NRC Publications Archive Archives des publications du CNRC

Camera-Projector Matching Using an Unstructured Video Stream Drouin, Marc-Antoine; Jodoin, Pierre-Marc; Prémont, Julien

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

*Proceedings of the IEEE International Workshop on Projector-Camera System,
2010, 2010-06-18*

NRC Publications Archive Record / Notice des Archives des publications du CNRC :
<https://nrc-publications.canada.ca/eng/view/object/?id=a279810e-ee73-4246-a72b-4980f27e57c5>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=a279810e-ee73-4246-a72b-4980f27e57c5>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

Camera-Projector Matching Using an Unstructured Video Stream

Marc-Antoine Drouin¹ Pierre-Marc Jodoin² Julien Prémont²

¹Institute of Information Technology
National Research Council Canada
Ottawa, Canada

²MOIVRE
Université de Sherbrooke
Sherbrooke, Canada

Marc-Antoine.Drouin@nrc-cnrc.gc.ca {pierre-marc.jodoin,julien.premont}@usherbrooke.ca

Abstract

This paper presents a novel approach for matching 2D points between a video projector and a digital camera. Our method is motivated by camera-projector applications for which the projected image needs to be warped to prevent geometric distortion. Since the warping process often needs geometric information on the 3D scene that can only be obtained from triangulation, we propose a technique for matching points in the projector to points in the camera based on arbitrary video sequences. The novelty of our method lies in the fact that it does not require the use of pre-designed structured light patterns as is usually the case. The back bone of our application lies in a function that matches activity patterns instead of colors. This makes our method robust to pose, to severe photometric and geometric distortions. It also does not require calibration of the color response curve of the camera-projector system. We present quantitative and qualitative results with synthetic and real life examples, and compare the proposed method with the scale invariant feature transform (SIFT) method and with a state-of-the-art structured light technique. We show that our method performs almost as well as structured light methods and significantly outperforms SIFT when the contrast of the video captured by the camera has been degraded.

1. Introduction

In the past decade, digital cameras and LCD/DLP video projectors have become almost ubiquitous as their price kept decreasing. This opens the door for numerous applications involving a projector and a camera such as multimedia applications, shows, digital arts, and plays to name a few.

One fundamental limitation that most projector applications have to deal with is the geometric distortion of the projected image. As far as the observer is concerned, geometric distortion appears when the projector is located far from the observer and/or when the 3D surface is badly oriented. As mentioned by Raskar *et al.* [17], in cases where the projector and the 3D surface cannot move, the only solution is to warp the projected image. In fact, given the 3D geometry of the scene, one can easily implement such a warping function to prevent distortion from the observer's stand point [16, 17, 20](if the warping is to be done for the camera's stand point, only the pixel mapping between the

Figure 1. (a) Our setup contains an LCD/DLP projector, a camera, and a piecewise planar 3D surface. The projected (b) and captured videos (c) are time-synchronized.

camera and the projector is needed [20]). Unfortunately, the geometry of the scene is often unknown *a priori* and thus needs to be estimated at runtime. One usual way of doing so is through the use of a camera and a two-step procedure. First, the camera and the projector are calibrated so their intrinsic and extrinsic parameters are known [24]. Then, pre-designed patterns of light (so-called *structured light patterns*) are projected on the surface so pixels from the projector can be matched to those of the camera. Depending on the complexity of the scene, one can project simple dots of light (in case of a planar surface for instance) or more complex patterns [18] in case of a compound surface. Once a sufficiently large number of matches has been found, a 3D surface can be recovered by triangulation. Given that both the scene and the projector stay fixed, the estimated 3D surface can be used to warp the projected image for any viewpoint.

One obvious problem arises when the camera/projector system and/or the 3D scene is moved after calibration is over. One typical example is in plays involving artistic staging. In this case, the use of structured light patterns to recover the 3D geometry becomes irrelevant as the system

should stop projecting the video to readjust. Such application thus requires the matching to be done directly from the projected to the captured image. Unfortunately, we empirically noticed that pure color-based matching strategies between two such images are doomed to fail. The reason being that the images captured by the camera are heavily degraded by non-linear color distortion (see Fig. 1 (b) vs (c)). This is true especially when the white balance and exposure time of the camera automatically readjust and/or when the projection surface is textured. In the experimental section, we will show how SIFT [13], although one of the most robust matching methods, fails in such scenarios.

In this paper, we propose to find camera-projector matches based on *unstructured light patterns*, i.e. based on the projected video itself. In this way, each time the system needs to recover the 3D scene, our approach analyses *activity patterns* recorded in both videos and finds matches based on it. These activity patterns are obtained following a motion detection method applied simultaneously on the video emitted by the projector and the video captured by the camera. They are then bundled with grayscale quanta and embedded into a cost function used to find matches. Once matches have been found, the 3D structure of the scene is recovered and the projected video warped. In this paper, we focus on piecewise planar surfaces, although our approach can be generalized to other geometric primitives, such as spheres and cylinders. Interestingly, our system needs between 15 and 30 frames (at most 1 second of video) to efficiently recover the 3D scene, thus allowing an artistic director to perform a quick readjustment of the system unbeknownst to the audience. We tested our method on different videos including music clips, animated movies, and home-made videos. Artistic animated patterns could also be used with our method.

2. Previous Work

Finding matches in two images (here camera-projector matches) is the first step for most applications involving a triangulation procedure. There has been a significant effort to develop simple and efficient matching strategies that we summarize in four categories.

Structured Light Certainly one of the most implemented strategy, structured light methods use pre-designed patterns of light to encode the pixel position. All kinds of patterns have been proposed so far including color, binary and grayscale patterns, patterns with spatial coding, others with time-multiplexing coding, some being dense, others being sparse, etc. [22, 18]. As far as our system is concerned, structured light is hardly a solution since the pose between the system and the scene may vary in time. In that case, the system would need to periodically readjust by stopping the user-selected video for recalibrating the system. This, of course, is unacceptable for obvious marketing reasons.

A solution to that problem is to embed imperceptible patterns of light into the projected image. One way of doing so is by reducing the dynamic interval of DLP projectors [1, 26]. This solution however, is only conceivable for high-end (and very costly) projectors. Our solution does not suffer from such limitations as it uses *unstructured light* based on the ongoing video.

Feature-Based Matching A second approach consists in matching feature points extracted from the projected and the captured images [21, 25, 11]. One such approach that drew a lot of attention lately is the scale invariant feature transform (SIFT) [13]. SIFT is one of the very few methods which provides a solution for both extracting feature points and finding point-to-point matches. The main advantage with SIFT lies in its robustness to geometric transformations and non-linear illumination distortions. This being said, we empirically observed that the number of matches SIFT returns rapidly decreases in presence of severe perspective transformations and/or illumination distortions (results with SIFT are presented in sec. 5). This is a major limitation as far as our application is concerned. Empirical results will be shown in Section 5.

Stereovision Stereovision methods are typically used on images taken by 2 cameras mounted side-by-side [19]. Unfortunately, it has long been documented that simple (but realtime) greedy optimization strategies such as *winner-take-all* underperform in textureless areas and that only global (and slow) optimizers such as graph cut or belief propagation provide decent matches [19]. This makes the stereovision strategy ill-suited for our camera-projector setup which calls for fast solutions. Also, since there is a significant color distortion between the projected and the captured videos, stereovision methods based on a color-constancy hypothesis are doomed to fail [19]. Note that cost functions based on mutual information have been designed to deal with color inconsistency problems [10]. Nevertheless, these cost functions are computationally expensive and are not adapted to in-line systems as ours.

Let us however mention that stereovision could be a sound solution for a two-camera/one-projector system [12]. That would be true especially when the projected video contains a lot of texture allowing for simple greedy methods to work. Such methods are known as spatio-temporal stereovision [2, 23].

Cooperative Scenes Markers made of vivid colors and an easy-to-locate design can be physically stitched to the scene. One example of such markers are ArTags [6] which show great robustness. However, we empirically observed that visual tags are not easy to detect when color patterns are projected on the scene. Also, for obvious aesthetic reasons, some applications involving live shows or home products forbid the use of markers. Let us mention that infrared LEDs with infrared cameras are sometime utilized. How-

ever, such a method is costly, requires extra hardware, and is not robust in areas where the ambient temperature fluctuates in time.

3. Overview of Our Method

In this section, an overview of our method is presented to allow high-level understanding. Our method is based on five steps that will be described in more details in sec. 4.

1. Calibrate the camera and the projector to estimate intrinsic and extrinsic parameters.
2. Start projecting and capturing the video and, at each time t , detect motion and assign a grayscale quantum to each pixel in both videos.
3. Find camera/projector matches based on grayscale quanta.
4. Out of these matches, estimate the 3D surface. Since we make the assumption that the surface is piecewise planar, the equations of m planes are estimated with RANSAC.
5. Given the current m planes, warp the projected video.

4. Details of Our Method

4.1. Camera-Projector Calibration

As opposed to what the schematic representation of Fig. 1 suggests, our camera and projector are screwed to a common plate so their relative position and orientation stay fixed during the entire projection. The camera and the projector thus need to be calibrated only once at the beginning of the process. To do so, we use Zhang’s calibration method [24] which enables the projector to work like a camera, and thus allows it to be calibrated like a camera. To avoid user intervention, structured-light patterns are first projected on a flat checkerboard to get a one-to-one correspondence between the pixels of the camera and the pixels of the projector. The checkerboard corners are then detected to calibrate the camera and recover the 3D plane. The projector is then calibrated using the correspondences and the known 3D position of the corners on the checkerboard. At the end of this stage, the intrinsic and extrinsic parameters of the camera and the projector have been estimated. These parameters will be used to rectify the videos (sec. 4.3), recover the 3D geometry of the scene (sec. 4.4), and warp the projected video (sec. 4.5). Let us stress the fact that the calibration has to be performed only once and that the parameters of the system can be used for many projections.

4.2. Motion Detection and Quantization

Once calibration is over, the system starts projecting the user-selected video on the 3D scene. At the same time, the camera captures the scene on which the video is projected (the camera is synchronized with the projector). As mentioned previously, the goal is to find matches in the projected and captured images so that the geometry of the scene

can be recovered. This is done based on motion labels that we estimate with a simple background subtraction strategy.

Let f_t^p and f_t^c be the projected and captured video frames at time t , both containing RGB values. At each time t , a reference image r_t^p and r_t^c is subtracted (and then thresholded) from the input frames, so that binary motion fields $\mathcal{X}_t^p$ and $\mathcal{X}_t^c$ are obtained :

$$\begin{aligned} r_{t+1}^i &= \alpha f_t^i + (1 - \alpha) r_t^i, \text{ and } r_0^i = f_0^i \\ U_t^i(x, y) &= ||\bar{f}_t^i(x, y) - \bar{r}_t^i(x, y)|| \\ \mathcal{X}_t^i(x, y) &= \begin{cases} 1 & \text{if } U_t^i(x, y) > \tau \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

where $i = c$ or p , $\alpha \in [0, 1]$, τ is a threshold, and $||\cdot||$ stands for the Euclidean norm.

We noticed that noise, illumination changes and local brightness variations make the use of a global and fixed threshold τ error prone. To avoid errors, τ is computed adaptively and locally. In this perspective, U_t^c and U_t^p are first split into $p \times q$ blocks. Then, for each block, we compute the threshold which maximizes the inter-class variance following the Otsu segmentation technique [14]. The value of τ for each pixel (x, y) is finally linearly interpolated from the threshold of the four nearest blocks.

To further improve robustness, active pixels are assigned a grayscale quantum ($Q_t^c(x, y)$ and $Q_t^p(x, y)$) following a quantization procedure. Since our system calls for CPU-aware solutions, we use a median-cut algorithm on grayscale versions of the videos frames f_t^p and f_t^c [9]. From the grayscale histogram of f_t^p and f_t^c , median-cut recursively divides the 1D space into bins of various sizes, each containing the same population. Once the algorithm has converged, each active pixel is assigned the bin index (read *quantum*) its grayscale falls into. In this way, $Q_t^i(x, y) \in \{1, 2, \dots, N\}$ for active pixels (*i.e.* pixels for which $\mathcal{X}_t^i(x, y) = 1$) and $Q_t^i(x, y) = 0$ for inactive pixels.

4.3. Camera-Projector Matching

Now that every active pixel has been assigned a grayscale quantum, the goal is to find for every pixel (x_c, y_c) in the captured image its corresponding point (x_p, y_p) in the projected image. Since the camera and the projector are mounted side-by-side, both videos are rectified so their epipolar lines are horizontally aligned [7]. This is done with the intrinsic and extrinsic parameters estimated in sec. 4.1. The matching procedure now looks for the best match (x_p, y_c) in the rectified projected image given the pixel (x_c, y_c) in the rectified captured image. We denote x_p as the horizontal position of the corresponding point at time t of pixel (x_c, y_c) and X_p as the correspondence map. The best correspondence map X_p is the one which minimizes a given criteria whose definition is pivotal for our method. Given that each pixel (x_c, y_c) in the camera is assigned a specific set of quanta $\Gamma^c = \{Q_{t-W}^c(x_c, y_c), \dots, Q_t^c(x_c, y_c)\}$

Figure 2. Proportion of outliers returned by *Winner-Take-All* (WTA) and dynamic programming (DP) at different time instants of a video.

over a period of time W , the goal is to find the pixel in the projector which has a similar set of quanta $\Gamma^p = \{Q_{t-W}^p(x_p, y_c), \dots, Q_t^p(x_p, y_c)\}$. This leads to the following formulation

$$X_p = \arg \min_{\hat{X}_p} \sum_{x_c, y_c} C(\Gamma^c, \Gamma^p, x_c, y_c, \hat{X}_p)$$

where $C(\cdot)$ is a cost function measuring how similar two sets of quanta Γ^c and Γ^p are. Since Γ^c and Γ^p are two vectors of equal length, $C(\cdot)$ could be a simple Euclidean distance. Unfortunately, this function is error prone and needs to be replaced by a more specific function. The cost function that we came up with considers that two sets of quanta Γ^c and Γ^p are similar when they contain activity at the same time instant and when their spatial and temporal gradient are similar. Mathematically, this leads to

$$\sum_{\tau=t-W}^t \delta(Q_s^c, Q_s^p) \left(\sum_r \omega(Q_s^c - Q_r^c, Q_s^p - Q_r^p) \right)$$

where $Q_s^c = Q_\tau^c(x_c, y_c)$, $Q_s^p = Q_\tau^p(x_p, y_c)$ and r is a first-order spatio-temporal neighbor of (x_c, y_c, τ) in Q^c and (x_p, y_c, τ) in Q^p $\delta(a, b) = 1$ when $a, b > 0$ or when $a, b = 0$ and COSTMAX otherwise. As for $\omega(a, b)$, it returns 0 when $\text{sign}(a) \times \text{sign}(b) \geq 0$ and COSTMAX otherwise.

Optimization method Eq. 1 could be solved with a simple greedy *winner-take-all* (WTA) optimizer [19]. Unfortunately, we empirically observed that WTA generates a large number of outliers (read “bad matches”) which can propagate errors in the upcoming steps of the method. In order to reduce the number of outliers, we enforce an ordering constraint (OC). The OC states that if a point A is to the left of a point B in one image, then point A is also to the left of point B in the other image. Although the OC can be violated in scenes containing thin objects and/or large occlusions [4], our system works only on 3D scenes that can be used as visualization surfaces. Thus, the OC is fulfilled in all scenes that we deal with.

The OC can be enforced without significant increase in CPU effort, thanks to dynamic programming (DP) [22]. In

our method, every epipolar line is processed with a DP algorithm as in [3, 22] where the OC replaces the visibility constraint¹. Note that every pixel with no activity (*i.e.* those whose quanta are all set to zero) are not processed. To further speed up the process, a *fast message passing strategy* can be used [5] to reduce DP’s complexity to that of WTA. Further details concerning DP will be given in the journal version of this paper.

As can be seen in Fig. 2, DP significantly reduces the number of outliers as compared to WTA.

4.4. Fitting m Planes on 3D Points

At this stage of processing, a match has been found for every pixel at which activity had been recorded. This gives a sparse correspondence map in which each camera pixel (x_c, y_c) is assigned to a horizontal position (x_p, y_c) (see Fig. 3). Since the projection surface is piecewise planar, m planes can be fitted onto these points. Let us first see how one plane can be fitted on such a correspondence map. We will then see how m planes can be fitted and how outliers are handled.

Fitting One Plane Let $P_t = \{p^1, p^2, \dots, p^N\}$ be a set of points $p^j = (x_c^j, y_c^j, x_p^j)$ in the projective space estimated at time t and stored in a correspondence map (see Fig. 3). Given that the 3D points are all inliers and distributed (more or less some noise) on a plane, a typical way of calculating the best-fitting plane is by minimizing the square of the offsets. The offset of a point is usually its *perpendicular* distance to the plane. However, since our points lie on a rectangular lattice (the correspondence map), we consider instead the offset along the third dimension of p^j since we only expect to have errors on the x_p^j coordinate.

Let $ax_c + by_c + cx_p + d = 0$ be the equation of a plane. Since a projection surface cannot be parallel to the viewing axis, one can set $c = 1$ to reduce by 1 the number of unknowns. Given a point (x_c^j, y_c^j, x_p^j) , its depth according to the plane is $x_p = -(ax_c^j + by_c^j + d)$ and its *squared depth offset* is $(x_p^j - x_p)^2$. Thus, the best plane given P_t is the one which minimizes the depth offset for every point, namely

$$E(P, A) = \sum_j (\hat{p}^j A + x_p^j)^2 \quad (1)$$

where $\hat{p}^j = (x_c^j, y_c^j, 1)$ and $A = (a, b, d)^T$. By forcing $dE/dA = 0$, one can show that $A = -M^{-1}B$ where

$$M = \begin{pmatrix} \sum_j (x_c^j)^2 & \sum_j x_c^j y_c^j & \sum_j x_c^j \\ \sum_j x_c^j y_c^j & \sum_j (y_c^j)^2 & \sum_j y_c^j \\ \sum_j x_c^j & \sum_j y_c^j & \sum_j 1 \end{pmatrix} B = \begin{pmatrix} \sum_j x_c^j x_p^j \\ \sum_j y_c^j x_p^j \\ \sum_j x_p^j \end{pmatrix}.$$

Let us mention that the just-estimated $[a, b, 1, d]$ plane (as well as any 3D point $p^j = (x_c^j, y_c^j, x_p^j)$) can be transposed

¹Our approach does not include a smoothing term between neighboring pixels as in [3]

Figure 3. (Left) Correspondence map X_p obtained with our method. A correspondence has been assigned to each pixel at which activity had been recorded. (Middle and right) 3D view of X_p on top of which we put a plane estimated with RANSAC. Inliers are in red, outliers are in black. Outliers correspond to 17% of the population.

in the 3D Euclidean space as follows : $(T^{-1})^T(a, b, 1, d)^T$ where T is a 4×4 matrix [8]. The reason why planes are fitted in the projective space (namely the correspondence map X_p) and not on 3D points in the Euclidean space is for a robustness issue whose details are beyond the scope of this paper. Let us only mention that noise in the projective space is along the third dimension only (read “ x_p ”). In the Euclidean space, noise is also anisotropic but oriented along an arbitrary direction which is costly to estimate. We will show in sec. 5 the difference between a plane fitted in the projective space and one in the 3D Euclidean space without taking into account noise orientation.

Fitting m Planes and Dealing with Outliers Assuming that the projection surface is piecewise planar, we use a modified version of RANSAC to find m different planes with their respective set of inliers [15]. Since RANSAC can only be used to fit one plane, we retained the following generalization of RANSAC:

```

minsize  $\leftarrow s \cdot \text{size}(P_t)$ ,  $i \leftarrow 1$ , exit  $\leftarrow$  false
DO
.   (inl[i], A[i])  $\leftarrow$  RANSAC( $P_t$ )
.   if (size(inl[i]) < minsize)
.        $m \leftarrow i - 1$ , exit  $\leftarrow$  true
.   else
.        $P_t \leftarrow$  remove the inliers inl[i] from  $P_t$ .
.        $i \leftarrow i + 1$ 
WHILE exit == false

```

where s is a fraction between 0 and 1. Once this procedure has converged, we have the m plane equations (here A), their related inliers (here inl) and P_t contains the outliers. Note that this algorithm does not need the number of planes m to be predefined. For more details concerning RANSAC, please refer to [8].

4.5. Warping the Projected Video

Now that the m plane equations have been recovered, the projected video can be warped. For $m=1$ plane, the procedure goes as follows :

Figure 4. 3D points recovered by our version of RANSAC as seen from the camera and the projector. The green line corresponds to the intersection between the two planes.

1. Select a viewpoint for which the geometric correction must be performed. At this position, put a virtual camera and compute its extrinsic parameters with respect to the camera frame.
2. Assign intrinsic parameters to the virtual camera.
3. Using the plane equation and the extrinsic and intrinsic parameters, compute two homography matrices : one relating the projector and the camera and one relating the camera and the virtual camera (see [8] for more details).
4. From the two homography matrices, compute a third homography matrix relating the projector and the virtual camera.
5. Warp the projected image with the third homography matrix.

Whenever $m > 1$ and the planes are connected (as in Fig. 4), the warping procedure must be applied for each plane :

- Find the intersection between each pair of planes so every pixel in the projector is associated to a plane (see Fig. 4(b)).
- Apply step 1 to 4

Figure 5. The *corridor* and *live band* videos used for testing. On the left, the projected video and on the right, the captured video.

Figure 6. Angular error between the ground truth plane and the plane estimated with our method and SIFT. Here the *corridor* sequence has been used.

- For each plane, apply step 5 on its related area in the projected image.

5. Experimental Protocol and Results

In order to gauge performances, we tested our system on scenes made of one and two planes. We tested two different videos containing different amounts of activity (see Fig. 5 (a) and (c)). The first video is called *corridor* (CRD) and contains 167 frames. It is a home video captured by a fixed camera and shows a pedestrian walking from the left to the right. The reason for this video is to see how our method works on family videos containing little activity. The second video, called *live band* (LB), contains 240 frames and shows a live music band filmed by a hand-held cellphone. This video suffers from severe compression artifacts and contains a lot of activity. For every test, we used the following parameters : $\alpha = 0.85$, $N = 6$, $COSTMAX = 2$, and $s = 0.1$. We also tested two projectors. The first one is a 1576×1080 projector that we used with the *corridor* sequence. The second one is a 225 lumen LED-based 1024×768 projector that we used for the *live band* sequence. We used a 3024×4334 camera whose images are reduced to fit the resolution of the projected videos.

Figure 7. Top) Reprojection error of a flat checkerboard recovered from 3D points (in red) obtained with our method (on the left) and SIFT (on the right). Two time instants have been selected namely $t=43$ (first row) and $t=140$ (second row).

5.1. Single-Plane Setup

Here, the video is projected on a plane to see how our method behaves when projecting on a flat surface such as a screen or a wall. The target contains fiducial markers at known positions so the plane equation can be computed using a photogrammetric method (and thus be used as a ground truth). Since those markers are visible in the captured video (see Fig. 5 (b)), it allows to see how the system behaves when the video is projected on a textured surface.

First, we tested the *corridor* sequence, which is by far the most difficult sequence due to its small amount of activity. Given a temporal window W of 30 frames, we recover the plane equation at each time t . As shown in Fig. 6 our method produces an average error of 2.8 degrees between the ground truth plane and the estimated plane². This corresponds to a reprojection error of at most 4 pixels (Fig. 7). This is obvious when considering Fig. 10 (a) and (b) in which a warped checkerboard has been projected on a planar surface. Fig. 6 also shows that plane fitting in the projective space is more robust than in the Euclidean space.

The left hand side of Fig 7 and Fig 8 shows inliers found by our method at two time instants. As can be seen, our method has been capable of recovering 10422 matches at frame 43 and 941 matches at frame 140. The number of matches found at a given time t depends on the amount of activity registered at that period of time.

In Fig. 3, a 3D plane has been recovered while projecting the *live band* video sequence with $W = 30$. An average of 13545 inliers has been found and the recovered 3D plane has an angular error of less than 3 degrees.

5.2. Comparison with SIFT

Here, we kept the same processing pipeline except for the matching procedure (sec. 4.3) which we replaced by

²Our method estimates the 3D surface only once. We estimated a plane at each time t in Fig. 6 and 9 only to show how our method behaves on different frames.

Figure 8. 3D points recovered by our system (on the left) and SIFT (on the right) at $t = 43$ and $t = 140$. In blue is the ground truth plane and in red 3D points. As shown in Fig. 6, due to a small number of 3D points found by SIFT at frame 43, the angular error rises above 40 degrees.

Sequence \ Angle	Number of matches found			
	0	5	25	50
Ours - LB - 1	50232	46670	35886	51035
Ours - LB - 2	49483	41300	36457	43952
Ours - LB - 4	46466	41900	34402	44228
Ours - CRD - 1	49722	45108	37291	49393
Ours - CRD - 2	48342	43086	38618	46358
Ours - CRD - 4	45800	42322	36997	46417
SIFT - LB - 1	2664	2897	3198	2269
SIFT - LB - 2	2117	2343	2612	1872
SIFT - LB - 4	1184	1358	1538	1055
SIFT - CRD - 1	276	337	398	331
SIFT - CRD - 2	169	213	247	224
SIFT - CRD - 4	25	38	49	47

Table 1. Number of matches found by SIFT and our method with the *live band* (LB) sequence and the *corridor* (CRD) sequence in a synthetic environment. The number next to each sequence’s name is the contrast degradation factor that we applied on the images captured by the camera. Each sequence was projected on a plane tilted at 4 different angles with respect to the camera.

SIFT. Note that to make the comparison fair with our method that uses temporal coherence, every match found by SIFT at time t is propagated on the upcoming frames to allow for more matches.

We used the *corridor* sequence that we projected on a flat textured surface. As can be seen in Fig. 6, our method outperforms SIFT as it produces a much lower angular error on the average. Due to the texture on the surface and severe spatial and photometric distortions, SIFT finds a small number of matches (less than 50 as shown in Fig. 8) whose distribution gets aligned at some time instant (see Fig. 7 and 8). This leads to an average reprojection error three times larger than that obtained with our method.

Table 1 shows the number of matches found by our method and by SIFT (RANSAC was used to filter out out-

Figure 9. (Top) Angular error at each time t for both planes and (bottom) the difference in millimeters between the 3D model obtained with structured light (14 patterns) and our method.

liers) for the *live band* and *corridor* sequences projected on planes at different angles. These tests were performed in a virtual environment. We added a contrast degradation to simulate the distortion effect of a real-life camera. Table 1 clearly shows that our method finds more matches than SIFT. Furthermore, matches obtained by our method contain more than 90% of inliers, even with a contrast degradation factor of 4. This shows that our methods is more stable than SIFT which sometimes returns less than 10% of inliers.

5.3. Two-Plane Setup

In this test case, we projected the video on a two-plane wedge located 800mm away from the camera. First, we re-constructed the 3D wedge with a structured-light technique involving gray code and phase shift [18] on the full resolution images. Then, another reconstruction was performed with our method by projecting the *live band* video sequence and using a temporal window W of 15 frames. As shown in Fig. 9, we superimposed both 3D results and computed their differences in millimeters. As can be seen, the maximum error is only 4mm. The average error for both planes is -1.9 mm and -1.3mm while the average angular error is approximately 2 degrees for both planes. We can see in Fig. 10 (a) a warped checkerboard projected on the wedge and (b) its projection as seen from an arbitrary point of view.

6. Conclusion

We have presented a new camera-projector matching procedure based on activity features instead of color. This *unstructured light* matching technique performs 3D reconstruction using an arbitrary video sequence. To be robust to severe geometric and photometric distortions, our method uses binary motion labels obtained from background subtraction bundled with grayscale quanta. We presented examples from which sparse correspondences are used to recover planar primitives then used for warping. Numer-

Figure 10. (a),(c) Warped video frames according to the 3D shape recovered by our method (1 and 2 planes). (b),(d) show the image of the projected image on the 3D surface captured by the camera. As can be seen, the 3 pixel distortion is barely noticeable.

our experiments have been conducted on real and synthetic scenes. Out of those results, we conclude that :

1. Our method finds significantly more matches than SIFT, especially when the captured video suffers from severe geometric and photometric distortions, and when the projection surface is textured.
2. The 3D results obtained with our method are close to those obtained with a state-of-the-art structured light technique (gray code + phase shift).
3. Results from our method have on the average less than 3 degrees error leading to an average reprojection error of approximately 1.5 pixels.
4. A temporal window of between 15 and 30 frames is required to find good matches. The length of the temporal window depends on the amount of activity.

Our method is motivated by applications requiring digital projection AND 3D reconstruction at the same time. One of the targeted applications is artistic projections for which the 3D information (initially unknown) is needed to prewarp the projected video. Let us mention that a non-technically savvy artist could easily design visually aesthetic unstructured light patterns that would be used by our matching procedure. Furthermore, patterns for dense correspondences could also be designed using a few seconds of video. The only constraint being that every pixel of the projector be active at some point in time.

In the future, we look forward to fit more complex geometries such as quadrics. Also, we would like to extend the pixel matching procedure to a sub-pixel version. While our method requires the scene (and the camera/projector system) to remain static during the acquisition, we would like

to combine our approach with featured tracking in order to allow for reconstruction in a dynamic environment.

References

- [1] D. Cotting, R. Ziegler, M. Gross, and H. Fuchs. Adaptive instant displays: Continuously calibrated projections using per-pixel light control. *CGF*, 24(3):705–714, 2005. 2
- [2] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Space-time stereo: A unifying framework for depth from triangulation. *PAMI*, 27(2):296–302, Feb. 2005. 2
- [3] M.-A. Drouin, M. Trudeau, and S. Roy. Fast multiple-baseline stereo with occlusion. In *3DIM*, 2005. 4
- [4] G. Egnal and R. P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *PAMI*, 24(8):1127–1133, 2002. 4
- [5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006. 4
- [6] M. Fiala. Artag, a fiducial marker system using digital techniques. In *CPVR*, 2005. 2
- [7] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vis. App.*, 12(1):16–22, 2000. 3
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 5
- [9] P. Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 16:297–307, 1982. 3
- [10] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CPVR*, 2005. 2
- [11] T. Johnson and H. Fuchs. Real-time projector tracking on complex geometry using ordinary imagery. In *PROCAMS*, 2007. 2
- [12] T. Johnson, G. Welch, H. Fuchs, E. L. Force, and H. Towles. A distributed cooperative framework for continuous multi-projector pose estimation. In *VR*, 2009. 2
- [13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2
- [14] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, 9(1):62–66, 1979. 3
- [15] P. Quirk, T. Johnson, R. Skarbez, H. Towles, F. Gyarfas, and H. Fuchs. Ransac-assisted display model reconstruction for projective display. In *VR*, 2006. 5
- [16] R. Raskar and P. Beardsley. A self-correcting projector. In *CPVR*, 2001. 1
- [17] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines. ilamps: geometrically aware and self-configuring projectors. In *SIGGRAPH*. ACM, 2003. 1
- [18] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recogn.*, 2004. 1, 2, 7
- [19] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(7), 2002. 2, 4
- [20] J. Tardif, M. Trudeau, and S. Roy. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. In *3DIM*, 2003. 1
- [21] R. Yang and G. Welch. Automatic projector display surface estimation using every-day imagery. In *WSCG*, 2001. 2
- [22] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *3DPVT*, pages 24–36, 2002. 2, 4
- [23] L. Zhang, B. Curless, and S. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *CPVR*, pages 367–374, June 2003. 2
- [24] L. Zhang and S. Nayar. Projection defocus analysis for scene capture and image display. *ACM Trans. Graph.*, 2006. 1, 3
- [25] J. Zhou, L. Wang, A. Akbarzadeh, and R. Yang. Multi-projector display with continuous self-calibration. In *PROCAMS*, 2008. 2
- [26] S. Zollmann, T. Langlotz, and O. Bimber. Passive-active geometric calibration for view-dependent projections onto arbitrary surfaces. *JVRB*, 4(6), 2007. 2