Inverse Rectification for Efficient Procam Pattern Correspondence

Yubo Qiu[1] yubo.q@queensu.ca
Jonathon Malcolm[1,2] j.malcolm@queensu.ca
Abhay Vatoo[1] 17av11@queensu.ca
Sheikh Ziauddin[1,2] zia.uddin@queensu.ca
Michael Greenspan[1,2,3] michael.greenspan@queensu.ca

[1]Department of Electrical and Computer Engineering, [2]Ingenuity Labs, [3]School of Computing
Queen's University, Kingston, Ontario, Canada

Abstract

A method called inverse rectification*, is proposed which facilitates the establishment of correspondences across a projected pattern and an acquired image. A pattern of features comprising vertical dashes is warped by the inverse of the rectifying homography of the projector-camera pair, prior to projection. This warping imparts upon the system the property that projected features will fall on distinct conjugate epipolar lines of the rectified projector and acquired camera images. This reduces the correspondence search to a trivial constant-time table lookup once a feature is found in the camera image, and leads to robust, accurate, and extremely efficient disparity calculations. A projector-camera range sensor is developed based on this method, and is shown experimentally to be effective, with bandwidth exceeding some existing consumer-level range sensors.*

1. Introduction

There are a variety of techniques for sensing depth from a scene, each of which have their unique characteristics which make them best suited for a particular task, and some of which have been successfully productized. The trinagulation based Microsoft Kinect I was the first such sensor to be successfully targetted at the consumer market, and produced dense point clouds in near realtime, albeit at only moderate accuracies. The time-of-flight based Kinect II improved on both bandwidth and accuracy, although it remained best suited for human-interactive applications, such as indoor gaming systems. Other commercial sensors have targetted industrial applications such as industrial inspection and vision-guided robotics, and provide higher data accuracy, with less emphasis on depth-of-field and bandwidth. Effort is currently being directed toward the development of commercial LiDAR systems which have characteristics suitable for use in autonomous vehicle navigation.

The many uses of range data coupled with distinct requirements for each application, result in a need for a variety of different sensing methods. While it is not entirely clear which method will ultimately be best suited for a given application, it is clear that no single method will satisfy all requirements. There remains an interest, therefore, in the development of novel sensing approaches that have improved characteristics for existing applications, as well as novel characteristics to enable new applications.

A taxonomy of 3D sensing is proposed in Fig. 1, and reviews are presented in [1, 14, 4]. The two broadest categories are: Passive techniques, which include stereovision, multiview stereo, and arguably depth from focus and structure from motion, and; Active techniques, which add light energy to the scene. Establishing correspondences between planar projections of scene elements is of central importance to both passive stereo and all active triangulation methods, and their ability to perform this robustly and efficiently is at the essence of the various methods.

This paper proposes a technique called *inverse rectification*, which facilitates the establishment of correspondences across a projected pattern and an acquired image. Rectification is a well-known method to align the two imaging planes of a stereovision system. Inverse rectification exploits the rectifying homographies differently, warping the pattern so that projected features will fall on distinct conjugate epipolar lines of the rectified projector and camera images. This reduces the correspondence search to a trivial constant-time table lookup once a feature has been found in the camera image, and leads to the potential for both robust and extremely efficient disparity calculations. The technique is presented here as the core of a projector-camera (*procam*) range sensor, although it could also be applied beneficially to other existing active triangulation-based methods.

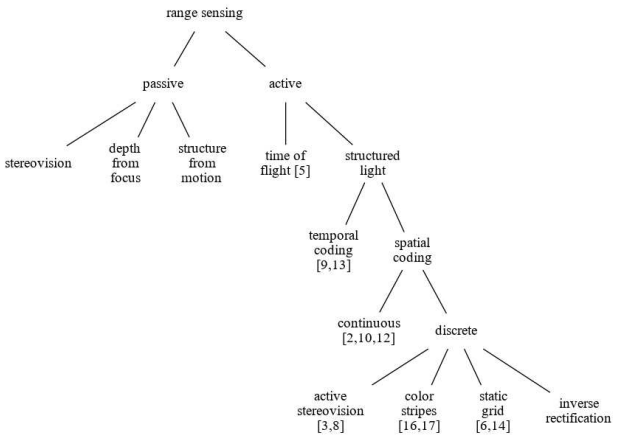


Figure 1. Taxonomy of 3D Sensing Techniques

2. Previous Work

Among the active techniques, the two broadest categories are time-of-flight sensors, which measure the time (and therefore distance) that it takes for emitted light to reflect off of a surface and return to the sensor [5], and triangulation sensors. Like passive stereovision, active triangulation sensors measure the disparity between offset planar projections of corresponding scene elements. In the case of active triangulation sensors, typically only one of the planes is an imaging sensor, whereas the other is a projected light source, such as a laser [10] or data projector.

All active triangulation-based methods establish correspondences across images by encoding the light that they transmit, so that each pixel in the acquired image can be decoded and associated with a specific projected pixel location [1, 14]. In classical temporal encoding, a series of patterns is transmitted which when stacked together identify each pixel distinctly [12]. A well-established temporal encoding method projects a sequence of binary gray coded patterns, which uniquely encode 2^n pixels with n images. Scharstein and Szeliski [15] exploited epipolar geometry to rectify the projected and acquired images, thereby isolating each row (or column) and reducing the number of required projected patterns by half.

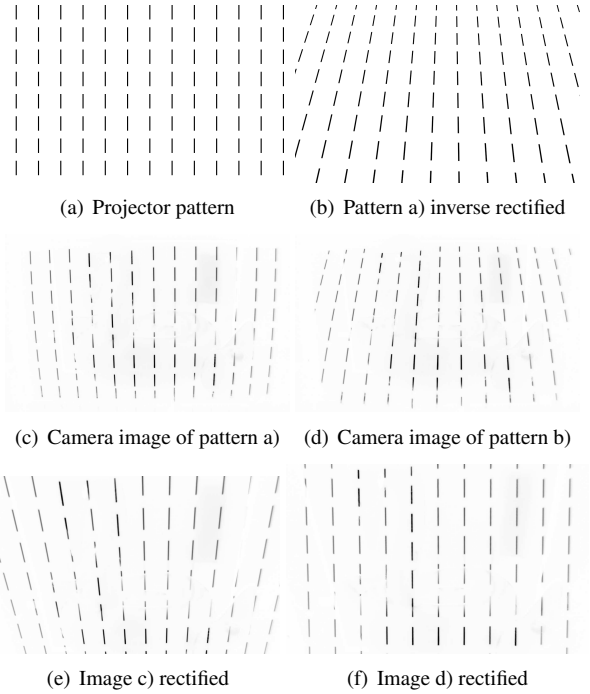
While temporal encoding can be robust and accurate, it requires multiple (often 10 to 20) images to be projected per frame, and so it is necessarily bandwidth limited. This limitation has motivated the development of spatially-encoded one-shot triangulation methods, which have demonstrated the potential of being both accurate and efficient, as they require only a single projected pattern. One-shot methods spatially encode the projected pixel locations, either through the projection of continuous or discrete patterns. Continuous methods project a single (or small number of) intensity-varying waves, and uniquely recover the phase of these signals at each acquired pixel [11, 9, 2]. These methods can be both fast and accurate, although they may be sensitive to variations in reflectance and color properties of the scene surfaces, which can confuse the interpretation of the received light intensity. There are also periodic ambiguities that can occur due to the wrapping of the projected waves, which may be resolved through the projection of additional continuous wave images of complementary frequencies.

Spatially-encoded discrete patterns form another category of one-shot method, which include the use of color coding. Zhang et al. [16] projected vertical stripes of distinct colors that followed a de Bruijn sequence, and facilitated the association of the recovered and projected sequences using a Dynamic Programming framework. Color coding suffers from similar limitations as continuous methods, when non-cooperative surfaces are imaged. Alternatives to color coding include the active stereovision approach of Scharstein and Szeliski [15], which projects color stripe patterns to improve establishing correspondences across images, which is especially beneficial for enhancing stereovision of radiometrically textureless surfaces.

Recently a monochrome binary one-shot method was proposed by Kawasaki et al. which makes use of a projected grid pattern [6]. This elegant method leverages the intersection of vertical and horizontal projected grid-lines to produce a system of linear equations from coplanarity constraints. By searching the nearest grid-line through the sum of squared differences, ambiguity is resolved and correspondence is achieved. The computational load of this method remains too great for realtime performance, as it executes at ~ 1 fps, with most of the time used for calculating the solution of a large linear system of equations that model the intersections of the grid lines. In subsequent work, a number of clever modifications have been used to improve the processing speed and resolution of this one shot method [13].

There are a number commercially available range sensors, which are based on variations of the above-described methods, and which are targetted to different application domains. Perhaps the most widely-used triangulation sensor has been the Kinect I, Introduced by Microsoft in 2010 to enhance human interaction with video games. Marketed as an RGB-D sensor due to its ability to acquire co-registered color and depth information at each pixel, works by projecting and capturing a random dot pattern from IR procam system followed by triangulation for 3D depth reconstruction. The Kinect II was introduced in 2014, and was based on time-of-flight technology, and had superior accuracy, range, resolution and bandwidth characteristics, with a similar interface. Other commercially available range sensors include the Intel RealSense [7] and the Ensenso N35 [3], both of which use a form of active stereo, and recently a set of sensors from Zivid labs [8] which use time-multiplexed structured light projection.

The proposed approach falls under the category of dis-



(a) Projector pattern (b) Pattern a) inverse rectified

(c) Camera image of pattern a) (d) Camera image of pattern b)

(e) Image c) rectified (f) Image d) rectified

Figure 2. Procam Pair, Without (left column) and With Inverse Rectification.

crete monochrome one-shot triangulation methods. It makes use of a property of the epipolar geometry of the projector-camera pair to ease the correspondence calculation, by first inverse rectifying the binary pattern prior to projection. This inverse rectification aligns the projected features to the image epipolar lines, reducing correspondence decoding to a trivial constant-time table lookup. The method is scalable, able to produce a sparser ($\sim$4K) point cloud at a very high frame rate, or a denser point cloud at a lower rate.

3. Inverse Rectification

The objective of inverse rectification is to design a pattern to be projected onto a scene such that correspondences between the projected pattern and acquired image points can be easily and efficiently determined, with a simple table lookup. The process of inverse rectification can be appreciated through the example in Fig. 2, which shows a set of projected patterns and acquired images, both without and with inverse rectification. The baseline of the projector and camera pair align vertically in this example, with the camera positioned above the projector, rather than horizontally as is more common in stereovision systems. The impact of this vertical alignment is that the rectified epipolar lines align by column.

Fig. 2a) is the original projector pattern, showing vertical dashes. In c) the pattern from a) is projected onto the (mostly planar) scene, and viewed from the camera after radial distortion correction (but not rectification) has been applied. It can be seen that the vertical dashes are no longer perfectly parallel, as they fan out towards the top, which is to be expected as the images have not been rectified. In part e) the image from c) has been transformed through the camera's rectifying homography. The vertical dashes from the original pattern in a) are clearly not aligned with the rectified camera frame's epipolar lines. This is also to be expected, as the vertical dashes in the original pattern in a) were not aligned with the epipolar lines of the projector frame.

Fig. 2b) shows the projected pattern from a), now with the inverse rectifying projector homography applied. Part d) is the pattern from b), projected onto the scene and viewed from the camera. Part f) is the image from d) with the camera's rectifying homography applied. It can be seen that the effect of applying inverse rectification to the pattern prior to projection has been to align the initial projected pattern dashes in a) with the rectified camera's vertical epipolar lines. The vertical pattern dashes between initial pattern a) and rectified image f) thereby uniquely correspond by horizontal column. The vertical dash positions in image f) are different than those in a) due to disparity resulting from the scene's depth.

Formally, let Q be an image acquired in the camera frame of pattern P projected onto a scene. Assume that the projector and camera are aligned so that their fields-of-view substantially overlap, and that they both adhere to the standard pinhole optical model.

Rectification is the process of transforming a pair of camera frames, or a camera and a projector frame, so that their epipolar lines are parallel and align. The two rectifying transformations are homographies, and are calculated from a combination of the intrinsic parameters of each optical device, and the extrinsic parameters linking each device's frame. Following rectification, corresponding epipolar lines between the camera and projector frames will align, in this case by column.

Let H_Q and H_P be the rectifying homographies for the camera and projector frames respectively. The rectified projector image $\widehat{P}$ and camera image $\widehat{Q}$ then have the property that the (vertical) epipolar line that falls on the i^{th} column of $\widehat{P}$, corresponds to the epipolar line on the i^{th} column of $\widehat{Q}$. This alignment does not mean that the corresponding columns of $\widehat{P}$ and $\widehat{Q}$ contain identical information, but rather that corresponding points between the rectified projector and image planes fall on the same column, with the differences of their row positions (i.e. the disparities) varying linearly with depth.

Thus let projector frame point p be mapped through the projector homography to rectified point $\widehat{p} = H_P p$. Points are expressed in homogeneous coordinates, so that $\widehat{p} = (\widehat{u}_p, \widehat{v}_p, 1)$. Let p be projected onto a 3D point in the scene, which is then in turn projected onto image point q and rec-

(a) Vertically aligned projector and camera (b) Substantially overlapping fields-of-view

Figure 3. Procam system setup

tified as $\widehat{q}{=}H_Q q{=}(\widehat{u}_q, \widehat{v}_q, 1)$. The column-alignment of the two rectified frames then gives:

$$p \propto q \Rightarrow \widehat{u}_p = \widehat{u}_q \tag{1}$$

where $\propto$ indicates the correspondence of two points.

Inverse rectification exploits the property of Eq. 1 through the design of a pattern containing features that can be indexed directly by epipolar line. Minimally, this pattern can contain a single feature per epipolar line, although we show that this can be extended to multiple features per line, by applying some reasonable constraints and characteristics of the pattern.

Let $\widehat{P}$ contain a single feature per epipolar line. As the images are vertically rectified, this could be as simple as designing a pattern that has a single unique white pixel for each column, against an otherwise black background. This quality must exist in the rectified pattern $\widehat{P}$, rather than the initial unrectified pattern P. It must therefore be generated first in rectified space as $\widehat{P}$, with inverse rectification then applied to achieve the desired pattern P for projection:

$$P = H_P^{-1} \widehat{P} \tag{2}$$

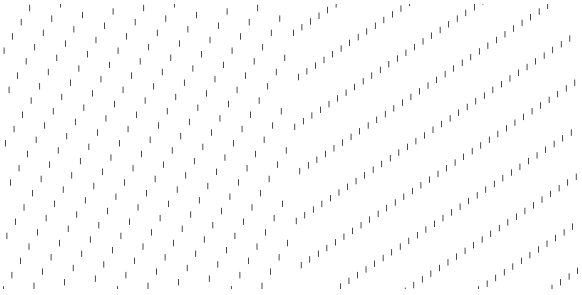
With P then projected, a feature $\widehat{q}$ extracted from rectified camera image $\widehat{Q}$ will correspond uniquely to projector feature $\widehat{p}$, by their common column index $\widehat{u}_q = \widehat{u}_p$, with the difference in their row values $\widehat{v}_q - \widehat{v}_p$ serving as disparity.

4. Pattern Generation and Detection

The simplest pattern comprises a single feature per epipolar line, which is a vertical dash as described in Sec. 4.1. The pattern density can be increased by constraining the maximum disparity range (Sec. 4.2) and making use of alternating inverse images (Sec. 4.3). Time performance can be further improved by tracking disparities across frames (Sec. 4.4).

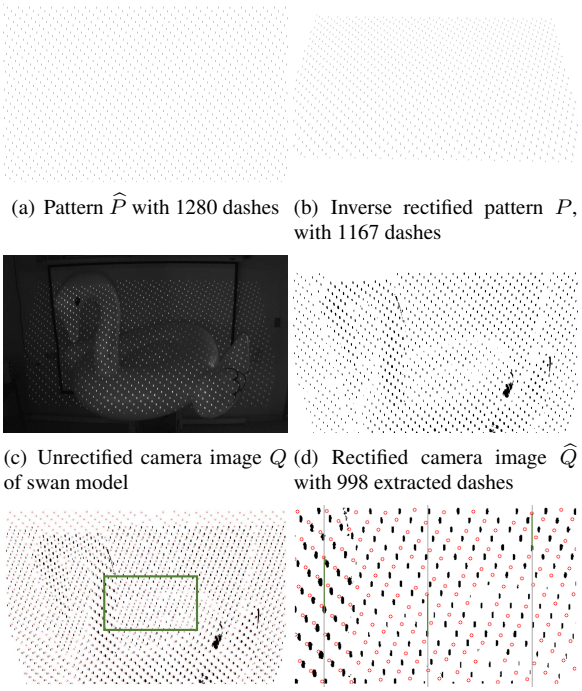
4.1. Single Dash per Epipolar Line

The objective is to generate a projector pattern $\widehat{P}$ that contains a single feature for each epipolar line. The features are simply white vertical dashes on a black background, which can be robustly and efficiently extracted, especially when image subtraction methods are applied, as in Sec. 4.3. The procam system is assumed to be vertically



(a) stride = 465 pixels (b) stride = 466 pixels

Figure 4. Pattern distributions (intensity inverted) for varying vertical strides.



(a) Pattern $\widehat{P}$ with 1280 dashes (b) Inverse rectified pattern P, with 1167 dashes

(c) Unrectified camera image Q of swan model (d) Rectified camera image $\widehat{Q}$ with 998 extracted dashes

(e) $\widehat{P}$ in red, overlayed on $\widehat{Q}$ (f) Cropped area of e), with epipolar lines (gray) and correspondences (green).

Figure 5. Single dash per epipolar line pattern

aligned, with the camera above the projector, as shown in Fig. 3a). As pattern $\widehat{P}$ lies in rectified space, the epipolar lines are themselves vertically aligned and are simply the pattern columns. This vertical alignment of the projector and camera exploits widescreen aspect ratios, which more columns than rows. This alignment therefore accommodates a greater number of pattern points than would be possible if the projector and camera were horizontally aligned, as is more typical in stereovision systems. The fields-of-view of the projector and camera are oriented to substantially overlap, as shown in Fig. 3b), where the striped projector pattern lies within the camera image, both of which have a resolution of 1280x800.

The pattern is generated by inscribing white dashes on a

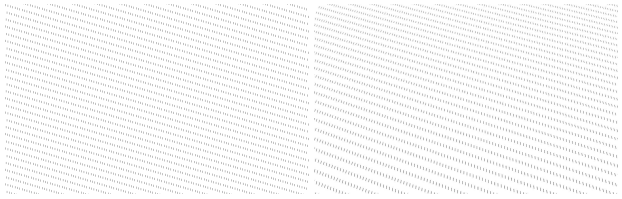
black background, one per column. The dashes themselves are 9x1 pixels in size, which were the smallest projected vertical dash features that could be reliably extracted from an acquired camera image of similar resolution, following inverse rectification of the projected pattern and radial undistortion and rectification of the acquired image. After calibration, the average reprojection error between corresponding projector and camera epipolar lines was slightly less than one pixel, and so dashes were deposited with large vertical offsets (i.e. *strides*) between adjacent columns, to reduce the possibility of correspondence errors with a neighboring dash during disparity calculation.

It is desirable for the dashes to be distributed as uniformly as possible across the pattern, which is impacted by the vertical stride between horizontally adjacent dashes. Examples of pattern sections (with inverted intensities for easier visualization) for dash distributions over two strides are illustrated in Fig. 4. It can be observed that stride 466 leads to a more linear pattern, whereas 465 results in a more uniform dash distribution.

The complete projector pattern $\widehat{P}$ containing 1280 dashes and with a vertical stride of 456 pixels between horizontally adjacent dashes, is illustrated (with intensities inverted) in Fig. 5a), with inverse rectified version P in Fig. 5b). Resizing of the image during inverse rectification effectively reduces the image size, resulting in only 1167 dash features in P. Fig. 5c) shows the unrectified camera's view Q of P, and Fig. 5d) shows the 998 dash features extracted from rectified camera image $\widehat{Q}$ using simple thresholding. Fig. 5e) shows the projected pattern $\widehat{P}$ overlayed on the dashes extracted from $\widehat{Q}$. The dash positions of $\widehat{P}$ have been rendered as red circles, and a rectangular area is highlighted in green. Finally, Fig. 5e) shows this highlighted area, cropped and zoomed. Three epipolar lines are rendered in gray, and corresponding features along these lines are highlighted as a green segment, the length of which indicates the disparity of these features.

Projector pattern $\widehat{P}$ and its inverse rectified version P need to be generated only once, during preprocessing. The resuling dash positions are extracted from $\widehat{P}$ and stored in a simple list L, the size of which is equal to the column rank of $\widehat{P}$. Each entry of L is indexed by epipolar line (i.e. column value u), and stores the row value v of the center of the unique pattern dash $\widehat{p}_i(\widehat{u}_{p_i}, \widehat{v}_{p_i})$ that resides on that epipolar line, i.e. $L[\widehat{u}_{p_i}] = \widehat{v}_{p_i}$

At runtime, camera image Q is acquired and rectified into $\widehat{Q}$, and the set of N camera image dash positions $\left\{\widehat{q}_j(\widehat{u}_{q_j}, \widehat{v}_{q_j})\right\}_1^N$ extracted. As image $\widehat{Q}$ is binary, it is straightforward to identify the position $\widehat{v}_{q_j}$ of the single sequence of white pixels in each column $\widehat{u}_{q_j}$ by applying a linear search. As the pattern dash is larger than a single pixel, and the camera and projector resolutions are similar, the height $\widehat{v}_{q_j}$ of any dash $\widehat{q}_j$ will be spread over a few



(a) Rectified pattern $\widehat{P}$, $N = 3390$ dashes (b) Inverse rectified P of $\widehat{P}$, resized and cropped, $N = 2163$ dashes.

Figure 6. Disparity bound pattern with on average 5.3 dashes per epipolar line, resolution $= 1280 \times 800$, $D_m = 151$.

(typically $h = 9$) pixels. In practice a two-phase sub-linear search is therefore more efficient, wherein only every k^{th} pixel, $k < h$, is visited during the first phase to fall somewhere on the dash, and then the dash position is fine-tuned in the second phase to find its center $\widehat{v}_{q_j}$. To further improve efficiency, the first phase need only search along the direction of increasing disparity, and the search can start at the row location stored in the pattern dash list, at which position disparity will be zero.

Once extracted, the disparity d_j of dash $\widehat{q}_j$ is determined by simply indexing the list:

$$d_j = L[\widehat{u}_{q_j}] - \widehat{v}_{q_j} \tag{3}$$

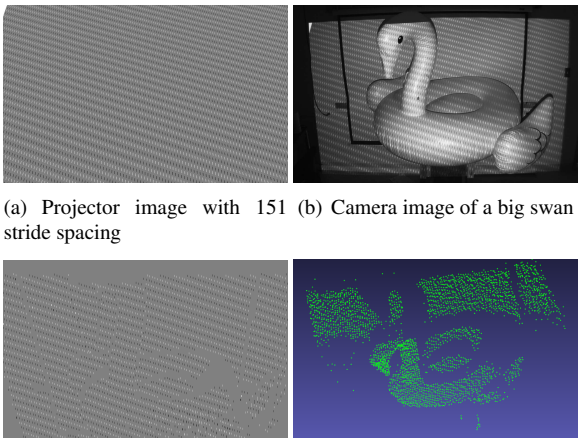
This correspondence decoding is therefore constant time and very efficient, requiring a visit of only $\sim H/k$ pixels per column of height H, and a simple table lookup. The disparities are then converted to 3D points in the usual way, accessing the procam pair's intrinsic and extrinsic parameters, at the cost of a few floating point operations.

4.2. Disparity Bounding

The most direct way to increase the density of projected dashes per frame is to simply allow P to contain more than one dash per epipolar line. P is thus designed so that each column contains a number of dashes, which are uniformly spaced and separated by a maximum disparity of D_m pixels. At runtime, the first phase of the correspondence search for each rectified camera image dash $\widehat{q}_j$ is limited to a range of D_m pixels. The value D_m therefore presents an upper bound to the range of disparity that can be determined for each dash.

The disparity d_j of a point has an inverse relationship $d_j = (B \times f)/Z_j$ to its depth Z_j, where B is the baseline and f is the effective focal length of the optical pair. The maximum disparity bound D_m in this way corresponds to the offset of the near sensing plane of the imaging frustum. So long as sensed points fall beyond this near plane, there can be no ambiguities among the correspondences established from the multiple dashes that share an epipolar line.

An example of such a pattern is illustrated in Fig. 6, which has a disparity bound of $D_m = 151$, yielding 5.3 dashes on average per epipolar line. This disparity bound



(a) Projector image with 151 stride spacing

(b) Camera image of a big swan

(c) Extracted camera features using b) and its inverse, with background filtered and stripes rerendered for clarity

(d) 3D point cloud of swan, rotated

Figure 7. Example of Dual Pattern Subtraction applied to a scene

translates to theoretical near sensing plane of 1.12 m, and a far plane of 170.9 m, under the current configuration of the calibrated optical system. Practically, the near was $\sim$2.3 m, and the far plane $\sim$ 15 to 25 m, limited by the brightness of the projector and the lighting and reflectance properties of the scene. As the dashes are packed closer together, and therefore the vertical stride between horizontally adjacent dashes necessarily compressed, only every other column is used to avoid potential errors with neighboring dashes when establishing correspondences. This results in a total of 2163 dashes in pattern $\widehat{P}$, a 185% increase in dash density per frame, after resizing and cropping P following inverse rectification.

4.3. Dual Pattern Subtraction

Feature density can be further increased by making use of two dual patterns comprising three gray tone intensities. Let $\widehat{P}_A$ be a pattern designed as previously described in Sec. 4.2, except with white dashes (pixel intensity $I = 255$) spaced at disparity bound D_m against a gray ($I = 128$) background, in place of the previous black ($I = 0$) background. Further, let $\widehat{P}_A$ contain an equal number of black dashes spaced exactly at the midpoint of each white dash along each column. Finally, let $\widehat{P}_B$ be the inverse of $\widehat{P}_A$, such that the white dashes of $\widehat{P}_A$ are the black dashes of $\widehat{P}_B$, and vice-versa. $\widehat{P}_A$ and $\widehat{P}_B$ are thus considered to be dual patterns: They contaon the same information, except with intensities inverted. The dual patterns are calculated during preprocessing, along with their respective inverse rectified dual patterns P_A and P_B.

At runtime, the projected pattern alternates each frame between P_A and P_B, and the current camera image Q_t acquired at frame t is subtracted from image Q_{t-1} acquired at frame $t-1$. Irrespective of whether the most recent projected pattern was P_A or P_B, the image subtraction results in high peaks (~ 255) where the white dashes of Q_t overlap with the black dashes of Q_{t-1}, and low troughs (~ -255) where the black dashes of Q_t overlap with the white dashes of Q_{t-1}. Gray regions between the dashes will subtract to a value of zero. An example of this approach is shown in Fig. 7.

This approach has the benefit of doubling the extractible pattern dash resolution, without reducing the disparity bound. As the locations of the white and black dashes produce peaks and troughs in the subtracted image, correspondences with white dashes can be distinguished as either occuring in $\widehat{P}_A$ or $\widehat{P}_B$. In this way, each pattern is treated simultaneously, as if they were independently acquired images, each with disparity bound D_m.

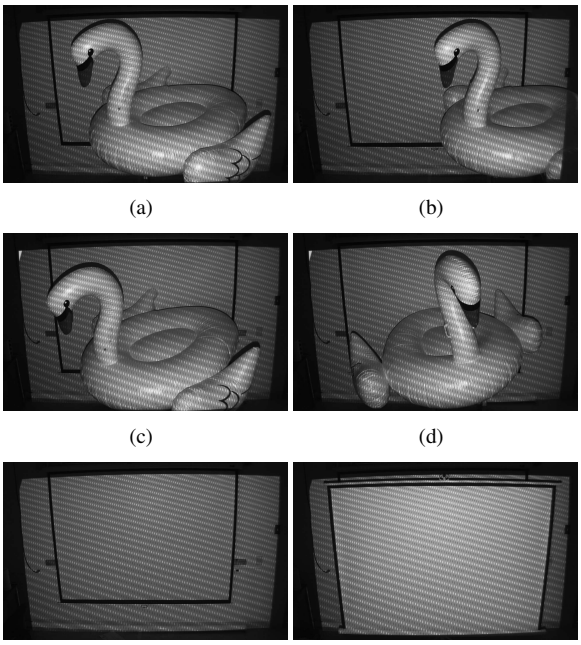
Another benefit of this approach is that the use of subtraction between consecutive image frames increases the signal-to-noise ratio (SNR) of the image dash extraction. It is unlikely for peaks or troughs in the subtracted image to occur at the neutral gray pixels, which are common to both images. The dash positions are therefore more robustly identified than if a simple threshold were applied, which is especially important when the scene contains non-cooperative colored surfaces of varying albedo. This increase in SNR is the main advantage of the subtraction of dual three-tone images, compared with simply shifting the positions of white dashes on a black background over a consecutive image pair.

The use of dual images over two successive frames maintaines the full frame rate of the method, as each new frame produces a new full set of $2N$ dashes. It does, however, change the interpretation of the instance in time that the surface was sampled to produce the point set. As both Q_t and Q_{t-1} contribute to the extracted dash locations, any motion of the scene surface will be integrated over this time step.

4.4. Disparity Caching

A further increase in efficiency can be realized by caching the disparity values calculated for each dash in the previous frame. These cached disparity values are then used for the respective dashes as the starting positions of the phase one search, rather than the corresponding projector dash position as previously described. In cases where the disparity has changed little from the previous frame's value, this will cause early termination of the first phase of the search, reducing the number of visited pixels and the resulting computational expense of the search.

In addition to caching the disparity values, the 3D point cloud generated from the previous frame is also cached. In cases where the dispartity has not changed between frames, this further increases efficiency, as there is no need to recalculate these 3D points from the disparity values.



(a) (b) (c) (d) (e) (f)

Figure 8. Sample images from (a-b) Swan Translate, (c-d) Swan Rotate, and (e-f) Plane Forward datasets.

5. Experiments

A set of experiments were conducted to characterize the performance of the proposed method. Unless otherwise specified, all tests used dual pattern subtraction with 2 phase search and disparity caching, and the imaged scenes contained substantially Lambertian matte white surfaces. All timing results are averaged over 1000 iterations for consistency. The machine used for all the tests was an i9 7920X CPU @2.90GHz with 16GB RAM, running Ubuntu 18.04.2 LTS, with OpenCV version 4.1.0. The code was written using C++ in a parallelized multi-threaded implementation using 24 threads.

We created three datasets of streams of images for our experiments, namely *Swan Translate*, *Swan Rotate*, and *Plane Forward*. Swan Translate is a set of 12 image pairs where the swan is moving across the scene. Swan Rotate contains 13 image pairs, with the swan rotated 360 degrees about the y axis, while Plane Forward, has 5 image pairs, in which the projector screen moves towards the camera in 40 cm intervals, from 3.9 m to 2.3 m. A few sample images from these datasets are shown in Fig. 8.

The overall time taken by the method using Swan Translate detaset is tabulated in Table 1. The majority (76%) of the execution time is consumed by the OpenCV *remap* function, which radially undistorts and applies the rectifying homography to the camera image. The disparity and depth calculations were comparatively efficient.

The times reported in the subsequent tables are only for the disparity and depth calculations, and do not include the time for the OpenCV *remap* and *threshold* functions. Execution time with and without disparity caching are shown in Table 2. It can be seen that the performance enhancement due to caching is heavily dependent on the scene. In general, disparity caching does in each instance produce a faster average runtime, although in the case of the Swan Rotate dataset, this performance improvement was minimal.

	Time (μs)			
Function	avg	max	min	std. dev
remap + threshold	1159	1530	1018	34.6
disparity + depth	368.8	1884	343.3	48.8
total time	1528	2984	1368	59.6

Table 1. Time taken per image for remapping and disparity extraction based on 4315 dashes with 60.1 average disparity

The execution times on Swan Translate dataset of the various enhancements to improve time performance discussed in the paper are summarized in Table 3. It can be seen that the single dash per epipolar line outperforms the disparity bounding, and that the average disparity in the disparity bounding test case was 50% larger than the single dash test case. As expected, the greatest improvement in speed was realized by using the 2 phase search and caching method, resulting in a bandwidth of 11.7 M points per second (pps).

5.1. Accuracy of Extracted Points

To access the effectiveness of inverse rectification applied on the projector pattern prior to projection, we performed an experiment using plane fitting to calculate the quantity and quality of extracted depth points with and without inverse rectification using single dash per epipolar line. The accuracy of the extracted points was measured by fitting an imaged plane to the disparity values generated from a large planar surface which was positioned at two different intervals from the procam pair (3.3 and 3.9 m). The plane of best fit was found by first estimating the plane equation using 100 RANSAC iterations, and then calculating the best fit plane to the inliers.

The metric we use for measuring depth accuracy is the percentage of pixels whose disparity error is greater than threshold t for t=1,2,3,4. Table 4 shows the results of plane fitting averaged over the two scenes. The advantage of inverse rectification is evident from the results. Compared with no prior rectification, the proposed method is able to capture much higher percentage of points, the average disparity is much closer to the ground truth disparity (39 pixels), and the percentage of bad pixels is much lower for all thresholds.

dataset	average disparity (px)	average δ disparity (px)	with cache runtime (μs) avg	max	min	std. dev	without cache runtime (μs) avg	max	min	std. dev
Swan Rotate	59.6	1.1	367.2	585.1	344.2	11.4	415.5	513.6	397.2	7.9
Swan Translation	60.1	1.2	368.8	1884	343.3	48.8	370.1	415.9	347.4	7.8
Plane Forward	59.3	4.1	365.9	474.1	336.8	10.6	405.5	545.1	385.8	10.4

Table 2. Runtime with and without caching on different datasets.

method	avg disparity	point count	pps (Million)	Total Time (μs) avg	max	min	std. dev
single dash per epipolar line	44.0	1241	8.5	146.8	245.4	134.9	9.5
disparity bounding	63.3	2174	8.1	269.8	324.3	208.1	11.7
dual pattern subtraction	60.1	4315	9.2	469.6	2017	420.0	76.5
2 phase search + caching	60.1	4315	11.7	368.8	1884	343.3	48.8

Table 3. Comparison of performance for variations of proposed methods.

		prior inverse rectification	no prior rectification
projected points		2225	2626
captured points		2212.5 (99.4%)	968.5 (36.9%)
bad points	t=4	12 (0.5%)	693.5 (71.6%)
	t=3	80.5 (3.6%)	720.5 (74.4%)
	t=2	233 (10.5%)	748.5 (77.3%)
	t=1	460 (20.8%)	792.5 (81.8%)
average disparity		38.9	54.9
std. dev. wrt. avg.		6.5	23.4
std. dev. wrt. GT		5.4	45

Table 4. Plane fit accuracy data

sensor	technology	resolution	fps	pps
Kinect V1	active IR stereo	640×480	30	9.2M
Kinect V2	time of flight	512×424	30	6.5M
Intel RealSense D435	active IR stereo	1280×720	90	83M
Proposed Sensor	active visible stereo	1280×800	2720	11.7M

Table 5. Comparison of our proposed range sensor with existing ones

5.2. Comparison with Consumer-Level Range Sensors

A comparison of the proposed sensor with existing consumer-level range sensors is presented in Table 5. Once again considering only the time to calculate disparities and depth, the proposed method has a much higher potential frame rate. This is an interesting feature, as there are many applications (such as tracking) where a high frame rate is desired with sparser data. Table 5 also shows that our bandwidth, as measured in pps, is higher than the Kinect I and Kinect II, but less than the Intel RealSense D435. It must be noted, however, that RealSense D435 uses 3 cameras (2 IR + 1 visible) and 1 projector whereas our setup uses only a single camera and projector. In addition, all 3 commercial systems are optimized at the hardware-level, whereas we are combining different out-of-shelf components, using a general purpose computer and only high level coding. With further hardware optimization, the bandwidth of the proposed method would be expected to increase further.

6. Discussion and Conclusion

In this paper, we presented a novel technique to establish efficient, robust and accurate point correspondences between a procam image pair. The core idea is to exploit epipolar geometry to generate a pattern of small dashes in the rectified procam image pair such that each dash lies on a unique epipolar line. An inverse rectification transform is applied on the projector image before projection in order to reduce point correspondence search to a simple table lookup. On top of this simple method, we introduced multiple enhancements such as two-phase search, disparity bounding, dual pattern subtraction, and disparity caching, and showed experimentally that these enhancements indeed increase the speed, resolution, and robustness of the resulting depth extraction.

The method was demonstrated as being very efficient. Despite being implemented on a general purpose CPU system, it had a higher bandwidth (at 11.7M pps) than the Kinect I (2.3M pps) and Kinect II (6.5M pps), although not the Intel RealSense (83M pps). In future work, we will focus on increasing resolution per frame, as well as increasing computational efficiency through a highly parallel and optimized implementation on an FPGA or GPU platform.

Acknowledgements

The authors would like to acknowledge Epson Canada, the Natural Sciences and Engineering Research Council of Canada, and the Ontario Centres of Excellence, for their support of this work.

References

[1] J. Battle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition*, 31(7):963–982, 1998.

[2] L. Ekstrand, N. Karpinsky, Y. Wang, and S. Zhang. High-resolution, high-speed, three-dimensional video imaging with digital fringe projection techniques. *Journal of visualized experiments : JoVE*, 12 2013.

[3] Ensenso. Ensenso stereo 3d cameras, 2019. https://www.ensenso.com/ [Online; accessed 6-June-2019].

[4] J. Geng. Structured-light 3d surface imaging: a tutorial. *Adv. Opt. Photon.*, 3(2):128–160, Jun 2011.

[5] M. Hansard, S. Lee, O. Choi, and R. Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. 10 2012.

[6] Hiroshi Kawasaki, Ryo Furukawa, Ryusuke Sagawa, and Yasushi Yagi. Dynamic scene shape reconstruction using a single structured light pattern. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[7] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2017.

[8] Z. Labs. Bin-picking, logistics, machine tending 3d color camera - zivid, 2019. http://www.zivid.com/ [Online; accessed 7-June-2019].

[9] S. Lei and S. Zhang. Flexible 3-d shape measurement using projector defocusing. *Opt. Lett.*, 34(20):3080–3082, Oct 2009.

[10] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 131–144, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[11] J. T. Martin D. Altschuler, Bruce R. Altschuler. Laser electro-optic system for rapid three-dimensional (3-d) topographic mapping of surfaces. *Optical Engineering*, 20(6):953 – 961 – 9, 1981.

[12] J. Posdamer and M. Altschuler. Surface measurement by space-encoded projected beam system. *Computer Graphics and Image Processing*, 18:1–17, 01 1982.

[13] R. Sagawa, R. Furukawa, and H. Kawasaki. Dense 3d reconstruction from high frame-rate video using a static grid pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1733–1747, 2014.

[14] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43:2666–2680, 08 2010.

[15] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[16] l. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured lightand multi-pass dynamic programming. pages 24–37, 01 2002.