# A MRF formulation for coded structured light

Jean-Philippe Tardif Sébastien Roy
*Département d'informatique et recherche opérationnelle*
*Université de Montréal, Canada*
{tardifj, roys}@iro.umontreal.ca

## Abstract

*Multimedia projectors and cameras make possible the use of structured light to solve problems such as 3D reconstruction, disparity map computation and camera or projector calibration. Each projector displays patterns over a scene viewed by a camera, thereby allowing automatic computation of camera-projector pixel correspondences. This paper introduces a new algorithm to establish this correspondence in difficult cases of image acquisition. A probabilistic model formulated as a Markov Random Field uses the stripe images to find the most likely correspondences in the presence of noise. Our model is specially tailored to handle the unfavorable projector-camera pixel ratios that occur in multiple-projector single-camera setups. For the case where more than one camera is used, we propose a robust approach to establish correspondences between the cameras and compute an accurate disparity map. To conduct experiments, a ground truth was first reconstructed from a high quality acquisition. Various degradations were applied to the pattern images which were then solved using our method. The results were compared to the ground truth for error analysis and showed very good performances, even near depth discontinuities.*

## 1. Introduction

Coded structured light is an active computer vision method employing multimedia projectors and cameras to solve problems such as camera or projector calibration [9], 3D reconstruction [5, 11, 10, 4] and disparity maps computation [14]. It encodes the position of each projector pixel with one or many patterns projected over some surface imaged by a camera. Those images are combined to recover the code and thus pixel correspondences between the camera and the projector. Many kinds of structured-light systems have been described, and a good overview is presented in [1, 12]. Errors in the correspondences occur from noise in the images and, in some cases, ambiguities in the patterns themselves. Indeed, solution usually make a compromise between the quality of the correspondences and the acquisition time (related to the number of patterns) [8, 3, 2]. The use of many cameras can also increase the pattern decoding robustness [7].

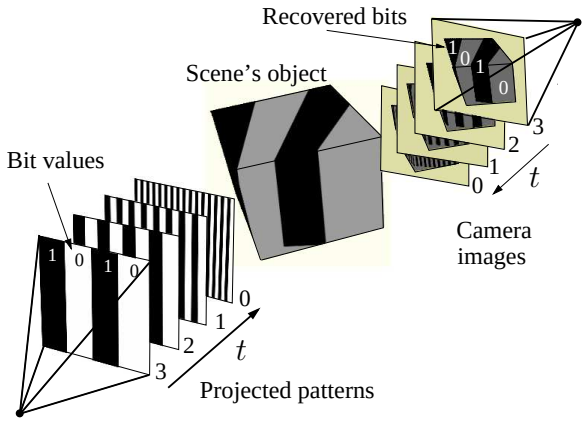



**Figure 1.** Projected patterns for bits $3, 2, 1, 0$ in $x$ direction for $T = 4$ . Inverse patterns and those used for the $y$ direction are not shown.

Most industrial scanners require a controlled environment to work properly. In some cases, the nature of the scene imposes a hostile environment which makes the scanning much more difficult. Also, today's systems usually assume that the camera-projector pixel ratio (the number of pixels of the projector seen by only one pixel of the camera) is around one. But this is not always true. For instance, structured light based multiple-projector systems, in which the pixel ratio decreases as the number of projectors increases, were demonstrated [15, 19]. Also, as these systems become less costly and more widely available, support for poor camera quality and bad environment is needed. A more robust structured light approach should be developed for those situations.

This paper introduces a new algorithm to establish correspondence in difficult cases of image acquisition. A probabilistic model formulated as a Markov Random Field uses

stripe images to estimate the most likely correspondences in the presence of noise. Our model is specially well-suited to handle the unfavorable projector-camera pixel ratios that occur in multiple-projector single-camera setups. Other degradations can be caused by low contrast due to strong ambient light, high image noise from low quality cameras, and also weak projector lamps or large scanning distances. The probabilistic nature of these degradations justifies the use of such a model.

Generally, the evaluation of the correspondence accuracy requires the use of calibrated camera, projector and reference object. To avoid this task, we used multiple uncalibrated cameras to compute disparity maps. These maps can be used directly to evaluate performance and could be used to compare to passive stereo methods [13]. We propose a robust approach to compute the disparity from many camera-projector correspondence maps. In our experiments, a ground truth disparity map was first reconstructed from a high quality acquisition. Various degradations were applied to the pattern images which were then processed using our method. The results were compared to the ground truth for error analysis.

The article proceeds as follows: first we introduce the structured-light approach we chose; next we introduce the model for code correction; then we show how to build multiple-camera disparity maps; finally, some results are shown and discussed.

## 2. System overview

To illustrate how a Markovian model can be used to achieve code correction, we present a complete structured light system. We deliberately chose a simple system in order to demonstrate more clearly the effectiveness of our reconstruction model, but it should also be directly applicable to other systems. In our case, the projected patterns are horizontal and vertical black and white stripes to allow an arbitrary projector/camera configuration. Inverse patterns are also used to increase the robustness of the decoding. This decoding defines the initial measurements for the camera-projector correspondences. In our model, they are the most likely values. However, errors can occur, so we show how to compute a confidence value associated to every bit. It is used to define a Markov Random Field for which the most likely configuration can be determined using the Iterated Conditional Mode (ICM) algorithm.

## 3. Complete structured light system

The complete encoding of a pixel position is done using multiple patterns. To simplify notation in this paper, we assume the projector image is square and its width is representable with $T$ bits. A position $(x, y)$ is encoded independently in each dimension so we illustrate our method using a single coordinate, say the $x$ one.

In order to define our structured light patterns, we define the binary encoding of a pixel $\alpha$ of the projector as:

$$P(\alpha) = \overline{\alpha_s} \oplus \alpha_s$$

where $\alpha_s$ is the binary encoding of $\alpha$, $\overline{\alpha_s}$ the binary complement of $\alpha_s$ and $\oplus$ the concatenation operator. We also define $P^t(\alpha)$ as the $t^{\text{th}}$ bit of $P(\alpha)$. The color of $\alpha$ in the pattern $t$ is white if $P^t(\alpha)$ is 1 and black if it is 0. This encoding corresponds the projection of patterns as shown in figure 1 and the binary complement formalizes the use of inverse patterns to increase robustness in the decoding process (section 3.1).

Subtraction of an image from its inverse and thresholding is the basic way to discover the value of a bit. Unfortunately, uncertainties occur when the difference becomes very small. This typically happens when the contrast in the images is low or when a border between stripes is projected onto a single pixel. This basic encoding has the drawback of keeping many stripe borders aligned which can make many bits uncertain in a single code. To correct this, we rely on Gray encoding [17] which minimizes the encoding's bitwise difference between spatial neighbors in the projector image. For example, if we consider two projector pixels having $x$ coordinates 127 and 128. The use of Gray encoding changes their binary representations from 0111111 and 1000000 to 0100000 and 1100000, thus reducing the number of stripe borders located between these pixels from 8 to 1. The encoding becomes:

$$P'(\alpha) = \overline{G(\alpha_s)} \oplus G(\alpha_s)$$

where $G$ is defined as:

$$G(\alpha_s) = \alpha_s \ \mathbf{xor}\ (\alpha_s >> 1)$$

and $>>$ is the *right bit shift*.

### 3.1. Pattern decoding

This step builds the first estimate of the correspondence of each camera pixel to a projector pixel and computes a confidence measure based on the observed pixelwise contrast.

For each camera pixel $\beta$ and each pattern $t$, an intensity $I^t(\beta)$ is measured. We define the image contrast as:

$$\delta(\beta) = \max_t \left(I^t(\beta)\right) - \min_t \left(I^t(\beta)\right).$$

The Gray code correspondence can then be recovered with a simple image difference:

$$C(\beta) = \bigoplus_{t=T-1}^{0} \text{Bin}\left(I^t(\beta) - I^{t+T}(\beta)\right)$$

where

$$\text{Bin}(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{otherwise} \end{cases}$$

(the decreasing index of the concatenation simply reflects the fact that the most significant bits appear first). Note that no thresholding is done at this step. However, a threshold is used to build a confidence mask for the measurement:

$$H_\beta = \left\{ \text{Conf}\left(I^t(\beta) - I^{t+T}(\beta), \beta\right) \right\}_{t=T-1}^{0}$$

where

$$\text{Conf}(d, \beta) = \begin{cases} 1 & \text{if } |d| > K_c\delta(\beta) \text{ and } \delta(\beta) > K_r \\ \frac{1}{2} & \text{otherwise.} \end{cases} \quad (1)$$

$H_\beta^t$ is the probability we determine for the $t^{\text{th}}$ bit of the Gray code of $\beta$'s projector $x$-coordinate to be equal to $C^t(\beta)$, the $t^{\text{th}}$ bit of $C(\beta)$. The constants $K_c = \frac{1}{2}$ and $K_r = 15$ are conservative enough to insure that the contrast is sufficiently high. These values rarely need to be changed in practice. A confidence value can be seen as a probability that the measurement can be trusted. Indeed, a value of 1 means that the bit was *unambiguously* recovered and a value of $\frac{1}{2}$ means total uncertainty. Finally, the number of 1's in $H_\beta$ indicates the overall quality of the code recovered for pixel $\beta$. Accordingly, the Boolean value:

$$C_v(\beta) = \begin{cases} 1 \text{ if the \# of 1's in } H_\beta \geq T - \max(\log_2 \rho, 0) - 1 \\ 0 \text{ otherwise} \end{cases}$$

where $\rho$ is an estimate of the smallest pixel ratio between the projector and its image in the camera and "$-1$" is plainly a margin of error. This function determines if $\beta$ was sufficiently illuminated by the projector.

The value of $C(\beta)$ is the most likely value of the projector correspondence of $\beta$. In ideal situations, this value is close to being exact, but most of the time many errors occur. The next section explains how the codes can be corrected.

### 3.2. A Markovian model for code correction

The low confidence in certain bits of a correspondence can result from two factors. The first is the projection of a border onto a pixel. Even though this occurs more frequently for low order bits, it can actually happen at any level. Fortunately, even if high order bits weren't recovered for a pixel, there is a good chance they were for its neighbors. When all of them have the same value for some bit, chances are this is the right value for the current pixel too.

The second factor is related to the pixel ratio between the camera and the projector. For similar resolutions, if the area covered by the projector is smaller than that covered by the camera, low order bits cannot be recovered. In this case, the neighbors are of no help. However, a hypothesis can be made, that the object surface is locally smooth, and thus the codes as well. In most cases, with the use of Gray codes, only a small number of bits of a correspondence will be uncertain. The scheme we present tries to find the code that best satisfies these assumptions. The Markovian approach is known to be well adapted to solve this type of problems.

We represent the camera image by a graph $G = (B, N)$. A site $\beta \in B$ is a pixel with a value of $C_v(\beta)$ equal to 1. Each site's neighborhood $\mathcal{N}_\beta$ is the usual 8-neighborhood (possibly consisting of less than 8 elements). The labels are the $2^T$ possible values of the coordinate in the projector image. When a value is associated to each site, the Markovian field $M$ is in a configuration $m$ whose probability is function of the measured code $C$. Given its computed value $c$, we are looking for the most likely value of $M$. In the following, we use $C_\beta^t$ for $C^t(\beta)$ to increase the equations compactness. We have:

$$\begin{aligned} &P(M = m \mid C = c) \\ &\quad \propto P(C = c \mid M = m) \cdot P(M = m) \\ &\quad \propto \left( \prod_{\beta \in B} P(C_\beta = c_\beta \mid M_\beta = m_\beta) \right) \cdot P(M = m) \end{aligned}$$

(if we suppose pixel- and bitwise independence)

$$\propto \left( \prod_{\beta \in B} \prod_{t=0}^{T-1} P(C_\beta^t = c_\beta^t \mid M_\beta^t = m_\beta^t) \right) \cdot P(M = m)$$

$$\propto \left( \prod_{\beta \in B} \prod_{t=0}^{T-1} P(C_\beta^t = c_\beta^t \mid M_\beta^t = m_\beta^t) \right) \prod_{\substack{\beta_1 \in B \\ \beta_2 \in \mathcal{N}_{\beta_1}}} e^{-\xi V(m_{\beta_1}, m_{\beta_2})}$$

where $V$ is the smoothing cost function and $\xi$ a smoothing factor. Taking minus the log, this is equivalent to minimizing directly the cost function:

$$-\sum_{\beta \in B} \sum_{t=0}^{T-1} \log P(C_\beta^t = c_\beta^t \mid M_\beta^t = m_\beta^t) + \xi \sum_{\substack{\beta_1 \in B \\ \beta_2 \in \mathcal{N}_{\beta_1}}} V(m_{\beta_1}, m_{\beta_2}) \quad (2)$$

w.r.t. $m$. The value of $P(C_\beta^t = c_\beta^t \mid M_\beta^t = m_\beta^t)$ is modeled using the confidence that was recovered previously. This can be expressed as:

$$P(C_\beta^t = c_\beta^t \mid M_\beta^t = m_\beta^t) \propto \begin{cases} H_\beta^t & \text{if } c_\beta^t = m_\beta^t \\ 1 - H_\beta^t & \text{otherwise.} \end{cases} \quad (3)$$

The corresponding likelihood of getting a value of 1 as a function of the intensity difference is shown in figure 2a.

Defining $V(\beta_1, \beta_2) = \left| G^{-1}(\beta_1) - G^{-1}(\beta_2) \right|$, where $G^{-1}$ converts a Gray code to its real value, is a logical

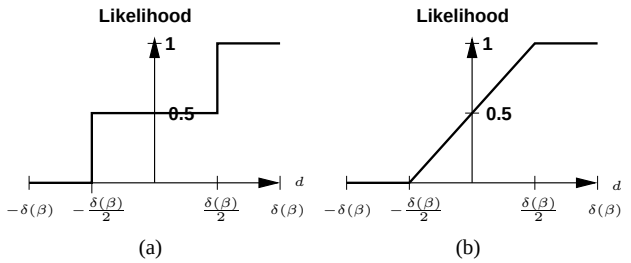



**Figure 2.** Likelihood that a recovered bit is 1 as a function of the intensity difference $d$ of the two corresponding patterns, based on **a)** eq. 1 and **b)** eq. 4.

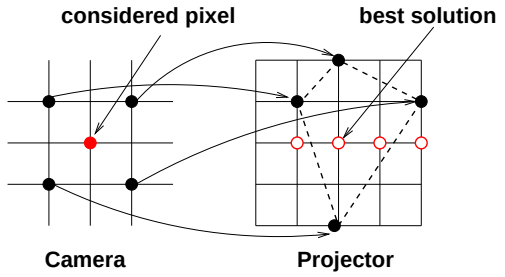



**Figure 3.** For a pixel whose $x$ projector correspondence has uncertain low-order bits 0 and 1, four solutions are possible (empty circles). According to our model, the most likely code minimizes the average distance to its neighbors (only the 4-neighborhood is shown for clarity).

choice. The effect of the smoothing term is that it favors codes that are in between those of the neighbors (figure 3). Unfortunately, it is not clear what PDF corresponds to this relation. .

Another definition clarifies the effect of the confidence and matching cost functions. For a given pixel $\beta$, a code $\nu$ is said to be *compatible* if it is identical to $\mu = C(\beta)$ for the bits *unambiguously* recovered. This can be expressed as:

$$\text{Comp}(\beta, \mu, \nu) = \begin{cases} 1 & \text{if } \forall\, t \text{ such that } H^t_\beta = 1 : \mu^t = \nu^t \\ 0 & \text{otherwise.} \end{cases}$$

From eq. 1 and 3, we see that all compatible codes have equal matching costs and that all the others have probability 0. This means that the correct code must be compatible. In practice, this function can be used for the optimization to reject labels without computing the cost function. Also, note that when using eq. 1 in the MRF, multiplying $\xi$ by any positive constant does not alter the amount of smoothing.

More sophisticated approaches can be used. Indeed, even when a certain bit of a code has low confidence, the value found by image difference is still more likely than its complement. A simple model for this is illustrated in figure 2b. The closer to zero this difference is, the more ambiguous the value becomes. The confidence function related to figure 2b is:

$$\text{Conf}(d, \beta) = \begin{cases} 1 & \text{if } \delta(\beta) > K_r \text{ and } |d| > K_c\delta(\beta) \\ \frac{1}{2} & \text{if } \delta(\beta) \leq K_r \\ \frac{|d| + K_c\delta(\beta)}{2K_c\delta(\beta)} & \text{otherwise} \end{cases} \quad (4)$$

where $K_c$ and $K_r$ are the same as in equation 1. Finally, another confidence measure could be used without resorting to thresholding; all labels would have non zero probabilities. Unfortunately, minimizing the corresponding function is far too complex in practice.

Cost functions such as eq. 2 are generally not too difficult to minimize globally and efficiently. However, in our experiments, the image resolution and the number of labels are overwhelming. Resolution by ICM yields convincing results within a reasonable computational time (see section 7 for details), even though only a local minimum is found.

## 4. Projector-to-camera correspondences

The projector-to-camera correspondence map is used for image construction in a multiple projector system [15] and also for disparity map construction (section 5). It is built by inverting the camera-to-projector mapping obtained by structured light. This inverse function is not easily determined. In our experiments, we were using a camera and a projector with similar resolutions. Since the surface illuminated by the projector was contained inside the camera image, only a sampling of all the codes could be achieved. In the projector domain, this means that not all projector pixels have a corresponding camera pixel, which creates holes in the inverse map. One way to fill these holes is to use interpolation. One must find a scheme with a solid geometrical interpretation that performs well in terms of accuracy and execution time.

A first scheme uses homography-based interpolation. This assumes that camera and projector models are linear for small areas of the image, and that the object surface can be approximated locally by a small planar patch. The camera image is divided into 4-pixel patches reprojected in the projector image onto the correspondence points. Then, if some projector pixel is located inside this patch, its value is calculated from a homography defined with the four corners of the patch.

Another scheme presented in [15] uses triangular patches with bilinear interpolation. The geometric interpretation is less intuitive, but it has been used successfully in applications where small inaccuracy could be tolerated. This scheme is specially useful when a fast implementation over a GPU is necessary.

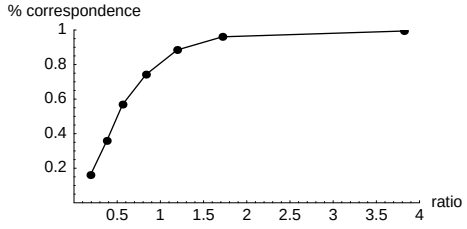



**Figure 4.** Percentage of exact correspondences between two cameras w.r.t. pixel ratio between the projector and its image in the cameras.

The result of the inversion is that for each pixel $\alpha$ of the projector, a corresponding camera pixel $C^{-1}(\alpha)$ is defined. When a reconstruction is very well recovered, the relation $C(\beta) = C^{-1}(C(\beta))$ should be valid for all pixels $\beta$ illuminated by the projector. This does not occur in practice, because of errors in the reconstruction and because the pixel ratio can be smaller than one.

## 5. Disparity map construction

Without calibration of the cameras and projectors, it is impossible to achieve a full 3D reconstruction of the scene. However, it is possible to build simple disparity maps [14].

The basic approach for disparity map construction is to use some camera as the main view. A pixel $\beta$ of this camera and its correspondence $\beta'$ in another camera $j$ both have the same $x$ and $y$ projector codes. Their image distance is the disparity. Estimating camera-camera correspondences amounts to locating common projector codes. For known epipolar geometry, this is done with a linear search, otherwise, a 2D search is needed. This process can be very long for large images. Also, when the pixel ratio is smalle than one or when a lot of errors occur in the correspondences, some codes are not present in the other images. Figure 4 illustrates the percentage of pixels in an image that have an exact correspondence pixel in another image, as a function of the camera-projector pixel ratio. As expected, almost all codes are available for pixel ratios larger than 2, but become scarce for low ratios. In this case, direct estimation of disparity is unusable.

A more robust approach is to use the inverse correspondence map presented in section 4. Let us define the correspondence of projector pixel $\alpha$ in camera $j$ as $C_j^{-1}(\alpha)$. Therefore, the pixel in camera $j$ corresponding to pixel $\beta$ of camera 0 is simply $C_j^{-1}(C_0(\beta))$. Because the inverse mapping function is interpolated, it allows code interpolation. For perfectly reconstructed scenes, $C_0(\beta) = C_j(C_j^{-1}(C_0(\beta)))$ should be true.

For many cameras located on a single baseline without any rotation, we compute the disparity of a pixel $\beta$ of camera 0 w.r.t. camera $j$ as:

$$D_{0,j}(\beta) = \frac{1}{n}\,\text{dist}(0,j)\sum_{i=1}^{n}\frac{\left\|\beta - C_i^{-1}(C_0(\beta))\right\|}{\text{dist}(0,i)}$$

where $\text{dist}(0,j)$ is the distance between the two cameras' optical centers. A correspondence for one camera is not used when the error between correspondence codes is too large. In our experiments, we rejected a correspondence when the distance was above 2 pixels. Rejection of correspondences for all cameras results in unknown disparity for this pixel.

## 6. Validation

Validation of a structured light system is difficult and sometimes involves precise setup and calibration. A full 3D reconstruction of a perfectly known scene can be used for error analysis. In the context of poor image acquisition, we propose a different approach to test the quality of the correspondences. A scene containing two parallel and overlapping planes was carefully reconstructed in a controlled environment (constant ambient lighting) with a powerful XGA $1024\times768$ projector of 2000 lumens and a low-noise Basler A201bc cctv $1008 \times 1018$ camera. Each plane featured a checkerboard texture of varying colors so the contrast in the images is not uniform. A disparity map was computed using one camera moved to six locations on a single baseline. Its accuracy was high enough to be considered as our ground truth. Pictures of the two planes, the disparity map and one sample slice are shown in figure 7.

In previous experiments, we had observe that our algorithm performs much better than classical approaches in the presence of high noise. We also wanted to show that significant improvements could be achieved in better, more realistic conditions. In order to do this, we measured the noise induced in the pattern images when compression is turned on, as it commonly is on low quality cameras (cf. figure 5). Than in our tests, we corrupted the images with Gaussian noise of mean 0 and standard deviation 2, a smaller value than the previous measurement. In addition, we gradually reduced the images contrast by $K$ percent (cf. figure 6).

Four algorithms were tested with gradually decreasing contrast to increase correspondence errors. We recovered the codes with each of them and built the disparity maps. Codes and maps were compared to the ground truth. The first algorithm (labeled $P$) consisted solely in the recovery of the codes without any further correction. In the second, labeled $Q$, we used a simple low-pass filtering of the codes. A $5\times5$ filter was empirically determined to be a good compromise between smoothing and discontinuity preservation. Finally for the third and fourth, respectively labeled $M_1$ and $M_2$, we tested our proposed Markovian approach with the

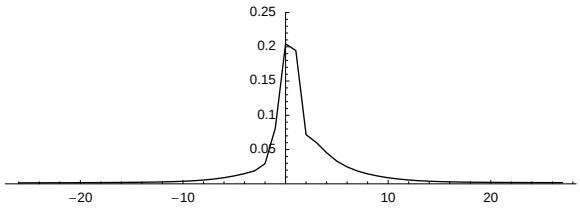
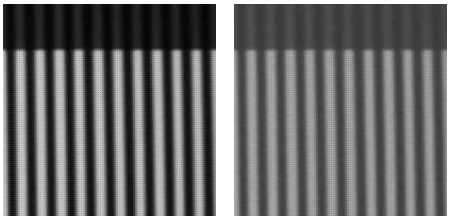



**Figure 5.** Histogram of the noise in the pattern images compressed with MJPEG (compression ratio around 10:1). The variance of the noise is higher than that of the noise we added to test our algorithms.



(a) (b)

**Figure 6. a)** Zoom on a section of a non-corrupted pattern. **b)** Same image with a 41% image contrast reduction and added noise.

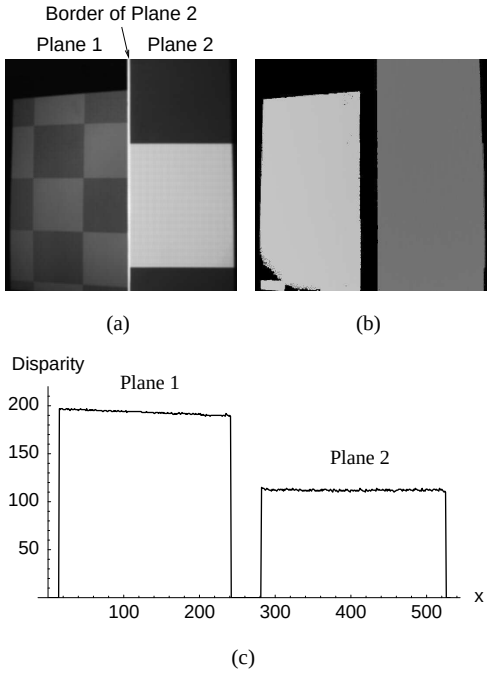



**Figure 7. a)** Image of the two planes. **b)** Disparity map reconstructed for the left camera. **c)** One slice of the disparity map.

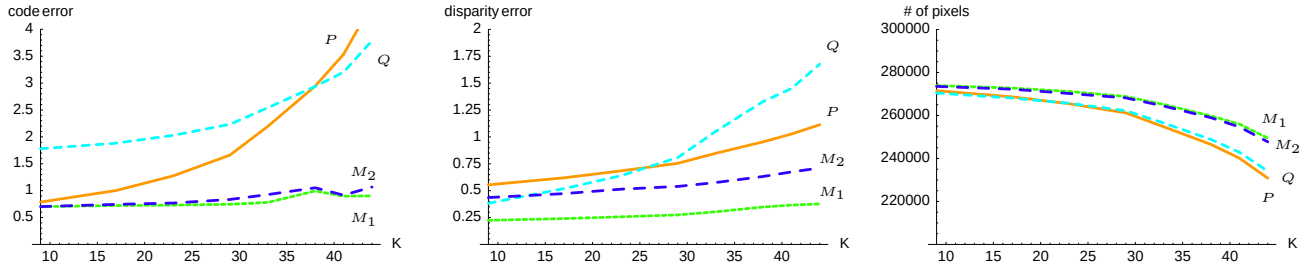
matching cost functions of eq. 1 and eq. 4. We used the ICM algorithm to perform the optimization, moving randomly from one pixel to another [6]. An operation consists of computing the cost function for every possible code for a given pixel and then select the best value. One iteration contains a number of operations equal to the number of pixels in the image. The configuration for which the cost function was minimum was kept and the process stopped after 7 consecutive iterations without improvement.

## 7. Results

Figure 8 gives the average error in pixels in the disparity maps and the recovered codes of each algorithm, as a function of the contrast reduction $K$. This error is computed as the euclidean distance between a recovered code and the ground truth, meaning that errors for high-order bits are worst than that for low-order bits. For the disparity error (figure 8b), the only pixels considered are those for which a disparity value was recovered. Figure 8c illustrates the number of pixels kept for different values of $K$. We observed that the Markovian approach is always superior to raw codes ($P$) and filtering ($Q$), particularly when the decoded patterns have a lot of errors. Moreover, the number of rejected pixels is always smaller. It also came as a surprise that the simpler model $M_1$ performs as well as $M_2$, and sometimes better. For the latter, finding a good solution is more difficult when the codes are highly corrupted, especially near discontinuities (figure 10). Moreover, each iteration for algorithm $M_1$ takes about 5 seconds, but takes more than twice as much for $M_2$, as the likelihood cannot be precomputed because of memory limitations. For our tests, convergence of ICM takes between $20$ and $40$ iterations, and similarly for $M_1$ and $M_2$. The parameter $\xi > 0$ has no influence on the solution when using $M_1$, but has a big impact for $M_2$. This is illustrated in figure 9, in which we also observe that $M_2$ never yields significantly better results as $M_1$. A value $\xi$ equal to $0$ is equivalent to no code correction. As $\xi$ gets larger, the solution for eq. 4 gradually converges to the one obtained with eq. 1. We observe that for a value above $0.02$, the error is close to stable, and above $0.05$, the solutions are exactly the same. In our tests, we used a value $\xi = 0.04$.

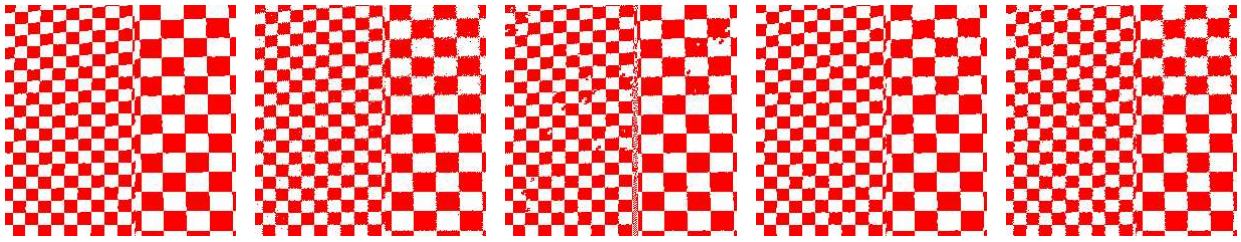
Figures 8 shows that even a small amount of error in the codes (8a) can result in large differences in the disparity maps (8b). These errors increase the number of rejected correspondences (8c), yielding holes in the maps, as illustrated in figures 12 and 11 .

The results of the filtering algorithm ($Q$) are surprising. Bad outliers occur frequently, and since they are not corrected in any way, they introduce very large disparity errors which appear as spikes in figure 12b. Consequently, filter-




(a) (b) (c)

**Figure 8.** Performance as a function of contrast reduction $K$. **a)** Mean error in the recovered codes. **b)** Mean error in pixels of the disparity maps. **c)** Number of pixels with a recovered disparity.



(a) Ground truth (b) Algo. $P$ (c) Algo. $Q$ (d) Algo. $M_1$ (e) Algo. $M_2$

**Figure 10.** Recovered codes for contrast $K = 41\%$ textured with a checkerboard image.

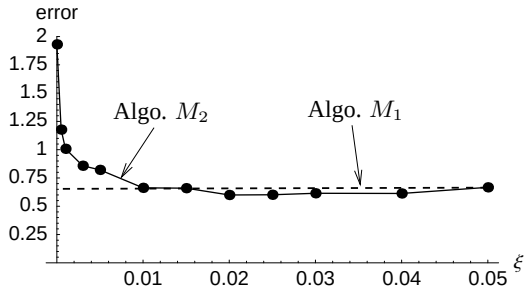



**Figure 9.** Mean error in the codes w.r.t. the smoothing parameter $\xi$. For $\xi = 0$, no correction is made to the codes. For $\xi \geq 0.05$, the choice of matching cost function makes virtually no difference.

ing is suitable only in the absence of error in high order bits. In fact, the filtering mechanism, unless combined with a robust pixel selection, always propagates the error of a pixel to its neighbors instead of correcting it.

## 8. Conclusion

This paper presented a Markovian approach to coded structured light reconstruction. We consider it is one step toward widespread use of structured light in uncontrolled environment with commonly available equipment. It performs more robustly than conventional methods and recovers accurate depth discontinuities. It was used successfully to calibrate a large multi-projector screens used in a public performance [16].

In the future, a more physically plausible model could be investigated, but the increased computational burden might prove unsurmountable.

## References

[1] J. Batllea , J. Salvia. Recent Progress in Structured Light in order to Solve the Correspondence Problem in Stereovision. ICRA 1997.

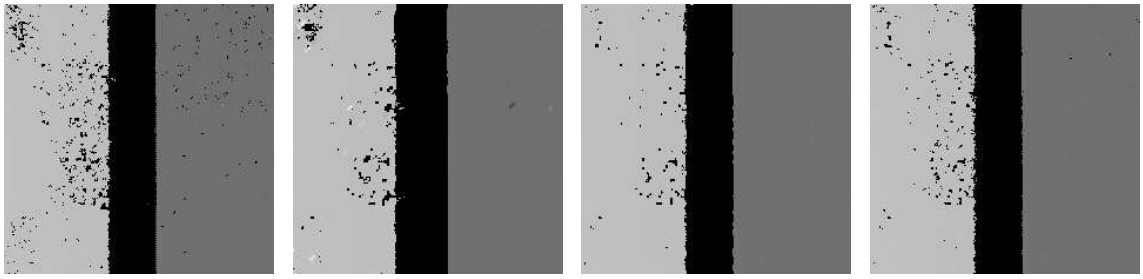
[2] Nelson L. Chang. Effi cient Dense Correspondences using Temporally Encoded Light Patterns. Procams 2003.

[3] E. Horn, N. Kiryati. Toward Optimal Structured Light Patterns. 3DIM 1997.

[4] P. Huang, S. Zhang. High-resolution, Real-time 3D Shape Acquisition. In IEEE Workshop on real-time 3D sensors and their uses, 2004.

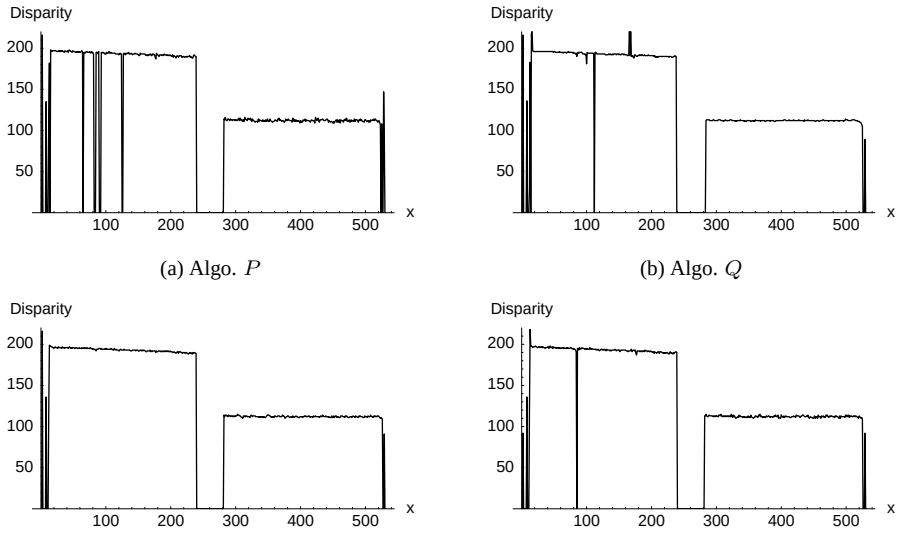
[5] T. P. Koninckx, I. Geys, T. Jaeggli , L. V. Gool. A Graph Cut based Adaptive Structured Light approach for real-time Range Acquisition. 3DPVT 2004.

[6] S.Z. Li. Markov Random Field Modeling in Computer Vision. Springer-Verlag 1995.



(a) Algo. $P$ (b) Algo. $Q$ (c) Algo. $M_1$ (d) Algo. $M_2$

**Figure 11.** Zoom on a section of the disparity maps computed with contrast $K = 30\%$. Black represents pixels with no recovered disparity.




(a) Algo. $P$ (b) Algo. $Q$

(c) Algo. $M_1$ (d) Algo. $M_2$

**Figure 12.** Horizontal slices of the disparity maps of figure 11. Pixels with unrecovered disparities are set to 0.


[7] Hans-Gerd Maas. Robust Automatic Surface Reconstruction with Structured Light. International Archives of Photogrammetry and Remote Sensing Vol. XXIX (1992).

[8] J. Pagès, J. Salvi, C. Matabosch. Implementation of a robust coded structured light technique for dynamic 3D measurements. ICIP 2003.

[9] R. Raskar , M. S. Brown , R. Yang, W. Chen , G. Welch , H. Towles , B. Seales , H. Fuchs. Multi-Projector Displays Using Camera-Based Registration. IEEE Visualization '1999.

[10] C. Rocchini , P. Cignoni , C. Montani , P. Pingi , R. Scopigno. A low cost 3D scanner based on structured light. Computer Graphics Forum (Eurographics 2001 Conf. Issue). Blackwell v 20,3.

[11] S, Rusinkiewicz , O. Hall-Holt , M. Levoy. Real-Time 3D Model Acquisition. ACM Transactions on Graphics 2002, 21, 3.

[12] J, Salvi , J. Pagès , J. Batlle. Pattern codifi cation strategies in structured light systems. Pattern Recognition 2004.

[13] D. Scharstein , R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. IJCV 47(1/2/3):7-42, 2002.

[14] D. Scharstein , R. Szeliski. High-Accuracy Stereo Depth Maps Using Structured Light. CVPR 2003.

[15] J.-P. Tardif , S. Roy , M. Trudeau. Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction. 3DIM 2003 .

[16] S. Roy , J.-P. Tardif. LightTwist, a multi-projector system. Live performance, ArtFutura 2004.

[17] E. W. Weisstein. Gray Code, MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/GrayCode.html.

[18] R. Yang, D. Gotz, J. Hensley, H. Towles, M.S. Brown. PixelFlex: a reconfi gurable multi-projector display system. IEEE Visualization 2001.

[19] R., Andrew, G. Gill, A. Majumder, H. Towles , H. Fuchs. PixelFlex2: A Comprehensive, Automatic, Casually-Aligned Multi-Projector Display. PROCAMS 2003.