Fast Projector-Driven Structured Light Matching in Sub-Pixel Accuracy using Bilinear Interpolation Assumption

Torben Fetzer[1], Gerd Reis[2] and Didier Stricker[1,2]

[1] University of Kaiserslautern
[2] German Research Center for Artificial Intelligence (DFKI)
{torben.fetzer, gerd.reis, didier.stricker}@dfki.de

Abstract. In practical applications where high-precision reconstructions are required, whether for quality control or damage assessment, structured light reconstruction is often the method of choice. It allows to achieve dense point correspondences over the entire scene independently of any object texture. The optimal matches between images with respect to an encoded surface point are usually not on pixel but on sub-pixel level. Common matching techniques that look for pixel-to-pixel correspondences between camera and projector often lead to noisy results that must be subsequently smoothed. The method presented here allows to find optimal sub-pixel positions for each projector pixel in a single pass and thus requires minimal computational effort. For this purpose, the quadrilateral regions containing the sub-pixels are extracted. The convexity of these quads and their consistency in terms of topological properties can be guaranteed during runtime. Subsequently, an explicit formulation of the optimal sub-pixel position within each quad is derived, using bilinear interpolation, and the permanent existence of a valid solution is proven. In this way, an easy-to-use procedure arises that matches any number of cameras in a structured light setup with high accuracy and low complexity. Due to the ensured topological properties, exceptionally smooth, highly precise, uniformly sampled matches with almost no outliers are achieved. The point correspondences obtained do not only have an enormously positive effect on the accuracy of reconstructed point clouds and resulting meshes, but are also extremely valuable for auto-calibrations calculated from them.

Keywords: Structured Light · Matching · Sub-Pixel · Linear · Consistent

1 Introduction

Structured light enables the determination of precise and dense point correspondences between a camera and a projector view. No features are required, making it applicable to a wide range of different object types. Often, sinusoidal patterns are projected to encode the scene in two directions. With the help of the deformed patterns, horizontal and vertical phase images are calculated for each camera view, that theoretically lead to a direct correspondence between each projector pixel and its position in the camera image. From these point correspondences, cameras and projector can be calibrated and a dense point cloud can be triangulated using the obtained camera matrices.

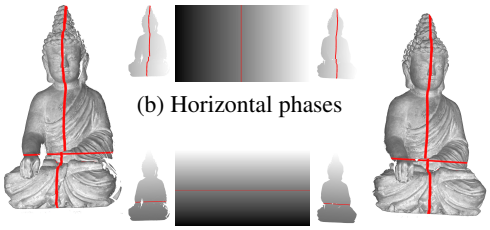


Fig. 1: Illustration of projector-driven matching of two cameras and a projector. Red lines visualize the encoding of a point by its phases.

Fig. 2: Resulting point clouds of FPDM with (b,c) and without (a) *TCC* and in addition with *ECC* (c).

In theory, a setup consisting of a projector as active device, holding the perfect phase, and a camera is sufficient for depth estimation. However, in many practical arrangements, several cameras, at least two, are used in addition to the projector. This is due to a much cleaner projective behavior of high quality cameras compared to most projectors. Since higher quality lenses are available, usually less distortions are caused. Also, most industrial cameras allow gamma correction to be disabled, which has a significant impact on assumptions in computer vision applications. Since this is not possible with affordable projectors, it is of considerable advantage to triangulate the point cloud with the camera information only. To cover the general case of any number of cameras, in this paper the situation with two cameras and one projector is considered. Thus, the procedure can be trivially extended to an arbitrary number of cameras.

The idea of projector-driven matching is to find suitable correspondences in the camera images for each projector pixel. In this way, the camera positions are transitively matched via the projector pixels. Figure 1 illustrates this procedure. (a) and (d) show the texture images of the two camera views. (b) and (c) show the corresponding horizontal and vertical phase images of the cameras and in the center of the projector. The red lines illustrate the unique encoding process of a pixel through the two phases. In the exact execution of dense matching a number of difficulties arises:

- Phase images are discrete samples of continuous functions. Therefore, there is usually no exact pixel-to-pixel mapping. Instead, it is very likely that matches lie between certain camera pixels.
- The topology of the pixels remains locally preserved during the projection process. Thus, certain conditions can be defined which must be fulfilled by the phases and met during the matching process in order to avoid noisy results.
- Matching is only a sub-step in 3D reconstruction and auto-calibration and should therefore be fast. The trivial procedure of searching optimal matches between all images is not practical at all. The procedure would be of quadratic complexity in terms of resolution, and with increasing camera resolutions this is very poor.

In the following, we develop a procedure that is extremely fast and can match any number of devices stably and consistently with sub-pixel accuracy. Each image pixel must be passed through exactly once resulting in a linear complexity.

2 Related Work

Matching is one of the main components in the field of 3D reconstruction. The goal is to find point correspondences as dense and precise as possible across the entire scene. Standard approaches search for suitable candidates along the epipolar lines and evaluate them according to their neighbors using suitable region descriptors [7]. This is a common approach, but requires a calibrated setup and can fail in many cases, as in uniform areas of the scene. If it is, in contrast, possible to create very precise matches without prior calibration information (e.g. [6]), modern auto-calibration methods, such as [4] and [5] allow to perform an exact calibration of the system directly from these matches, which makes such a computer vision system much more user-friendly and flexible. It also makes it suitable for a variety of other applications where pre-calibration is not possible, extremely tedious or problematic, since the setup may de-calibrate over time.

Common matching procedures without pre-calibration are based on SIFT features [8], which provide robust matches if sufficient object texture is available. Also, there are methods that do not only include appearance but also object geometry into the search [9]. However, both of them most likely fail in the case of very smooth uniform objects, which limits the applicability. To reconstruct untextured objects the structured light approach is a common tool. In [11] the authors use structured light information to handle large disparities in binocular matching. Similarly, in [12] the wrapped phase is used to refine the stereo matches. [14] shows how to get accurate dense matches using only the reconstructed phase. [3] introduces a sub-pixel matching for unsynchronized structured light, while for each match an energy is minimized by gradient descent. Matching based on peak calculation [15] and [1] also achieves sub-pixel accuracy but requires higher computational effort than the method presented.

In [2] a deep learning approach for structured light matching was proposed recently. It uses a Siamese network trained on a synthetic data set that expects rectified images, which is not suitable for arbitrary uncalibrated systems and auto-calibration. Also in [10] structured light and deep learning are combined to achieve good and exact matches. In [13] matching is even skipped and depth is directly calculated using deep learning.

3 Fast Projector Driven Matching (FPDM)

The task of fast projector-driven matching is to find corresponding positions in the camera phases for each integer projector pixel. Since this is usually not again an integer position, it must be estimated with sub-pixel accuracy. Figure 3 (a) illustrates this for the projector in the middle and camera images left and right. Everything at the sub-pixel level can only be described by the pixels in its environment, since no smaller information is available in an image. In order to compute the sub-pixel matches, it is therefore necessary to find integer camera pixels that span a quadrilateral (Figure 3, green pixels) that encloses the optimal sub-pixel match as closely as possible. From this quad, the sub-pixel match can then be interpolated in a subsequent step. The quadrilateral does not necessarily have to be square or rectangular, but should at least be convex. This constraint is fulfilled in the general case and only violated at regions with depth discontinuities. It ensures that the enclosed area can be described smoothly through its corners. In addition, there are certain consistency characteristics that should be met.

3.1 Matching Integer Pixel Quads

In a first step, best possible convex quads enclosing the sub-pixel match for each projector pixel are found in each camera image. Each camera pixel should only be processed once in order to maintain linear complexity. Therefore, for each projector pixel, we store the four corner points whose quadrilateral contains the optimal camera correspondence. An array four times the size of the projector resolution is needed as a buffer. Note that the projector image can be selected in any resolution as it is completely imaginary. The resulting density of the point cloud can be precisely controlled in this way. Practice has shown that the projector resolution should be selected in approximately the same order as the camera resolutions, since usually both cover about the same area of the scene.

In the following we assume horizontal and vertical phase images $\boldsymbol{\Phi}_H$ and $\boldsymbol{\Phi}_V$ of a camera with values in the interval $[0, 1]$. The phases run from left to right and from bottom to top according to the common coordinate axes. Similarly, the optimal projector phases run from 0 to 1 at a selected resolution (w_P, h_P). This is depicted in Figure 1.

For each camera pixel (x, y), the theoretical corresponding position in the projector image is uniquely given by the vertical and horizontal phase values $\boldsymbol{\Phi}_H(x, y)$ and $\boldsymbol{\Phi}_V(x, y)$. Therefore, a camera pixel (x, y) would theoretically match projector pixel

$$(\hat{x}_P, \hat{y}_P) = \big(\boldsymbol{\Phi}_H(x, y)w_P, \boldsymbol{\Phi}_V(x, y)h_\mathrm{P}\big), \tag{1}$$

which is not an integer value, as sought. Nevertheless, it is an approximate position and likely a lower and upper corner of the next integer projector pixels, which is the basic idea of the presented fast (linear) method.

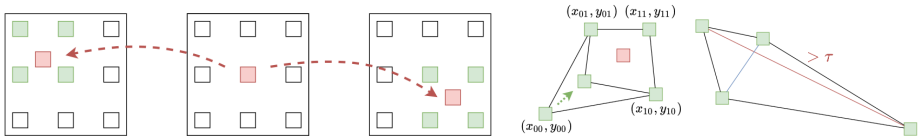
For each integer projector pixel (x_P, y_P) the vertices of the spanned matching quad in the camera image are noted as indicated in Figure 4 (right). So (x_{00}, y_{00}), (x_{10}, y_{10}), (x_{01}, y_{01}) and (x_{11}, y_{11}) denote the pixels of the quad around sub-pixel match $(\hat{x}, \hat{y})$ with respect to (x_P, y_P). Using the notations $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ for *floor* and *ceil* integer rounding, a camera pixel (x, y) would be a feasible corner point of four adjacent quadrilaterals containing sub-pixel camera matches with respect to four projector pixels. Thereby, it would be exactly one bottom left, one bottom right, one top left and one top right corner of the four corresponding quadrilaterals. The buffers for the projector pixels are filled by traversing the image and assigning each image pixel as:

$$(x, y) \;\longrightarrow\; \begin{matrix} \big(\lceil\hat{x}_P\rceil_{00}, \lceil\hat{y}_P\rceil_{00}\big) \\ \big(\lfloor\hat{x}_P\rfloor_{10}, \lceil\hat{y}_P\rceil_{10}\big) \\ \big(\lceil\hat{x}_P\rceil_{01}, \lfloor\hat{y}_P\rfloor_{01}\big) \\ \big(\lfloor\hat{x}_P\rfloor_{11}, \lfloor\hat{y}_P\rfloor_{11}\big) \end{matrix} \tag{2}$$

Since phases in arbitrary real scenes are usually sampled non-uniformly, it may be possible that several camera pixels are feasible corner points of a specific quadrilateral. Using the example of a lower left corner point, the quality of the corner point can be calculated by its distance to the optimal sub-pixel value:

$$E = \left|\hat{x}_P - \lceil\hat{x}_P\rceil_{00}\right| + \left|\hat{y}_P - \lceil\hat{y}_P\rceil_{00}\right| \tag{3}$$

If a corner point is already occupied when running through the image, it is only replaced if this error is less than that of the previously stored pixel. This ensures that the enclosing quadrilateral becomes minimal.



(a) (b) (c)

Fig. 3: (a) Visualization of optimal sub-pixel matches (red) between projector (center) and two cameras (left and right). Example of a corner point update (b) (Consistency properties stay fulfilled). Quad, that would be removed by diagonal check (b).

3.2 Topological Consistency Check (TCC)

An important property of a projection is that the topology of the projected points remains consistent. Therefore, also surface points that have been encoded using structured light must remain consistent in the corresponding phase images. Some tests are introduced, that enforce the topology preservation property. Most importantly, they remain valid for non-minimal quads, allowing their application on non-final temporal stores of corner points. This way, incorrect and noisy phase values are excluded from matching, resulting in smoother and more accurate matches with way less outliers.

Before saving any image pixel to a corner point (x_{00}, y_{00}), (x_{10}, y_{10}), (x_{01}, y_{01}) or (x_{11}, y_{11}) with respect to a projector pixel (x_P, y_P), it is ensured that a lower left pixel in the camera phase is also a lower left pixel in the projector phase and so on. In this way, many faulty matches can be detected and avoided. Moreover, it ensures that the resulting quads are convex. The following simple checks have to be fulfilled:

$$\begin{array}{ccc} (x_{01}, y_{01}) & \xrightarrow{x_{01} \leq x_{11}} & (x_{11}, y_{11}) \\ \uparrow{\scriptstyle y_{00} \leq y_{01}} & & \uparrow{\scriptstyle y_{10} \leq y_{11}} \\ (x_{00}, y_{00}) & \xrightarrow{x_{00} \leq x_{10}} & (x_{10}, y_{10}) \end{array} \tag{4}$$

The tests are applied to the pixels during the storing process while looping through the images. Naturally, therefore, during the storing process, one vertex is checked for consistency with respect to other vertices that are not final and that may be part of non-minimal representations of a quad around a sub-pixel match. As already mentioned these tests are also valid for non-minimal quads as long as they do not represent severe outliers, which moreover would simply lead to finding no match for the projector rather than an outlier. Figure 3 (b) illustrates an update of a corner point to a closer representation. It is easy to see that the convexity properties are fulfilled throughout by all points, while converging to the minimal representation.

Diagonal Check for Weak Quads Theoretically, the quadrilaterals can take a wide variety of shapes and still satisfy the desired topology and convexity. But the more unusual the shape, the worse its content is determined by bilinear interpolation. An additional optional test avoids unnatural quads by checking the diagonal values:

$$|x_{00} - x_{11}| + |y_{00} - y_{11}| < \tau, \qquad |x_{10} - x_{01}| + |y_{10} - y_{01}| < \tau \tag{5}$$

The quads should not be of arbitrary size just because they might theoretically be feasible. Usually they will not provide an accurate measurement if the corners are above a certain distance, which can be generously set to $\tau = 5$ pixels for most applications.

For illustration, Figure 3 (c) shows an example of an unfavorable quad that would be removed. Note that this check should only be done after the entire quad matching, otherwise some quads may be removed due to non-minimal representations that may have improved over time.

Epipolar Consistency Check (ECC) In many practical scenarios a rough calibration of the setup is already available. This can be extremely advantageous and easily involved into the scheme. In this case a camera point should only be mapped to a corner if the symmetric epipolar error is below a certain threshold.

In order to illustrate the effect of the checks on calculated matches, Figure 2 shows the resulting point cloud of *FPDM* without (a) and with *TCC* (b). There are significantly fewer outliers, resulting in less flying points. (c) shows how the matches can be further improved by *ECC* by avoiding faulty assignments, especially in discontinuities of the scene. False matches can also occur due to incorrect but permissible phase regions, caused by (inter-)reflections.

4 Bilinear Sub-Pixel Matching

After the quad matching, for each permissible projector pixel a consistent convex quadrilateral is given per camera image. Under certain assumptions it is possible to determine the sub-pixel position of the optimal match from the corners of the quad and their phase values. The optimal sub-pixel position is calculated in the unit patch using bilinear interpolation assumption and then mapped to the convex region as shown in Figure 4.

4.1 Sub-Pixel Position in Unit Patch

Given a unitary patch, with horizontal phase values ϕ_{H00}, ϕ_{H10}, ϕ_{H01} and ϕ_{H11} of the corner points as depicted in Figure 4, the bilinear interpolated value $\phi_H(\tilde{x}, \tilde{y})$ for any position $(\tilde{x}, \tilde{y}) \in [0, 1]^2$ is given by

$$\phi_H(\tilde{x}, \tilde{y}) = a_0 + a_1\tilde{x} + a_2\tilde{y} + a_3\tilde{x}\tilde{y}, \qquad \begin{aligned} a_0 &= \phi_{H00} \\ a_1 &= \phi_{H10} - \phi_{H00} \\ a_2 &= \phi_{H01} - \phi_{H00} \\ a_3 &= \phi_{H11} + \phi_{H00} - \phi_{H10} - \phi_{H01} \end{aligned} \tag{6}$$

and analoguesly for the vertical phase by $\phi_V(\tilde{x}, \tilde{y}) = b_0 + b_1\tilde{x} + b_2\tilde{y} + b_3\tilde{x}\tilde{y}$. Figure 5 (a) illustrates how two bilinearly interpolated phases on the unit patch can look like. Task is to find the optimal sub-pixel position inside the patch meeting the phase values $(\hat{\phi}_H, \hat{\phi}_V)$. The patch that interpolates the horizontal phase values defines a two-dimensional curve on which the value $\hat{\phi}_H$ is assumed. The same applies to the patch of the vertical phase values, which describes a curve for $\hat{\phi}_V$. Such curves are visualized by red lines in Figure 5 (a) and in a top view (b). The intersection of the curves within the patch is the optimal position of the sought sub-pixel match and marked by green dots.

In order to find optimal positions $\tilde{x} \in [0,1]$ and $\tilde{y} \in [0,1]$ at which the bilinear interpolated patches meet the sought values $\hat{\phi}_H$ and $\hat{\phi}_V$ we explicitly solve:

$$\begin{matrix}\hat{\phi}_H = a_0 + a_1\tilde{x} + a_2\tilde{y} + a_3\tilde{x}\tilde{y} \\ \hat{\phi}_V = b_0 + b_1\tilde{x} + b_2\tilde{y} + b_3\tilde{x}\tilde{y}\end{matrix} \rightarrow \quad \tilde{x} = \begin{cases} -\frac{v}{2u} \pm \sqrt{\left(\frac{v}{2u}\right)^2 - \frac{w}{u}} & , \quad u \neq 0 \\ -\frac{w}{v} & , \quad u = 0 \end{cases} \tag{7}$$

$$\text{with} \quad \begin{matrix} u = b_1a_3 - b_3a_1 \\ v = b_1a_2 + (b_0 - \hat{\phi}_V)a_3 - b_2a_1 - b_3(a_0 - \hat{\phi}_H) \\ w = (b_0 - \hat{\phi}_V)a_2 - b_2(a_0 - \hat{\phi}_H) \end{matrix}$$

The vertical position $\tilde{y}$ can then be computed from Eq. (7). Note that the properties of the quads received in Sec. 3 ensure the existence of intersection within each patch.

Existence of Solution The interpolated value $\hat{\phi}_H$ is by construction achieved inside the patch and moreover the following holds true due to the consistency checks:

$$\phi_{H00}, \phi_{H01} \leq \hat{\phi}_H \leq \phi_{H10}, \phi_{H11} \tag{8}$$

Of course for any convex combination with $\tilde{y} \in [0,1]$, we also have:

$$(1-\tilde{y})\phi_{H00} + \tilde{y}\phi_{H01} \leq \hat{\phi}_H \leq (1-\tilde{y})\phi_{H10} + \tilde{y}\phi_{H11} \tag{9}$$

Therefore the curve, defined by the first equation of (7), that maps feasible positions $\tilde{x}$ to any value $\tilde{y} \in [0,1]$ has the following property:

$$\tilde{x} = \frac{\hat{\phi}_H - a_0 - a_2\tilde{y}}{a_1 + a_3\tilde{y}} = \underbrace{\frac{\overbrace{\hat{\phi}_H - (1-\tilde{y})\phi_{H00} - \tilde{y}\phi_{H01}}^{=(i)\geq 0 \ (9),}}{(1-\tilde{y})\underbrace{(\phi_{H10} - \phi_{H00})}_{>0 \ (8)} + \tilde{y}\underbrace{(\phi_{H11} - \phi_{H01})}_{>0 \ (8)}}}_{=(ii)>0 \ (9)} \geq 0 \tag{10}$$

Thereby, we neglect the situation in which all corner points carry the same value. In this case division by zero would not be defined. Nevertheless, in this case an optimal integer pixel match exists and interpolation is not necessary.

Additionally, the denominator (ii) in this fraction is greater or equal than the numerator (i), which limits the fraction to 1:

$$(ii) - (i) = (1-\tilde{y})\phi_{H10} + \tilde{y}\phi_{H11} - \hat{\phi}_H \overset{(9)}{\geq} 0 \tag{11}$$

Similar for the vertical phase we obtain for the curves of equations (7) the properties:

$$\tilde{x} = \tfrac{\hat{\phi}_H - a_0 - a_2\tilde{y}}{a_1 + a_3\tilde{y}} \in [0,1] \text{ for } \tilde{y} \in [0,1], \ \tilde{y} = \tfrac{\hat{\phi}_V - b_0 - b_1\tilde{x}}{b_2 + b_3\tilde{x}} \in [0,1] \text{ for } \tilde{x} \in [0,1]$$

Therefore, the curves are defined for all $\tilde{y}, \tilde{x} \in [0,1]$ and map to values $\tilde{x}, \tilde{y} \in [0,1]$. This proves that the continuous curves describe continuous connections between the top and bottom of the patch (for $\tilde{x} \in [0,1]$) and from left side to right side (for $\tilde{y} \in [0,1]$) that both run inside the patch. These curves must therefore intersect at least once within the patch. This guarantees a solution, which can be explicitly computed by solving the resulting quadratic equation (7) and choosing the feasible one inside the patch. □

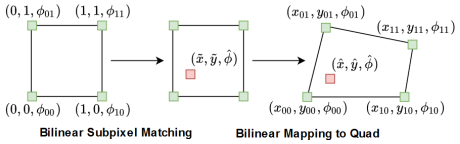
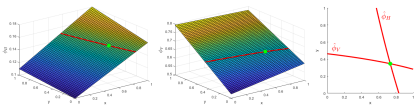


Fig. 4: Procedure of bilinear sub-pixel matching. The position is computed in the unit patch and mapped to the convex quad.



(a) (b)

Fig. 5: Unit patch with interpol. phases (a). Red curves are possible solutions, the intersection (green) solves the problem.

4.2 Mapping to Convex Quad

In general, the corner points around a sub-pixel match will not span a square region. However, for convex quads, the method can be applied by assuming an additional bilinear interpolation scheme. With corresponding corner points in the image given by

$$\begin{aligned}(x_{01},y_{01}) &\leftrightarrow (0,1) \qquad (x_{11},y_{11}) \leftrightarrow (1,1)\\ (x_{00},y_{00}) &\leftrightarrow (0,0) \qquad (x_{10},y_{10}) \leftrightarrow (1,0)\end{aligned} \tag{12}$$

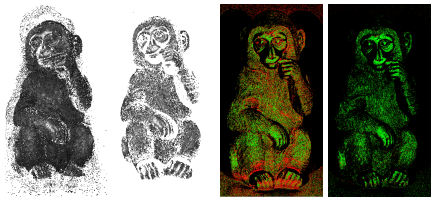
a point $(\tilde{x},\tilde{y}) \in [0,1]^2$ in the unit square can be mapped to the convex quadrilateral by:

$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}x_{00} & x_{10} & x_{01} & x_{11}\\ y_{00} & y_{10} & y_{01} & y_{11}\end{pmatrix} \begin{pmatrix}1 & -1 & -1 & 1\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & -1\\ 0 & 0 & 0 & 1\end{pmatrix} \begin{pmatrix}1\\ \tilde{x}\\ \tilde{y}\\ \tilde{x}\tilde{y}\end{pmatrix} \tag{13}$$

5 Results

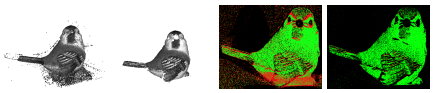
Figure 6 shows the reconstructed point clouds of different scenes as a qualitative illustration of the reconstructions obtained. For each scene, the left reconstruction shows the result of the matches obtained with best-pixel correspondences. The right point cloud shows the result of the *Fast Projector Driven Consistent Sub-Pixel Matching (FPCSM)* presented in this work. For (a-d) the images on the far right show in addition the back-projections of the points onto the projector image, with in- and outliers marked in green and red. The reconstructions are significantly smoother and contain almost no outliers. Of particular note is the *Monkey* data, which was taken from a highly specular metallic brushed monkey statue, which clearly shows the influence and improvements of the consistency checks. Of course there are methods to smooth out noisy results and to remove flying points in post processings, but the method presented here removes outliers during the matching process without any additional computational effort. Also, in contrast to smoothing, erroneous measures are removed and not smeared over the entire point cloud. Especially, if the correspondences are used for auto-calibration procedures this can be a huge advantage. Figure 7 (b) and (c) shows the enlargement of two regions in the reconstructed Buddha statue (a). Due to the optimal sub-pixel matching, the surface is much more uniformly sampled and less noisy. Especially for subsequent meshing and highly precise depth measurement this may have a significant influence.

(a) *Buddha* dataset

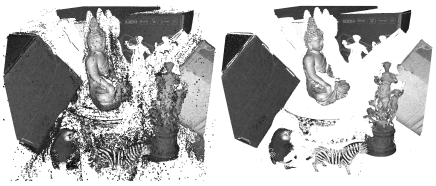


(b) *Monkey* dataset

(c) *Totem* dataset

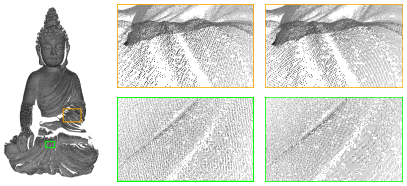


(d) *Bird* dataset



(e) *Scene* dataset

Fig. 6: Results of *FPCSM* (right) applied to exemplary scenes in comparison to point clouds obtained by standard best-pixel matching (left). Each set shows the point clouds and for (a-d) their backprojection to the projector image with labeled in- and outliers.



(a) (b) (c)

Fig. 7: Reconstr. point clouds using best-pixel matches (b) and FPCSM (c).

Set	**Best-Pixel Matches**		**FPCSM Matches**	
	Cam 1	Cam 2	Cam 1	Cam 2
Buddha	0.35447	0.35486	0.25326	0.25401
Bird	0.37116	0.37248	0.26199	0.26130
Totem	0.32996	0.33831	0.25896	0.26139
Monkey	0.37887	0.37552	0.27757	0.27818
Scene	0.26731	0.27925	0.17088	0.17866

Table 1: Median backprojection errors for best-pixel matching and FPCSM.

Finally, Table 1 shows the reduction of the median backprojection errors on the camera images from which they were triangulated. The median error was chosen to avoid overweighting extreme outliers of the standard approach without consistency checks.

6 Conclusions

In this work a matching strategy has been presented which generates high-precision correspondences for structured light systems with any number of cameras. The matches are estimated in sub-pixel accuracy. Therefore, an explicit formula has been derived,

This work was partially funded by the projects MARMORBILD (03VP00293) and VIDETE (01W18002) of the German Federal Ministry of Education and Research (BMBF).

which provides matches under the assumption of bilinearly interpolated patches. The existence of such matches has been mathematically investigated and proven. An important contribution is that this is achieved with linear complexity, while simultaneously ensuring topological consistency over the views. This results in high quality matches with nearly no outliers, that are uniformly sampled over the scene. Overall, a method has been developed which reaches extremely high accuracy with extremely low (linear) computational effort, that may be applicable to many active reconstruction applications.

References

1. Donate, A., Liu, X., Collins, E.G.: Efficient path-based stereo matching with subpixel accuracy. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **41**(1), 183–195 (2010)
2. Du, Q., Liu, R., Guan, B., Pan, Y., Sun, S.: Stereo-matching network for structured light. IEEE Signal Processing Letters **26**(1), 164–168 (2018)
3. El Asmi, C., Roy, S.: Subpixel unsynchronized unstructured light. In: VISIGRAPP (5: VISAPP) (2019)
4. Fetzer, T., Reis, G., Stricker, D.: Robust auto-calibration for practical scanning setups from epipolar and trifocal relations. In: 2019 16th International Conference on Machine Vision Applications (MVA). pp. 1–6. IEEE (2019)
5. Fetzer, T., Reis, G., Stricker, D.: Stable intrinsic auto-calibration from fundamental matrices of devices with uncorrelated camera parameters. In: The IEEE Winter Conference on Applications of Computer Vision. pp. 221–230 (2020)
6. Fetzer, T., Reis, G., Stricker, D.: Simultaneous bi-directional structured light encoding for practical uncalibrated profilometry. In: International Conference on Computer Analysis of Images and Patterns. Springer (2021)
7. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
8. Hu, Y.: Research on a three-dimensional reconstruction method based on the feature matching algorithm of a scale-invariant feature transform. Mathematical and computer modelling **54**(3-4), 919–923 (2011)
9. Isack, H., Boykov, Y.: Energy based multi-model fitting & matching for 3d reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1146–1153 (2014)
10. Li, F., Li, Q., Zhang, T., Niu, Y., Shi, G.: Depth acquisition with the combination of structured light and deep learning stereo matching. Signal Processing: Image Communication **75** (2019)
11. Ma, S., Shen, Y., Qian, J., Chen, H., Hao, Z., Yang, L.: Binocular structured light stereo matching approach for dense facial disparity map. In: Australasian Joint Conference on Artificial Intelligence. pp. 550–559. Springer (2011)
12. Pribanic, T., Obradovic, N., Salvi, J.: Stereo computation combining structured light and passive stereo matching. Optics Communications **285**(6) (2012)
13. Ryan Fanello, S., Rhemann, C., Tankovich, V., Kowdle, A., Orts Escolano, S., Kim, D., Izadi, S.: Hyperdepth: Learning depth from structured light without matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5441–5450 (2016)
14. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. vol. 1, pp. I–I. IEEE (2003)
15. Xie, J., Mo, F., Yang, C., Lia, P., Tian, S.: A novel sub-pixel matching algorithm based on phase correlation using peak calculation. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **1** (2016)