robots (ECMR)*, Sept 2015.

[8] R. S. Andersen, O. Madsen, T. B. Moeslund, and H. Ben Amor, "Projecting robot intentions into human environments," in *25th IEEE International Symposium on Robot and Human Interactive Communication*, New York, USA, Aug. 2016, pp. 294–301.

[9] K. D.D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler, "Sidebyside: Ad-hoc multi-user interaction with handheld projectors," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, USA, 2011, pp. 431–440.

[10] N. Sakata, F. Sato, T. Tominaga, and Y. Hijikata, "Extension of smartphone by wearable input/output interface with floor projection," in *Collaboration Technologies and Social Computing*, 2017, pp. 168–181.

[11] B.K.P. Horn and B. G. Schunck, "Determining optical flow," Tech. Rep., Cambridge, USA, 1980.

[12] E. G. Steinbach, P. Eisert, and B. Girod, "Model-based 3-d shape and motion estimation using sliding textures," in *Proceedings of the Vision Modeling and Visualization Conference (VMV)*, Nov 2001, pp. 375–382.

[13] D. Morenoand G. Taubin, "Simple, accurate, and robust projector-camera calibration," in *Proceedings of the International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, Washington, DC, USA, 2012, pp. 464–471.

[14] I. Sobel, "An isotropic 3 x 3 image gradient operator.," in *Machine vision for three-dimensional scenes*, 1990, pp. 376–379.

[15] Z. Zhang, "Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting," *Image and vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[16] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2432–2439.