Projected Texture Stereo

Kurt Konolige
Willow Garage, Menlo Park, USA
konolige@willowgarage.com

***Abstract*— Passive stereo vision is widely used as a range sensing technology in robots, but suffers from *dropouts*: areas of low texture where stereo matching fails. By supplementing a stereo system with a strong texture projector, dropouts can be eliminated or reduced. This paper develops a practical stereo projector system, first by finding good patterns to project in the ideal case, then by analyzing the effects of system blur and phase noise on these patterns, and finally by designing a compact projector that is capable of good performance out to 3m in indoor scenes. The system has been implemented and has excellent depth precision and resolution, especially in the range out to 1.5m.**

I. Introduction

Passive stereo vision is an important 3D sensing technology for object recognition and manipulation at short range (30cm – 300cm). It produces dense point clouds with excellent depth resolution, at high frame rates, and can deal with moving objects. Because stereo systems use standard imaging components, they are also potentially small, low cost, and low power. However, textureless surfaces cannot be matched by stereo, and produce dropouts in the stereo results (see Figure 1, top). One method for removing dropouts is to paint the scene with projected light (Figure 1, middle). But finding the optimal texture is a complicated problem, influenced by characteristics of both the projector, the pattern, and the stereo cameras. This paper addresses this problem from a theoretical and practical standpoint, with the end goal of developing a compact, high-performance projected texture system. The contributions of this paper are

- A method for generating near-optimal patterns using techniques of Hamming codes and simulated annealing.
- Analysis of effects of phase and blur of the projector and camera system.
- Design and construction of a compact projected texture stereo system, and its experimental validation.

A. Projected Texture Systems

Block-matching stereo computes range by triangulation, matching a small block in one image against a range of blocks in the other [1]. The best match generates the range to the center of the block.

Many complex stereo algorithms attempt to solve this problem by "filling in" the low-texture areas using regularization methods that propagate information from other areas [2], [3]. A more direct method is to simply project a highly-textured pattern [4], [5]. In Figure 1 middle, a projected texture now covers the scene, and good stereo matches are found almost everywhere (the reflection of the cup in the table produces a bad match which is filtered).

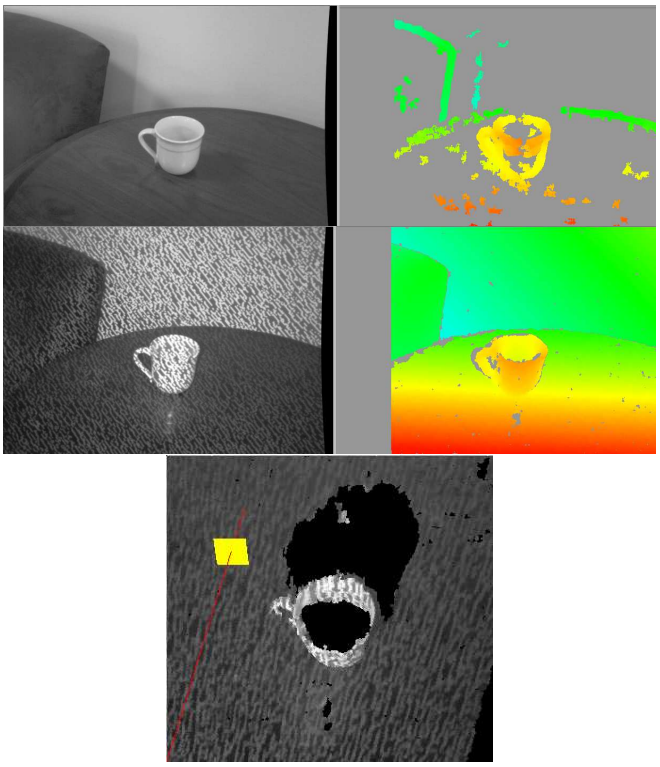

Fig. 1. Effects of texture on stereo. Top is a difficult scene, with dark furniture and white reflective cup; right side is a color-coded disparity image, showing that stereo is difficult with no texture. Middle is the same scene and stereo algorithm with our projected texture system. Bottom is a reprojection of the cup detail from an overhead view - note that the projected texture is painted on the 3D points. At distances under a meter, the system has an error of less than 2mm in all directions.

But what pattern should be projected? In the first experiments, we used a random black and white texture, in some cases resulting in pockets of dropouts (see Figure 9, leftmost image). What's going on is that the pattern is too self-similar in certain places, and the match is ambiguous over the stereo search range. One way to approach the problem is from ideas in coding theory. To transmit letters from an alphabet, they are encoded in such a way as to differ from each other as much as possible. In a similar manner, one could try to make the matching blocks of stereo as dissimilar as possible across the match region. The first part of this paper pursues this idea using lexicographic codes, and shows how to construct patterns that are better than the best current

alternative, non-recurring De Bruijn patterns [4]. A further technique is developed using simulated annealing, and shown to be even better.

While there is some research on good projected textures for stereo, it is invariably done without taking the imperfections in the projector and camera into account, or the resolution and phase differences between pattern and image. It turns out that these factors are large determiners of the quality of matching – patterns that are good in the ideal case degrade under non-ideal conditions. In the second section, the process of simulated annealing is extended to find good patterns under realistic conditions, and show how their performance dominates the ideal patterns, both in simulation and with an experimental setup.

Finally, we develop a compact fixed-pattern projector with simple optics, capable of texturing an indoor scene at distance up to 3m, even under bright daylight conditions. There are several practical advantages to using a texture projector with stereo. First, it requires only that the stereo pair be calibrated, which is already easily achieved with current stereo systems. Second, it supplements natural texture: the projector need not overcome ambient light and poor surface reflectance, it just has to add enough texture to featureless surfaces to enable block matching to work. These features make projected texture stereo much more robust and suitable for real-world applications than systems that must view and reconstruct a structured light pattern.

B. Related work

There are alternatives to stereo for close-range 3D sensing, but they lack some of its advantages. Flash ladars [6] have poor depth and spatial resolution, and have non-gaussian error characteristics that are difficult to deal with. Line stripe systems [7] have the requisite resolution but cannot achieve 10 Hz operation, nor deal with moving objects. Structured light systems [8] are achieving reasonable frame rates and can sometimes incorporate motion, but still rely on expensive and high-powered projection systems, while being sensitive to ambient illumination and object reflectance.

An interesting and early technology is the use of stereo with *unstructured* light [9]. Even with projected texture, block-matching stereo still forces a tradeoff between the size of the match block (larger sizes have lower noise) and the precision of the stereo around depth changes (larger sizes "smear" the depth boundary). One possibility is to use smaller matching blocks, but reduce noise by using many frames with different projection patterns, thereby adding information at each pixel. This technique is known as *Space-time Stereo* (STS) [4], [10], [11]. It produces outstanding results on static scenes and under controlled illumination conditions, but moving objects create obvious difficulties. While there have been a few attempts to deal with motion [11]–[13], the results are either computationally expensive or perform poorly, especially for fast motions and depth boundaries. In our case we use just a fixed pattern, and perform stereo only in the spatial domain.

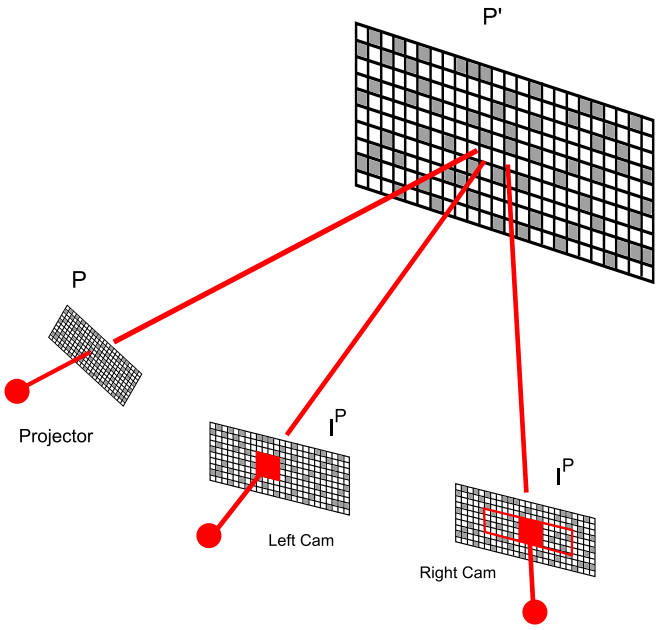

Fig. 2. Projector / stereo camera system. A pattern P is projected onto a surface to produce P', which is imaged by a left and right camera. For stereo matching, the small red block in the left image is matched against a range of blocks in the right image at the same vertical offset, indicated by the outlined rectangle.

Several recent papers address the problem of finding projected patterns that are not self-similar over some range. Molinier et al. [5] randomly generate binary pixels to incrementally fill a pattern, and test that each 5x5 block is unique. Lim [4] uses the technique of De Bruijn sequences to find similar patterns. However, these patterns are only *satisficing*, that is, they are not necessarily patterns that are maximally dissimilar. This paper improves on those results, showing that patterns created with Hamming codes dominate De Bruijn and random patterns. We also address problems of resolution, phase and blur in practical systems, which to our knowledge have not been considered in the projected texture literature.

II. Ideal Block-Matching Patterns

We consider a projector and stereo cameras configured to be as nearly coincident as possible (see Figure 2). For simplicity, the focal length of the projector and cameras are similar, so that at any distance the projected pattern appears to be the same size in the camera images. The pattern P is a grid of black and white squares, projected by a compact device with a fixed pattern that we design (Section IV). When it is seen by a camera, it produces an image I^P. In this section, we take $P = I^P$, to study the properties of patterns abstracted from the characteristics of the projector/camera system. In the next section we introduce more realistic image transfer functions that incorporate phase and blur noise.

A. Block-Matching Stereo

To test the effect of the pattern on stereo, we use a strictly local block-matching stereo algorithm (Figure 3). A square block of size $n \times n$ in the left image is matched against

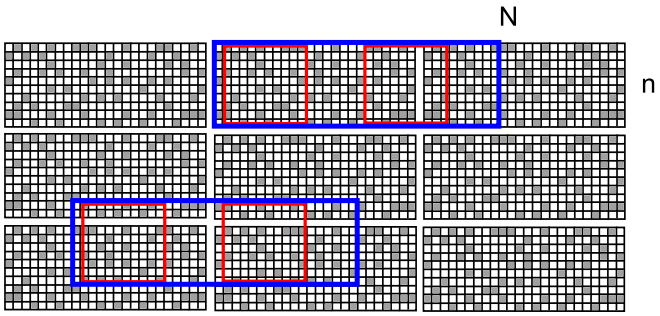

Fig. 3. Block matching and pattern repetition. A pattern of size $n \times N$ is tiled horizontally and vertically. The minimum matching distance of any block in the upper blue rectangle is d; two such blocks are shown. No matter where the rectangle is placed, the same minimum matching distance will hold, e.g., in the lower left blue rectangle.

N other blocks in the right image, along the same scanline (epipolar geometry is assumed for the images); N is the search range for stereo disparities. The match score is the sum of absolute differences of the corresponding pixels in the blocks (SAD). There are many other correlation measures that could be used, such as Sum of Squared Differences (SSD) or Normalized Cross-Correlation (NCC), or even non-parameteric measures such as Census (see [14]). SAD is reasonable in curbing the effect of outliers. Standard correlation measures such as NCC or SSD would exaggerate the differences between the various patterns reported here.

Since the two images view the same pattern, within every rectangle of P whose size is $n \times N$, each $n \times n$ block must differ from every other one, and we can measure how good the stereo matching is by looking at the minimum difference. The bigger the minimum difference, the better the pattern for stereo. We write:

$$S(P) = \min_{i \neq j} \text{SAD}(P, i, j), \tag{1}$$

where P is a pattern, $i, j \leq N$ are indices of the blocks, and $\text{SAD}(P, i, j)$ is the SAD score of blocks i and j of pattern P. A good score occurs when every block is maximally at least a minimum SAD distance from every other block. This idea is similar to Hamming codes, where each code is a minimum Hamming distance from all other codes, and we exploit the connection to find good patterns in the next section.

The reason we use the score of every block against every other one, rather than just the first block, is to allow the pattern to repeat. Given a score $S(P)$ for a pattern P of size $n \times N$, we can construct any size pattern by just repeating P horizontally and vertically. It is easy to show that the order of rows of P has no effect on $S(P)$, so any $n \times N$ image rectangle in a vertical stack of patterns P will have the exact same rows as any other, and hence the same score $S(P)$. Horizontally, it doesn't matter where we place the $n \times N$ rectangle, since all blocks are at least $S(P)$ different from every other block along a row (Figure 3).

B. Minimum Hamming Distance Patterns

Each $n \times n$ block i in the pattern is a binary vector v_i of size n^2. In this case, SAD computes the Hamming distance between vectors $\text{SAD}(i, j) = \sum v_i \oplus v_j$. If v_i and v_j were independent for all $i \neq j$, we could use the theory of Hamming codes to find a set of vectors with a minimum Hamming distance d (all vector pairs differ by at least d) [15].

Unfortunately, the vectors aren't independent: v_{i+1} incorporates $n(n-1)$ elements of v_i. Instead, we note that the column vectors c_i of the $n \times N$ pattern can be chosen independently. If we choose these vectors to have a minimum Hamming distance d, then each block is guaranteed to differ by at least d from every other block.

The problem then becomes: what is the maximum Hamming distance for a set of N binary vectors of size n? Although the problem in general is hard, a class of codes known as "lexicographic codes," or "lexicodes," produces near-optimal codes with a simple greedy algorithm [16]. For a given d and n, start with the set $L_n^d = \{0\}$. Using dictionary ordering of vectors (i.e., $000, 001, 010, 011, \ldots$), find the next vector of at least distance d from all vectors in L_n^d, and add it to L_n^d. The length of sequences produced by varying d from 1 to n are all powers of 2; for example, we get the following sequence for $\| L_7^d \|$: $128, 64, 16, 4, 2, 2, 2$.

To construct an $n \times N$ pattern from the set of column vectors of L_n^d, we add all vectors, repeating if necessary until the pattern is filled, and then perform a random shuffle. For example, with $\| L_7^3 \| = 16$ and $N = 128$, we add the set 8 times, and then shuffle it. For the pattern length $N = 128$, for each value of n from 3 to 15, we constructed Hamming Code patterns in this manner for all possible d's, ran each 100,000 times, and picked ones with the best $S(P)$ for each n. The results are in Figure 4. As might be expected, the graph shows a quadratic behavior as the size of the blocks increases by the square of the side. Surprisingly, even though there are only 8 vectors of length 3, it is possible to find a pattern of length 128 that has non-zero $S(P)$.

C. De Bruijn and Random Vectors

We compared the Hamming Code method against two methods found in the literature: random selection and non-recurring De Bruijn sequences [4]. De Bruijn sequences of length s over an alphabet A ($B(s, A)$) contain all subsequences of length s exactly once. For our purposes, A are column vectors c_i of the pattern, and we replace A by the vector size n. If we set $n = 2$, we are guaranteed that every block in the sequence is unique, since no two blocks can contain the same subsequence. A non-recurring sequence $NB(s, n)$ is a sequence where no two neighboring vectors are the same. Note that *NB* does not try to optimize the separation between vectors, unlike the case with Hamming codes.

In a manner similar to Hamming codes, we construct patterns P using the vectors from $NB(2, n)$, choosing the next vector at random to satisfy the De Bruijn conditions. For each n, we do this 100,000 times, and choose the best

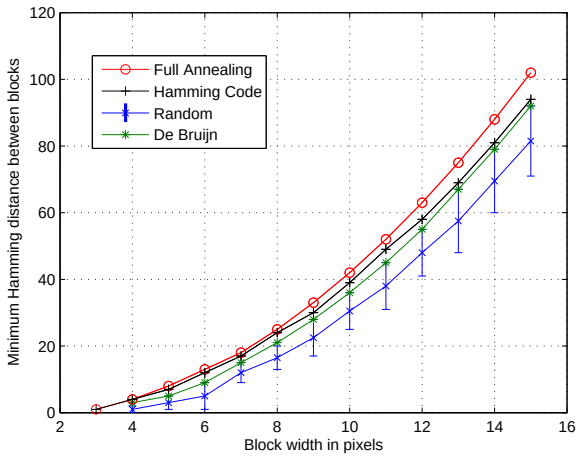

Fig. 4. Graph of $S(P)$ for patterns with varying block widths, using several different algorithsm. A matching distance of $N = 128$ was used.

response $S(P)$. The results, summarized in Figure 4, are all slightly worse than the Hamming codes. Data for the case $n = 3$ is missing, since no sequence $NB(2, 3)$ exists with length 128.

Additionally we constructed random patterns of size $n \times N$ for comparison. Again we ran a set of 100,000 tests to find the best such patterns for each n; we also show the range of generated patterns. Surprisingly, the best random patterns do as well as the De Bruijn codes, but on average the random patterns perform poorly. This explains why a single random pattern does not fare well in comparisons in the literature.

D. Simulated Annealing

The results from random patterns above indicate that it may be possible to find good patterns by search. Starting from a random pattern, we use simulated annealing [17] to search the large space of possible patterns, using the cost function $S(P)$. At any point, we find pairs of blocks that have the minimum Hamming distance. We randomly choose one of these pairs, and swap two random dissimlar pixels. The change is accepted with probability $\exp^{-\Delta S/T}$, where ΔS is the change in the score, and T is a "temperature" that goes to zero with increasing iterations. The minimum Hamming distance $S(P)$ is actually too coarse a scoring function, since many pairs may be at the minimum distance. To compensate we include in the score the fraction of block pairs at the minimum distance.

For each n, we run simulated annealing for 100,000 iterations with 100 random restarts, and choose the best result. Figure 4 shows that annealed patterns do significantly better than Hamming distance, especially at larger n.

III. Imaged Patterns

Patterns that are good under ideal imaging conditions do not fare well when phase and blur noise are introduced. In this section we continue the use of simulated annealing to find good patterns under these conditions.

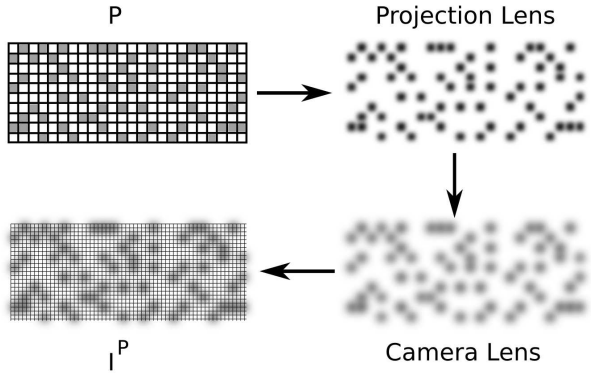

Fig. 5. Pattern as viewed by the cameras. The projected pattern is blurred by the optics of both the projector and the camera, and then re-sampled by the camera imager.

A. Resolution

In general the pattern resolution will differ from the image resolution – digital projectors have typical resolutions of 1024x768, while consumer-grade cameras can have 10 megapixels. The difference in resolutions is usually ignored in the projected texture literature, but is critical in designing good patterns for projection. Figure 5 shows the superposition of the image pixels on a projected pattern (we assume that the image has at least as good a resolution as the pattern). The ratio of pixel sizes is defined as:

$$\alpha = \frac{\text{pixel width of pattern}}{\text{pixel width of image}} \qquad (\geq 1) \qquad (2)$$

The ratio α defines the block size n_p of the pattern, given a block size n of the image:

$$\begin{aligned} n_p &= \lceil n/\alpha \rceil \\ N_p &= \lceil N/\alpha \rceil \end{aligned}$$

For example, with an image search range of 7×128, and a ratio of $\alpha = 2$, the corresponding repeating pattern is 4×64.

B. Phase and Blur

When projecting and viewing a pattern, the pattern and image can be offset by an arbitrary amount, introducing phase noise. For example, Figure 6 shows a pattern that is sampled at 1/2 pixel offset in the image (red grid). The original pattern on the left is mostly degraded to gray, with only a few black and white pixels.

We model both phase noise and blur by introducing a transfer function from the pattern to image pixels. Assume that the 0,0 pixel of the pattern is aligned with the 0,0 pixel of the image, with an offset given by x, y. We blur the pattern P by convolving with a gaussian with standard deviation σ to produce a continuous pattern $P(\sigma)$. The value of an image pixel at u, v is the average intensity of its footprint in $P(\sigma)$:

$$I_{x,y}^{P(\sigma)}(u, v) = \int_{A(u+x,v+y)} P(\sigma) dA \qquad (3)$$

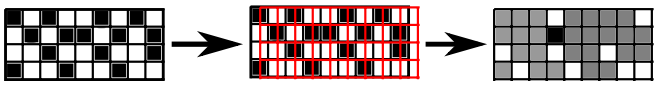

Fig. 6. Phase noise introduced by sampling at non-grid points.

where $A(u, v)$ is the area of $P(\sigma)$ taken by the image pixel at u, v. The pattern $P(\sigma)$ models the total system blur, to which both the projector and camera optics contribute. In practice, the integral of Equation 3 is computed by placing a fine grid over P, convolving with a discrete gaussian, and then summing up over the image pixel area. $I_{x,y}^{P(\sigma)}$ indicates the image formed by taking $I_{x,y}^{P(\sigma)}(u, v)$ for every u, v at a fixed offset x, y.

The scoring function $S(P)$ of Equation 1 can be modified to take phase into account, by minimizing over every possible displacement of less than 1 image pixel in the horizontal and vertical directions.

$$S^+(P(\sigma)) = \min_{0 \le x,y < 1} \min_{|i-j|>1} \mathrm{SAD}(I_{x,y}^{P(\sigma)}, i, j). \quad (4)$$

The scoring function S^+ incorporates both phase shifting and system blur. It also changes the minimization over blocks to consider only block pairs that are at least 2 positions apart. When the correct disparity is at a half-pixel boundary in the image, the response will be evenly split between blocks at two neighboring positions, and we don't want to penalize this.

C. Good Patterns under Noise

The optimization developed in the previous section for finding good patterns carry over directly here, using the new match score S^+. We first generate a random pattern P, transform it to an image $I^{P(\sigma)}$, then compute the SAD scores $S^+(P(\sigma))$ by minimizing over all phases. Because of the blur step, there is a unique minimum-score pair; we translate their indices to the respective n_p blocks in the pattern, and interchange two random dissimilar pixels in that pair. The change is accepted if it passes the annealing criterion, and the cycle repeats with decreasing temperature. Because the calculation of I^P and the scoring function are more complicated, we limited the annealing to 5000 steps and 10 restarts.

Figure 7 shows results for phase noise only (no blur) using $n = 11$, for all values of n_p from 3 to 11. The results are quite interesting. First, the best minimum SAD score, 40, is substantially less than the ideal pattern value of 52; this is the effect of phase noise. The best value also occurs at $\alpha = 11/8$. There are two competing phenomena that are balanced: phase noise penalizes small α because of aliasing in the transfer function; and large α have fewer pattern elements in an $n_p \times n_p$ block, limiting the Hamming distance between blocks.

For comparison, we also derive ideal patterns – i.e., using $S(P)$ on the pattern and not taking the image transfer function into account – using simulated annealing at the same resolutions, and then measure their scores under $S^+P(0)$. The ideal patterns can be optimized for one particular phase,

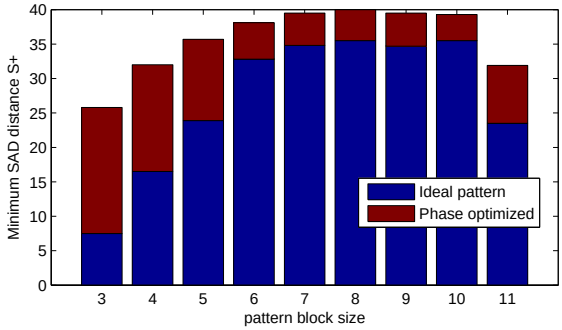

Fig. 7. Minimum SAD scores for $n = 11$ under different pattern sizes from 3 to 11. Only phase noise is considered ($\sigma = 0$). The blue bar indicates the score for idealized patterns, the red indicates the additional score for phase-optimized patterns.

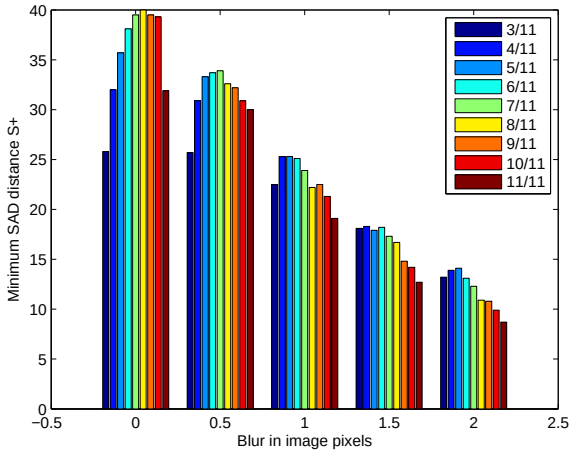

Fig. 8. Minimum SAD scores for $n = 11$ under different pattern sizes from 3 to 11, with blur from 0.0 to 2.0 pixels. Minimum SAD score gets lower for higher blur, and the best pattern shifts to bigger pixels (coarser blocks). The legend indicates the block size of the pattern and the image.

but there is no guarantee that they will have good scores in other phases. Overall, the scores are lower, emphasizing the importance of creating patterns that explicitly account for phase noise.

When blur is added, the scores go down, and the best α shifts higher (the pattern becomes coarser). Both these phenomena are expected, since the pattern becomes more diffuse and loses fine structure. Figure 8 summarizes the scores for image block size 11, and blur from 0.0 to 2.0 image pixels.

D. Experiments

We tested the optimized patterns against both ideal patterns and completely random patterns, at various image block sizes. A planar target was placed at an appropriate distance for each of the image block sizes to keep the pattern blocks in correspondence; the target contained the different patterns under test (see Figure 9). The printed planar target is much easier to change for experiments (and much cheaper) than

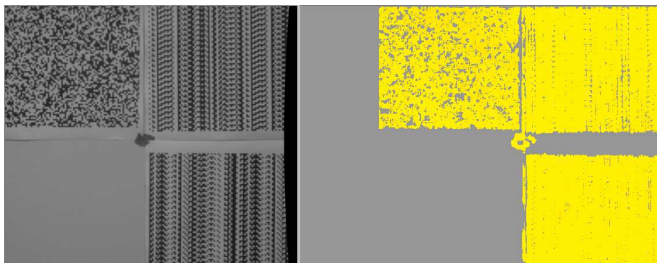

Fig. 9. Disparity images from three different patterns. Left image is the planar target with 3 patterns, as viewed by the left stereo camera; right is the computed stereo disparity. In the target, the upper left subimage is a random pattern; upper right is an ideal repeating pattern with $n_p = 3$; lower right is an optimized pattern with $n_p = 3$. Lower left corner has been left blank.

the etched pattern in the projector (Section IV). The target was presented fronto-planar to the cameras, which is a good approximation of how the projected texture will appear – note that a projected pattern always appears the same to a camera if it is co-located with the projection, and the focal lengths match. To a first approximation, these conditions are satisfied by our system.

For these tests, we used $n = 5, 7, 9, 11$ for the size of the image blocks, and $n_p = 3$ for the patterns, with a search range of $N = 128$; all pattern pixel sizes were the same. We also filtered based on the uniqueness of the response, eliminating any disparity that was not at least 50% better than the next best match. Figure 9 shows typical results from the stereo, at $n = 7$. The exposure on the cameras was turned down, to soften the pattern contrast and make matching more difficult.

Note first that it is difficult to distinguish the ideal pattern from the optimized pattern (right images of the first set). Still, the upper right ideal pattern has a smaller minimum SAD distance, which causes dropout striping in the yellow disparity image. By contrast, the optimized pattern is filled in. The completely random pattern fares most poorly, as expected, with numerous holes scattered throughout.

More qualitatively, we compared the dropout rate at each of the block sizes, for the three different pattern types: fully random, ideal pattern, and optimized pattern[1]. For the optimized pattern, we estimated the image blur to be about 0.5 pixels, based on images of sharp vertical lines. We computed the dropout rate as the percent ratio of the number of pixels not passing the next-best-match cutoff, to the total number of pixels in the image of the pattern. The results are summarized in the table below.

First, note that the larger SAD matching blocks do much better at removing dropouts. At the lowest block size of $n = 5$, all the patterns have a large percentage of dropouts, with the random pattern actually doing better than the ideal one. In all cases, the optimized pattern does best, with no significant dropouts until $n = 5$. The random and ideal patterns both have significant dropouts at $n = 7, 9$.

[1]The De Bruijn patterns do not exist for $n_p = 3$, and in any case they are subsumed by the ideal patterns.

TABLE I

DROPOUT PERCENTAGE FOR PATTERNS WITH $n_p = 3$

n	random	ideal	optimized
5	28.6	33.3	10.9
7	14.2	3.0	0.6
9	4.1	1.3	0.2
11	0.7	0.4	0.01

IV. COMPACT PROJECTOR

We have developed a compact, fixed-pattern projector based on the results of the previous sections. Figure 10 shows the optical design of the manufactured projector. A high-powered, large-format LED is collimated onto a metal-covered glass disk etched with the pattern (called a "gobo"). The pattern is projected using a standard C-format camera lens, with a focal length similar to that of the stereo cameras.

A. Power and Synchronization

The main issue with the projector is to send enough optical energy out to be seen easily by the camera under ambient light, while maintaining eye safety. A high-power LED with optical output of 5 watts and large surface area was used; for comparison, a typical bright display projector has about 3 watts of output power. Good performance was achieved by:

- Using a red LED – red is much less harmful to the eye than any other visible color. The device is eye safe as defined by the limits of IEC 62471 for LED emissions [18], as tested by an independent lab. It would also be safe in the infrared range.
- Pulsing. The LED is pulsed in synchrony with the exposure of the camera. By limiting the exposure time to several milliseconds, the average power of the projector is reduced, while maintaining high power with respect to ambient light during exposure.

A bandwidth filter at the LED frequency could improve performance by rejecting ambient light, but was not included because we want to be able to capture untextured images with the same stereo pair. The system can be run in a mode where every other frame is captured with the texture, giving registered stereo and normal images separated by 1/30 of a second. It is also possible to run the system at 60 Hz, and we are upgrading the camera electronics to do this.

B. Performance

The system functions well in even bright daylight conditions indoors, out to a distance of 2 to 3 meters. Without bandwidth filtering, it is not strong enough to overcome direct sunlight (1300 W/m^2) except very near the projector, but it can still supplement whatever natural texture exists, unlike structured light devices.

We tested the device with a 50 degree field of view, using both white and 5% reflectance black planar targets at different distances. The SAD block-matching stereo algorithm of [1] was used, with an image block size of 11×11. The error is taken to be the standard deviation from the best-fit plane.

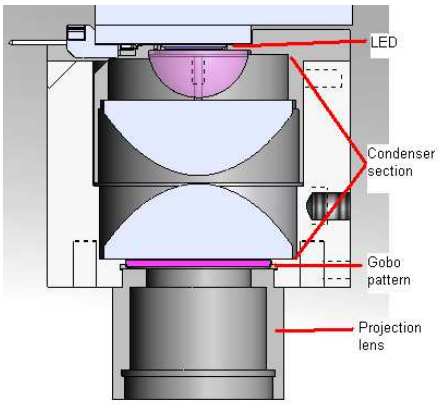

Fig. 10. Optical design of the compact projector, showing major components: high-power LED, condenser, gobo pattern disk, and projection lens. Overall length is ~10cm.

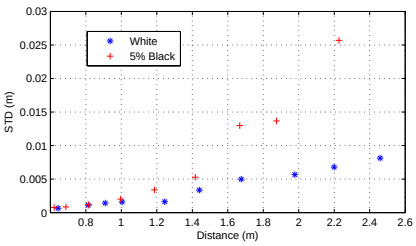

Fig. 11. STD error of a planar target. For a white target, the error stays below 2mm until after 1.2m, then goes up to about 1cm at 2.5m. For a very dark target, it is also low close up, then becomes larger at distance, when the pattern is difficult to see.

From Figure 11, the system shows very low error, even out to 2.5 meters. For the white target, the error stays below 1cm throughout this range. Some of the error at the larger distances comes from calibration, as the reconstructed plane will not be perfectly flat. Up to over 1 meter, the error is about 2mm, which is good enough to reconstruct fine objects.

Note that we are using a standard stereo block-matching algorithm for these experiments, without modification. A concern is that, because of the blockiness of the pattern, subpixel resolution in the disparity calculation might not be possible, because the block correlation does not have a smooth transition across the image. However, because of significant blur in both camera and projector, subpixel resolution works well, as can be seen in Figure 1. The 3D points on the cup, for example, would not fit a curve so well without sub-pixel resolution.

Even with a very dark (5% reflectance) target, the system gives good results up to 2m, with some degradation in error at the larger distance. Another issue with very dark targets is that dropouts start to occur. We set a cutoff of 18% for the minimum distance between the highest and second-highest SAD response. For the white target, every pixel made this cutoff. For the dark target, dropouts start to occur at 1.2m, and increase linearly to 2.5m, when there are no pixels that make the cutoff.

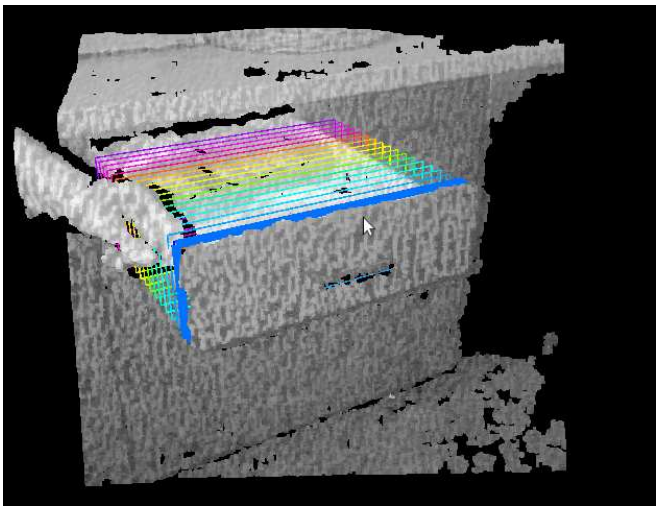

Fig. 12. Application to estimating the articulation model of a drawer. This view shows the point cloud extracted from the stereo system, with the projected texture, and the track of the door extracted from previous point clouds as it moved. Image courtesy of Juergen Sturm.

We have used the projected texture system in several applications, most especially the recognition of tabletop objects using both 2D and 3D features. An interesting application is learning the articulation models of common objects (Figure 12, from [19]).

One interesting aspect of the system is that it can work with fast block-matching algorithms, without the performance loss associated with more sophisticated methods. High-performance block-matching algorithms have been developed as part of the ROS open-source robotic software system [20]. Running on a 3GHz Intel i7 processor, with a single core, 640x480 images with 64 disparities and subpixel disparity resolution achieve a 30 Hz frame rate.

C. Algorithm and Datasets

The stereo algorithms and the pattern generators are available under open source BSD license as part of the ROS application packages (Robot Operating System - `http://www.ros.org/Papers/ICRA2010_Konolige`).

V. Conclusions and Future Work

We have explored the concept of good projection textures for overcoming stereo dropouts, both at the conceptual level of finding patterns that exhibit good dissimilarity between all blocks in a search range, and at the system level of dealing with image resolution, phase and blur. The fixed-pattern projection and stereo device that we constructed is a practical sensor system for robotics applications, filling a niche for short range dense 3D sensing that can deal with moving objects. It is being incorporated into the PR2 mobile robots built at Willow Garage (`http://www.willowgarage.com`), and will be the primary sensor for tabletop manipulation.

One problem with the sensor is that the strong red projected texture is annoying when viewed directly. The aversion response helps to make the device safe, but it is not useful

for directly imaging faces because of this drawback. An IR version, with lower power and a more sensitive imager, is a possible solution.

REFERENCES

[1] K. Konolige, "Small vision systems: hardware and implementation," in *Eighth International Symposium on Robotics Research*, 1997, pp. 111–116.

[2] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Jour. Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, 2002.

[3] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. Conf. on Computer Vision and Pattern recognition (CVPR 2005)*, vol. 2, 2005, pp. 807–814.

[4] J. Lim, "Optimized projection pattern supplementing stereo systems," in *in ICRA*, 2009.

[5] T. Molinier, D. Fofi, J. Salvi, P. Gorria, and F. Meriaudeau, "Projector view synthesis and virtual texturing," in *2nd International Topical Meeting on Optical Sensing and Artificial Vision*, 2008.

[6] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash ladar devices," in *International Conference of Sensing and Technology*, November 2005.

[7] B. Curless and M. Levoy, "Better optical triangulation through space-time analysis," in *ICCV*, 1995.

[8] J. Salvi, J. Pages, and J. Batlle, "Pattern docification strategies in structured light systems," *Pattern Recognition*, vol. 37, no. 4, 2004.

[9] H. K. Nishihara, "Prism: A practical real-time imaging stereo matcher," Cambridge, MA, USA, Tech. Rep., 1984.

[10] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework dor depth from triangulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, February 2005.

[11] L. Zhang, B. Curless, and S. Seitz, "Spacetime stereo: shape recovery for dynamic scenes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[12] O. Williams, M. Isard, and J. MacCormick, "Estimating disparity and occlusions in stereo video sequences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[13] F. Tombari and K. Konolige, "A practical stereo system based on regularization and texture projection," in *Proc. Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2009)*, Milan, Italy, July 2009, pp. 5–12.

[14] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR 2007)*, vol. 1, 2007, pp. 1–8.

[15] A. Vardy, "Algorithmic complexity in coding theory and the minimum distance problem," in *ACM Symposium on Theory of Computing*, 1997.

[16] J. H. Conway and N. J. A. Sloane, "Lexicographic codes: error-correcting codes from game theory," *IEEE Transactions on Information Theory*, vol. 32, pp. 337–348, 1986.

[17] L. Ingber, "Simulated annealing: Practice versus theory," *Math. Comput. Modelling*, vol. 18, pp. 29–57, 1993.

[18] IEC 62471, "Photobiological safety of lamps and lamp systems," International Commision on Illumination, Tech. Rep., 2006.

[19] J. Sturm, K. Konolige, C. Stachniss, and W. Burgard, "Vision-based detection of cabinet doors and drawers for learning articulation models in household environments," in *ICRA*, 2010.

[20] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *International Conference on Robotics and Automation*, ser. Open-Source Software workshop, 2009.