Deconvolution-Based Structured Light System with Geometrically Plausible Regularization

Drouin, Marc-Antoine; Godin, Guy

The 2008 International Congress on Image and Signal Processing (CISP2008) [Proceedings], 2008

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Deconvolution-Based Structured Light System with Geometrically Plausible Regularization *

Drouin, M.-A., Godin, G.
May 2008

* published at the 2008 International Congress on Image and Signal Processing (CISP2008). Sanya, China. May 27-30, 2008. NRC 49901.

Canada

Deconvolution-Based Structured Light System with Geometrically Plausible Regularization

Marc-Antoine Drouin and Guy Godin
Visual Information Technology Group
National Research Council Canada
{first name}.{last name}@nrc-cnrc.gc.ca

Abstract

This paper presents a new deconvolution-based energy formulation for segmenting the image of stripe-based patterns projected by structured light systems. Our framework features an explicit modeling of the blurring introduced by the lens of a structured light system. This allows a significant improvement when working out of focus, a situation which occurs when performing depth measurement. The proposed iterative algorithm includes two steps: a deconvolution and a segmentation. For both steps, a geometrically plausible regularization term is used. It considers the projected displacement induced by the camera, projector and scene configuration. We validate our method using real imagery acquired using off-the-shelf equipment.

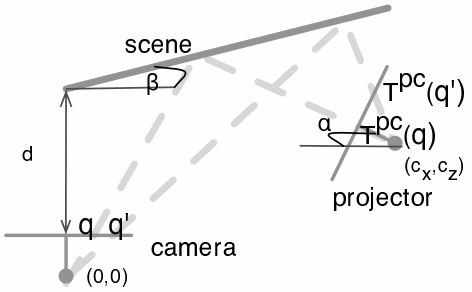

Figure 1. Representation of the structured light system. The distance along the optical axis between the camera and the scene (which is a plane) is d. The angle between the camera's image plane and scene plane is β, and α is the angle between the projector's optical axes and the camera's image plane. The center of projection of the projector is (c_x, c_z). The camera pixels $\mathbf{q}$ and $\mathbf{q}'$ correspond to projector pixel $T^{pc}(\mathbf{q})$ and $T^{pc}(\mathbf{q}')$ respectively.

1 Introduction

In the last decades, projector-camera systems have been used in many applications ranging from 3D reconstruction to multi-projector visualization. For many applications, the main task is to establish the correspondence between the pixels of a projector and those of a camera (see Fig 1). This is accomplished by projecting known light patterns on an unknown scene viewed by a camera. The projector "applies" textures to the scene and the patterns are designed (or structured) such that it is possible to identify which projector pixel illuminates the part of the scene viewed by a camera pixel. Different patterns can be used: a survey by Salvi *et al.* classified them in 3 categories, namely, direct codification, neighborhood codification and time-multiplexing. The Gray code is probably the best known time-multiplexing code[11]. It uses a succession of patterns composed of white and black stripes. For this reason, it is also known as a stripe-based code. The width of the stripes vary with the patterns. Whilst the examples in this paper use Gray code patterns, our approach is applicable to other stripe-based codes. We propose an energy framework for segmenting the Gray code patterns designed to work in harsh acquisition conditions. When working with off-the-shelf equipment, many sources of error affect the segmentation. We classify them as originating from projection, imaging or from the overall system:

- **Projection** Depth of field and aberration affect the detectability of spatial transitions. Calibration parameters vary due to thermal instability. Error is introduced by dithering and signal synchronization.
- **Imaging** Depth of field, aberration, anti-aliasing filter and small fill factor limit the detectability of the pattern. Also, when using color cameras, artifacts are introduced by Bayer color processing and the Bayer matrix itself.
- **Overall system** Surface properties and ambient light impact the Signal-to-Noise Ratio (SNR). It is also reduced by high frame rates, or non-uniform sensitivity of projection and camera.

Our framework explicitly models the blurring induced by the limited depth of field (camera and projector) and reduces the impact of the other error sources by using regularization. Blurring was described as a significant limitation of digital projector profilometry[1] and our framework allows a significant improvement when working out of focus, a situation which occurs when performing depth measurement. Our algorithm iteratively performs a deconvolution and a segmentation of the patterns. Moreover, it takes into consideration the projected displacement which is induced by the camera, the projector and the scene configuration. This allows the use of regularization terms which are geometrically plausible. Our algorithm is targeted at scenes composed of many quasi-planar surfaces. This type of situation may be encountered in industrial inspection and in planar calibration applications. The remainder of this paper is divided as follows: in Section 2, previous work is presented; Section 3 presents our formulation; the geometry plausible regularization is described in Section 4; the point spread function is discussed in Section 5; implementation details are presented in Section 6; experimental results are discussed in Section 7.

2 Previous work

A recent paper by Salvi *et al.* [23] contains an extensive survey of structured light (SL) techniques. Few methods use energy minimization frameworks for the decoding of patterns[5, 29, 13, 26, 14]. Some use it to remove ambiguity in their code [5, 29, 13], whilst others use it to reduce sensitivity to noise [26, 14]. In [17], the best focus distance is used as a constraint in the self-calibration of a SL system. To the best of our knowledge, our algorithm for decoding SL patterns is the only one to explicitly model blurring. There is a large body of literature on the problem of thresholding, segmentation and deconvolution [25, 19]. Some methods simultaneously segment and deconvolve the images [21, 2]. Their goal is to deconvolve images, while ours is to segment the patterns of a SL 3D scanner. The closest work to ours is [21] where a segmentation-based regularization term for image deconvolution is proposed. As will be shown, there are significant differences in how we incorporate the segmentation information into the deconvolution and in how we regularize the segmentation. Also, blind deconvolution and segmentation have been used for bar code thresholding [6]. This application is 1D, whilst the underlying structure of ours is 3D.

3 Our formulation

Our algorithm takes as input two images G and R of size $m \times n$. The first one, G, is an image of a static scene acquired while one of the Gray code patterns is projected. The image R is simply G with white and black stripes reversed and $R(\mathbf{q})$ is the intensity of image R at pixel $\mathbf{q}$. Using G and R allows an increase in robustness and we define $I = G - R$. When neglecting the noise, the image formation for I is described as $I(\mathbf{q}) = H(\mathbf{q}) * * X(\mathbf{q})$ where X is the non-degraded image of $G-R$ and H is the Point Spread Function (PSF) of the system which depends on the projector, the camera and the scene configuration. Note that this is different from conventional deconvolution where blurring is introduced only by the camera lens. The system PSF will be further described in Section 5 and for now it will be assumed that it is spatially invariant. With H and I known, the non-degraded image is the one minimizing

$$||\mathbf{Hx} - \mathbf{i}||^2 \tag{1}$$

where $\mathbf{H}$ is a mn by mn block Toeplitz matrix with Toeplitz blocks representing the convolution with the PSF H [9]. The vector $\mathbf{x}$ and $\mathbf{i}$ are stacks of the columns of the images X and I respectively and are of length mn. Since this deconvolution is *ill-conditioned* we propose to simultaneously segment and deconvolve the image I using the result of the segmentation to regularize the deconvolution. We add a regularization term based on the heuristic that two neighboring pixels should have the same intensity if they belong to the same class (they are either white or black). This allows to reduce the impact of noise. The matrix $\mathbf{S}$ contains, for each two-pixel neighborhood interaction, a row with two non-zero entries corresponding to the neighbors. Those entries are 1 and -1 respectively. Explicitly, each line of $\mathbf{Sx}$ represents an equation of the form $X(\mathbf{q}) - X(\mathbf{q}') = 0$ where $\mathbf{q}$ and $\mathbf{q}'$ are neighbor pixels. The neighborhood structure is presented in Section 4. The regularization term is

$$\lambda^2 ||\mathbf{M_d Sx}||^2 \tag{2}$$

where λ is a user-defined parameter and $\mathbf{M_d}$ is a diagonal matrix where each diagonal entry is the probability that all the pixels involved in the interaction of the corresponding row of $\mathbf{S}$ belong to the same class. While the previous regularization term improves the conditioning, the deconvolution is still *ill-posed*. In order to fix the scaling and anchor the solution, we add the regularization term

$$\gamma^2 \left(||\mathbf{M_b}(\mathbf{x} - \mathbf{u_b})||^2 + ||\mathbf{M_w}(\mathbf{x} - \mathbf{u_w})||^2 \right) \tag{3}$$

where γ is a user-defined parameter and $\mathbf{M_b}$ and $\mathbf{M_w}$ are a diagonal matrix for the black and white class respectively. The vector $\mathbf{u_b}$ and $\mathbf{u_w}$ are the expected intensity for the black and white class respectively. They are not necessarily spatially constant. Those vectors can be estimated by projecting white and a black images, or approximated using the image I directly or iteratively estimated as in [21]. Diagonal entries of $\mathbf{M_b}$ and $\mathbf{M_w}$ are the probabilities that the corresponding pixel of X belong to the black and white

class respectively. In [21], the segmentation of the image is used as part of the regularization term, while we use probability densities. When $\mathbf{M_b}$, $\mathbf{M_w}$, $\mathbf{M_d}$, $\mathbf{u_b}$ and $\mathbf{u_w}$ are fixed, the terms of Eq. 1, 2 and 3 form a Tikhonov regularization which may be solved iteratively [9]. The matrix $\mathbf{M_b}$, $\mathbf{M_w}$ and $\mathbf{M_d}$ depend on the segmentation which is described in the next section.

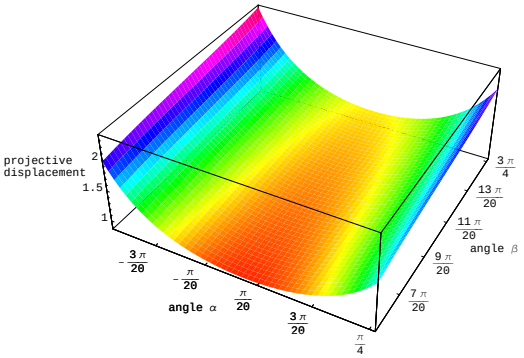

Figure 2. Projected displacement as a function of angle α and β for a projector and camera having a focal length of 5 and a sensor element size of 0.1. The distance along the optical axis between the camera and the plane is 200, The projector is located at (20,0) and the camera is located at the origin.

3.1 Segmentation

The set $\mathcal{P}$ contains the pixels of image X and the set $\mathcal{L} = \{w, b\}$ contains the label for white and black pixels. A $\mathcal{L}$-configuration $C : \mathcal{P} \mapsto \mathcal{L}$ associates a label to every pixel of $\mathcal{P}$. The probability of a configuration C given X is

$$p(C|X) = \frac{1}{Z} \prod_{q \in \mathcal{P}} p(\mathbf{q}|C(\mathbf{q})) \prod_{(\mathbf{q},\mathbf{q}') \in \mathcal{N}} p(C(\mathbf{q}), C(\mathbf{q}')) \tag{4}$$

where Z is the normalization factor of the Bayes formula and $\mathcal{N}$ is the set of pair of neighbor pixels. The probability $p(X(\mathbf{q})|C(\mathbf{q}))$ is a Gaussian distribution having mean identical to those of $\mathbf{u_b}$ and $\mathbf{u_w}$. In our implementation, the variance is assumed to be equal for both classes. We have observed that those hypotheses are acceptable. Also, $p(C(\mathbf{q}), C(\mathbf{q}'))$ is the probability of having a class transition between the pixel $\mathbf{q}$ and $\mathbf{q}'$ which is related to the projected displacement presented in the next section. For each pixel $\mathbf{q}$, we need in order to deconvolve X the probabilities $p(C|X, C(\mathbf{q}) = b)$ and $p(C|X, C(\mathbf{q}) = w)$, that is, the probabilities of the most likely segmentation of image X with the pixel $\mathbf{q}$ having the label black and white respectively. These probabilities can be approximated using loopy Belief Propagation which must compute these in order to find an approximate solution to the maximization in C of

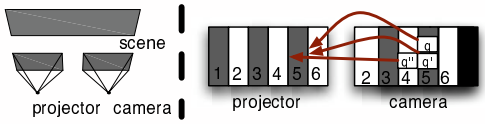

Figure 3. Left) Rectified projector-camera system with a fronto-parallel scene. **Right)** Mapping camera pixel to projector column for the configuration shown on the left side.

Eq. 4 [28]. Note that in our segmentation, the regularization is based on a geometric model and does not attempt to reduce the noise which is handled in Eq. 2.

Our algorithm starts by evaluating $p(C|X, C(\mathbf{q}) = b)$ and $p(C|X, C(\mathbf{q}) = w)$ for all pixels $\mathbf{q}$ which are used to fill the matrix $\mathbf{M_b}$ and $\mathbf{M_w}$. Here, we denote the diagonal entry of $\mathbf{M_b}$ corresponding to pixel $\mathbf{q}$ as $\mathbf{M_b}(\mathbf{q})$: it is set to $p(C|X, C(\mathbf{q}) = b)$. The matrix $\mathbf{M_w}$ is filled similarly and each row of $\mathbf{M_b}$ and $\mathbf{M_w}$ is then normalized such that $\mathbf{M_b} + \mathbf{M_w} = \mathbf{I}$. The matrix $\mathbf{M_d}$ can be computed from the normalized $\mathbf{M_b}$ and $\mathbf{M_w}$. The diagonal entry of $\mathbf{M_d}$ corresponding to two neighbors $\mathbf{q}$ and $\mathbf{q}'$ is $\mathbf{M_b}(\mathbf{q})\mathbf{M_b}(\mathbf{q}') + \mathbf{M_w}(\mathbf{q})\mathbf{M_w}(\mathbf{q}')$. The energy function containing the terms of Eqs. 1, 2 and 3 can be minimized using gradient descent. A few gradient descent iterations are performed before the re-computation of the probabilities. Those steps are repeated until a non-ambiguous segmentation of X is obtained or until a maximum number of iterations is reached. In our implementation, the first segmentation is performed directly on I.

4 Projected displacement and 3D geometry

The class transition probabilities used in the previous section depend on the 3D structure of the scene. Explicitly, they depend on the projected displacement which is induced by the scene geometry and the internal and external parameters of the projector and camera. Fig. 1 illustrates a 2D projector-camera system with a planar scene. The projected displacement ΔD is

$$\Delta D(\mathbf{q}, \mathbf{q}') = T^{pc}(\mathbf{q}) - T^{pc}(\mathbf{q}') \tag{5}$$

where T^{pc} is a function that applies the homography mapping a point on the camera plane onto the projector plane. Fig. 2 shows, for a 2D camera-projector system, the value of the projected displacement ΔD for varying angles α and β. Note that the variation is non-negligible. Also, ΔD is a scalar rather than a vector in this 2D case.

For now, we will assume that the camera and projector planes are rectified and that the scene is a single plane which is fronto-parallel to the SL system (see Fig. 3-left) [10]. In this case, the image viewed by the camera is that of the projector shifted along the horizontal axis. The magnitude of this displacement - that is, the disparity - depends

on the depth of the plane in the scene. Fig. 3-right shows the camera and projector images corresponding to such a scene when the stripes in the projector (and camera) are one pixel wide. The camera pixels $\mathbf{q}$ and $\mathbf{q}'$ belong to the class *black* and correspond to the same column in the projector. $\Delta D(\mathbf{q}, \mathbf{q}')$ is $(0, 0)$, whilst pixels $\mathbf{q}'$ and $\mathbf{q}''$ belong to different classes and are necessarily from adjacent columns (4 and 5) in the projector and $\Delta D(\mathbf{q}', \mathbf{q}'') = (1, 0)$. When a smoothing penalty is applied between the labels of $\mathbf{q}$ and $\mathbf{q}'$, the presence of a depth discontinuity is penalized. However, when smoothing is applied between $\mathbf{q}'$ and $\mathbf{q}''$, the absence of a depth discontinuity is penalized since a configuration having $\mathbf{q}'$ and $\mathbf{q}''$ match with the same projector pixel is favored. Thus, according to this rectified and fronto-parallel model with one pixel wide vertical stripe, the probability of transition between $\mathbf{q}'$ and $\mathbf{q}''$ is 1, while it is 0 between $\mathbf{q}$ and $\mathbf{q}'$. For a calibrated SL system in a general configuration, those probabilities can be estimated from the calibration parameters and the expected geometry of the scene. In our experiment, for the narrowest stripe, we set the probability of transition for horizontal neighbor to 0.5. This corresponds to a transition occurring at every 2 camera pixels and is the targeted value for a 3D scanner that would use our algorithm. For the vertical transitions, the probability is set to 0.15. This is a crude estimate based on the number of distinct objects expected in the scene and the slant of the surfaces (i.e. we expect 15 transitions in an interval of 100 pixels). Note that the neighborhood interaction having a transition probability of 0.5 can be removed from $\mathcal{N}$. This allows the use of a 2-connected neighborhood, making it possible to compute the exact conditional probabilities using dynamic programming. Note that, as the stripes become wider, the difference between horizontal and vertical transition probabilities becomes smaller since there are far less horizontal transitions that are expected, and the number of vertical transitions is expected to remain the same. Since those wide stripes are less ambiguous, we always use a 2-connected neighborhood for all patterns. This results in a reduction of the computational burden without affecting the quality. We use the same neighborhood structure in Eq. 2 than in Eq. 4.

5 The system PSF

The system PSF H used in our algorithm is defined as $H(\mathbf{q}) = H_c(\mathbf{q}) ** W(\mathbf{q}) ** H_p(\mathbf{q})$ where H_c and H_p are the PSF of the projector and camera and vary with the distance to the scene. These are conventional lens PSFs, while W is the scene convolution matrix which depends on the distance to the scene, the calibration parameters of the SL system and surface properties. As an example, on the rectified configuration with a fronto-parallel scene of Fig. 3, W would be a diagonal matrix horizontally shifted. The magnitude of the shift is equal to the disparity, and the magnitude of the diagonal entries depends on the surface albedo. Note that H may not sum to one because of the surface albedo. In our deconvolution framework, it is assumed that H does sum to one. However, the components of $\mathbf{u_b}$ and $\mathbf{u_w}$ vary spatially, which allows to accommodate for the albedo of the scene.

Since we expect the scene depth to vary, the system PSF will also vary across the image. Many approaches have been designed to cope with spatially varying PSF [3, 12, 22, 7, 15]. Some of them assume that the PSF is piecewise constant[12, 22, 7, 27], others use coordinate transformations [24, 20], while others use interpolation of the PSF across the image [3, 22, 15]. In this paper, we use an interpolation approach based on an Karhunen-Loève decomposition of the PSFs [15]. It allows a compact orthogonal decomposition of the PSFs. The bases are denoted $\mathbf{H^k}$. The minimization of Eq. 1 with regularization terms 2 and 3 becomes

$$|| \sum_k \mathbf{H^k C^k x} - \mathbf{i}||^2 + \text{regularization} \qquad (6)$$

where $\mathbf{C^k}$ is a diagonal matrix. Each diagonal entry is the coefficient that scales the individual pixel of $\mathbf{x}$ prior to the convolution. More details can be found in [15]. In this paper, we focus on the image processing aspects of the pattern segmentation and we assume that the PSFs at different points in the image are known (see the future works described in the conclusion). The PSFs for other points are interpolated. We thus have all $\mathbf{H^k}$ and $\mathbf{C^k}$ available beforehand.

6 Implementation details

During the deconvolution step of our algorithm the most expensive operation is the computation of $\mathbf{H^k C^k x}$ which is performed using a FFT [9]. The convolution for all Karhunen-Loève bases can be computed simultaneously. There are many commercially available products that could be used to efficiently perform 2D FFT on programmable hardware. The regularization terms of Eq. 2 and Eq. 3 can be computed in linear time. The minimization of Eq. 6 is performed using a simple steepest descent algorithm which converges slowly. Faster algorithms exist [9].

The segmentation can also be implemented on programmable hardware. An implementation of belief propagation on graphics hardware is presented in [4]. When using a two-connected neighborhood, as we do, all the columns can be processed simultaneously. It is possible to further increase the amount of parallelism by reformulating Eq. 4. This reformulation is presented in the Annex, and we implemented it on off-the-shelf graphics hardware. The decoding of SL patterns using a 1024×768 camera is accomplished at a rate of more than 120 patterns per second on an old

NVIDIA 6800 graphics card. General purpose computation using graphics hardware has greatly evolved in the last years and the proposed formulation can be implemented using the high-level programming language CUDA. For this reason, we do not describe further its implementation.

Since both steps of our algorithm can be implemented on programmable hardware, we believe that an implementation of our algorithm running at video frame-rate is possible. Our current software prototype runs on a computer with 8 Dual-Core AMD Opteron Processor 8218. At most 6 cores are used simultaneously. the FFT code and BLAS implementation of the ACML are used. We use the standard FFT driver of the ACML with OpenMP support enabled. Our non-optimized prototype processes one 640×480 image in 20 seconds.

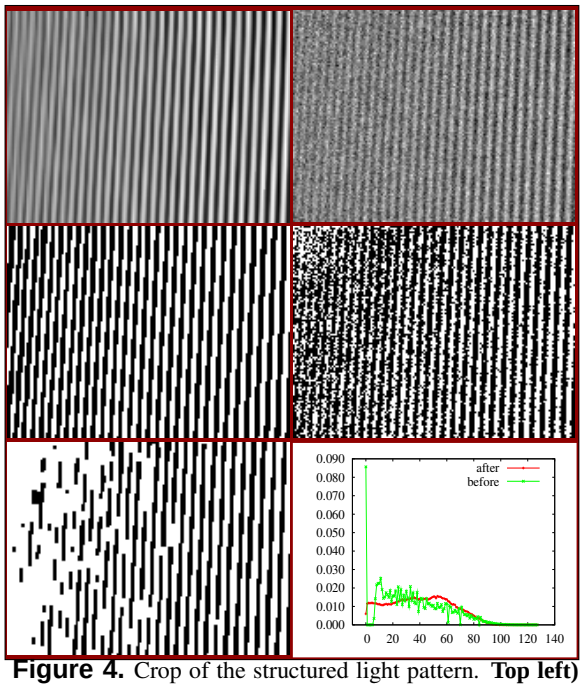

Figure 4. Crop of the structured light pattern. **Top left)** deconvolved image obtained by our algorithm. **Top Right)** original image **Middle left)** result of our algorithm **Middle right)** result of standard thresholding **Bottom left)** result from an Markovian segmentation **Bottom right)** difference histogram (see text).

7 Experimental results

Fig. 4 contains an example of Gray code decoded under difficult acquisition conditions. The scanned image was acquired with both camera and projector out of focus and with a low SNR. The scene is composed of a single tilt plane, thus making it easier to assess the quality of the segmentation by visual inspection. The result obtained by standard thresholding is shown [23]. Note how noisy the segmentation is. We also provide the results when segmenting using a Markovian framework. The method of [26] applied to one pattern was used. The method did significantly over-smooth the left part of the image. This over-smooth solution and the noisy one of thresholding are explained by the histogram of the distance between the threshold value and the pixel intensities (see Fig. 4). More than 8.6% of the pixels have an intensity equal to the threshold value, thus making those very difficult to classify. In our deconvolved image, less than 0.6 % of the pixel are in this situation, making the segmentation much easier to perform.

Residual error for different Gray code patterns		
pattern index	standard	ours
1	>0.351	0.127
2	0.150	0.125
3	0.180	0.133
4	0.138	0.133
5	0.143	0.139
6	0.143	0.134
7	0.129	0.114
8	0.190	0.157

Table 1. Pattern 1 has the narrowest stripe. The images of the first pattern are shown in Fig 4. For this pattern, some stripes are inter-connected in the solution obtained by standard thresholding and they were manually removed before computing the error. This manual removal is somewhat arbitrary and we show the most conservative values that could be obtained.

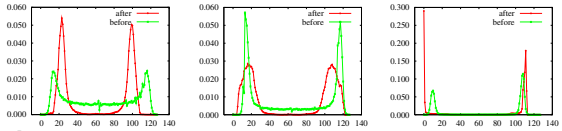

Figure 5. Intensity histogram before and after our algorithm for the second, third and fifth patterns of the Gray code. The images of the first pattern are shown in Fig 4.

Fig. 5 contains the intensity histogram of the images before and after the deconvolution for different sizes of stripes. Note the reduced inter-class ambiguity after applying the deconvolution. The pattern with large stripes are easier to decode since they contain fewer transitions. As an example, with the fifth pattern, both images have very few ambiguous pixels. Nevertheless, the number of pixels having an intensity in the interval ranging from 50 to 80 in the deconvolved image is smaller than the original by an order of magnitude (the class transition occurs at 63).

Since linearity is preserved under projective transform, a linear regression was performed along each label discontinuity in the camera. The residual error of the fit was used as a metric to compare our approach with standard decoding on the different patterns. Table 1 presents the results. The most significant improvement is for the pattern with the narrowest stripes. Fig. 6 contains graphs of the variation of the residual error with the change of λ for the patterns 1 to 3 of Table 1. Note that the minimum residual error occurs with the same λ for all patterns. In our tests, we observed that

the best λ is related to the noise level and seems independent of the width of the stripes. Note that γ was set to 0.5 in all our experiments. While λ is used to suppress noise, γ weighs the energy term of Eq. 3 which anchors the solution. This value was appropriate in all our tests and the algorithm seems much less sensitive to changes of γ than of λ.

Fig. 7 shows the images associated with a scan of planar surfaces acquired in even harder conditions. In this dataset, the first 3 patterns are difficult to segment and standard thresholding yielded useless results for those patterns. Between 30% and 66% of the pixels have the same intensity in both the image G and R. In the deconvolved image this drops to 3.3%, allowing a significant improvement in the decoding. Fig. 8 shows the images associated with a scan of two planes with different orientations.

8 Conclusion

We presented a new deconvolution-based energy formulation for segmenting the images of a stripe-based SL system. Our framework explicitly models the blurring induced by the optics of a SL system. The proposed iterative algorithm is composed of two steps, a deconvolution and a segmentation, and uses a geometrically plausible regularization. Our approach can decode SL patterns that cannot be adequately decoded by either energy-based segmentation or simple thresholding.

As for future work, we plan to investigate the integration in our framework of the estimation of the coefficients C^k based on the calibration (geometric and of the Karhunen-Loève basis) and on the 3D information obtained when combining the segmentation of all the Gray code patterns. We would also like to quantitatively assess the improvement into the 3D measurement that occurs when using the proposed algorithm in a SL 3D scanner.

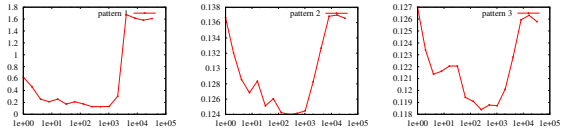

Figure 6. Resistance to the change of λ for patterns 1 to 3 of Table 1.

Acknowledgement

The authors would like to thank professor Sébastien Roy from Université de Montréal for insightful discussions about the geometric model described in section 4.

Annex: parallel dynamic programming

When the neighborhood of Eq. 4 is 2-connected the optimal solution can be computed using Dynamic Programming

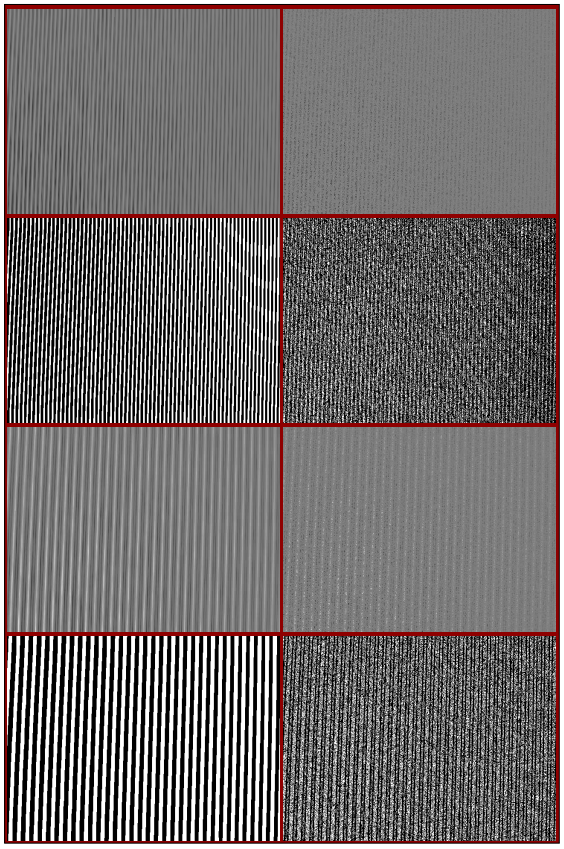

Figure 7. The first 4 images are for a pattern with narrow stripes. The 4 last are for wider stripes. Please zoom in the electronic version for small details. For each data set the order is **Top left)** deconvolved image obtained by our algorithm. **Top Right)** original image **Bottom left)** result of our algorithm **Bottom right)** result of standard thresholding.

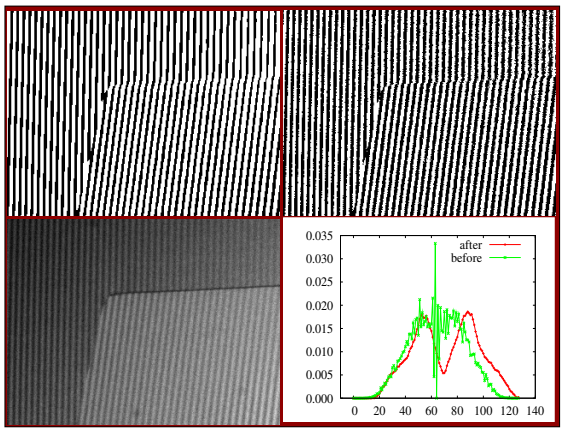

Figure 8. **Top left)** result of our algorithm. **Top right)** result of standard algorithm. **Bottom left)** image G (normalize for visualization). **Bottom rigt)** intensity histogram, note that it contains some discretization artefacts.

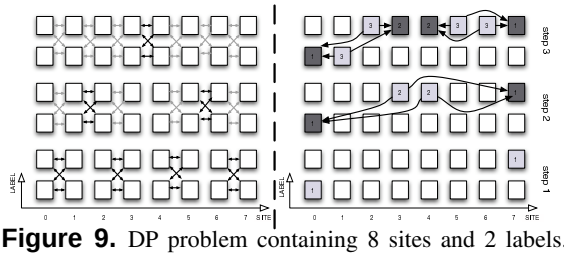

Figure 9. DP problem containing 8 sites and 2 labels. **Left)** Energy computation phase. The gray arrows represent the smoothing cost already being taken into account, whilst the black ones represent the smoothing cost computed at the current step. **Right)** Recovery phase. The light gray boxes represent sites for which the label is fixed in the current step. The dark gray boxes represent sites for which the label has been fixed in the previous step. The number on a site is the step at which the label is determined. The arrows indicate the dependency between sites. As an example, the labels of sites 0 and 3 must be known before computation of the label of sites 1 and 2.

(DP). The maximization of Eq. 4 can be reformulated as an energy minimization with a Potts smoothing model [16]. The division and multiplication of the maximization are replaced by summation and subtraction in the minimization framework, making it more attractive for an implementation using programmable hardware. More details can be found in [16]. When a Potts smoothing model is used, DP has a complexity of $\Theta(N.L)$ where N is the number of sites and L is the number of labels [8]. The depth of the recurrence relations is linear in the number of sites. Lui *et al.* proposed a GPU-based Smith-Waterman DP that exploits the parallelism already present in the standard recursive formulation [18]. We propose to use a divide-and-conquer strategy and to reduce the depth of the recurrence relation in order to increase the amount of parallel computation. This is done at the price of increasing the total amount of computation. We assume that the number of sites on a DP line is a power of two. Otherwise, the number of sites can be expanded to a power of two, by adding dummy sites and setting their likelihood cost to a constant value. In a GPU, this can be efficiently done using the texture's clamping mode of OpenGL. On the left of Fig. 9, we illustrate the concept on a small problem of 8 sites and 2 labels. In the first step, the energies for all the combinations of pairs of sites $(0,1)$, $(2,3)$, $(4,5)$ and $(6,7)$ are computed simultaneously. Four values are computed for each pair of sites. In the next step, the lowest energy for the combinations of labels of sites $(0,3)$ and $(4,7)$ are computed using the result of the previous step. Finally, the lowest energy for the combinations of labels of sites $(0,7)$ are computed. In general, this binary tree structure significantly reduces the number of steps required to process all the sites. The table $t(m_{i,j}, n_{i,j}, k, l)$ is the lowest energy of all maps of sites in the interval $[m_{i,j}, n_{i,j}]$ with site $m_{i,j}$ and $n_{i,j}$ at label k and l respectively. We define $m_{i,j} = i2^j$ and $n_{i,j} = m_{i+1,j} - 1$. Explicitly, the table t is defined inductively as

$$t(m_{i,1}, n_{i,1}, k, l) = \begin{array}{l} e(m_{i,1}, k) + e(n_{i,1}, l) \\ +s(m_{i,1}, n_{i,1}, k, l) \end{array} \quad (7)$$

where $e(m,k)$ is the likelihood term related to the probability that site m has label k and $s(m,n,k,l)$ is related to the probability that two neighbors m and n have label k and l respectively [16]. For $j > 1$

$$t(m_{i,j}, n_{i,j}, k, l) = \min_{k',l' \in \mathcal{L}} t'(i,j,k,l,k',l') \quad (8)$$

with

$$t'(i,j,k,l,k',l') = \begin{array}{l} t(m_{i,j}, m'_{i,j}, k, k') \\ +t(n'_{i,j}, n_{i,j}, l', l) \\ +s(m'_{i,j}, n'_{i,j}, k', l') \end{array} \quad (9)$$

and where $n'_{i,j} = (2i+1)2^{j-1}$ and $m'_{i,j} = n'_{i,j} - 1$. The entry of t can be evaluated in $\Theta(N \cdot L^2)$ operations where N is the number of sites and L the number of labels. This is more than the $\Theta(N \cdot L)$ of ordinary DP with the Potts smoothing model. However, the depth of the relation for our formulation is in $\Theta(logN)$ rather than $\Theta(N)$. The minimal energy is $\min_{k,l} t(0, N-1, k, l)$ where $N-1$ is the index of the last site. However, the entry of table t are not what we required in order to minimize Eq. 6. We need the minimum energy of a label map having site q at label k for all q and k. This is also known as the *min marginal*. Those values can be computed from a table $v(m_{ij}, n_{ij}, k, l)$ which is the lowest energy map (including all sites) with sites m_{ij} and n_{ij} having label k and l respectively. Explicitly, the table v is defined inductively as

$$v(0, N-1, k, l) = t(0, N-1, k, l) \quad (10)$$

and for $\{m_{ij}, m'_{ij}, n'_{ij}, n_{ij}\} \subseteq [0, N-1]$,

$$v(m_{i,j}, m'_{i,j}, k, k') = \min_{l,l' \in \mathcal{L}} v'(i,j,k,l,k',l') \quad (11)$$

$$v(n'_{i,j}, n_{i,j}, l', l) = \min_{k,k' \in \mathcal{L}} v'(i,j,k,l,k',l') \quad (12)$$

with

$$v'(i,j,k,l,k',l') = \begin{array}{l} v(m_{i,j}, n_{i,j}, k, l) \\ -t(m_{i,j}, n_{i,j}, k, l) \\ +t'(i,j,k,l,k',l'). \end{array} \quad (13)$$

The minimum energy of a label map having site $m_{i,j}$ at label k is simply $\min_l v(m_{i,j}, n_{i,j}, k, l)$. This can be computed similarly for a site $n_{i,j}$. Using the equivalence between probabilistic and energy formulations[16], the entries of table v can be converted into the conditional probabilities required in order to fill the matrix $\mathbf{M_b}$ and $\mathbf{M_w}$.

Sometimes, the label map of minimum energy is required rather than the min marginal. Fig. 9-right shows the

recovery process of this label map of minimum energy for a 2-label problem containing 8 sites. At step 1, the labels of sites 0 and 7 are computed by looking only at the minimum entry of table t for the pair of sites $(0, 7)$. At step 2, the labels of site 3 and 4 are found by using t for the pair $(0, 3)$ and $(4, 7)$ combined with the result of step 1 and the smoothing cost between site 3 and 4. The process is similar for step 3, except that the results of all previous steps are needed. The label map $c_{i,j}^{k,l}$ is the lowest energy map for all sites in the interval $[m_{i,j}, n_{i,j}]$ having sites $m_{i,j}$ and $n_{i,j}$ at label k and l respectively. Explicitly, the label map $c_{i,j}^{k,l}$ is defined inductively as

$$(c_{i,j}^{k,l}(m_{i,j}), c_{i,j}^{k,l}(n_{i,j})) = (k, l) \tag{14}$$

and for $\{m'_{i'j'}, n'_{i'j'}\} \subseteq]m_{i,j}, n_{i,j}[$

$$\begin{aligned}(c_{i,j}^{k,l}(m'_{i'j'}), c_{i,j}^{k,l}(n'_{i'j'})) = \arg\min_{k',l' \in \mathcal{D}} \\ t'(i', j', c_{i,j}^{k,l}(m_{i',j'}), c_{i,j}^{k,l}(n_{i',j'}), k', l').\end{aligned} \tag{15}$$

The depth of the relation is also in $\Theta(\log N)$.

References

[1] M. Baker and J. Chcharo. Accuracy limitations introduced by digital projection sources in profilometric optical metrology systems. In *Conf. on Optoelectronic and Microelectronic Materials and Devices*, pages 261–264, 2004.

[2] L. Bar, N. Sochen, and N. Kiryati. Variational pairing of image segmentation and blind restoration. In *Europ. Conf. on Comput. Vision*, pages 166–177, 2004.

[3] A. Boden, D. Redding, R. Hanisch, and J. Mo. Massively parallel spatially-variant maximum likelihood restoration of Hubble space telescope imagery. *J. Opt. Soc. Am. A*, 13:1537–1545, 1996.

[4] A. Brunton, C. Shu, and G. Roth. Belief propagation on the GPU for stereo vision. In *Canad. Conf. on Comput. and Robot Vision*, 2006.

[5] C. Chen, Y. Hung, C. Chiang, and J. Wu. Range acquisition using color structured lighting and stereo vision. *Image Vision Comput.*, 15:445–456, 1997.

[6] S. Esedoglu. Blind deconvolution of bar code signals. *Inverse Problems*, 20:121–135, 2004.

[7] M. Faisal, A. D. Lanterman, D. L. Snyder, and R. L. White. Implementation of a modified Richardson-Lucy method for image restoration on a massively parallel computer to compensate for space-variant point spread function of a charge-coupled device camera. *J. Opt. Soc. Am. A 12*, 1995.

[8] M. Gong and Y.-H. Yang. Fast stereo matching using reliability-based dynamic programming and consistency constraint. In *Int. Conf. on Comput. Vision*, 2003.

[9] P. C. Hansen. Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29, 2002.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004.

[11] S. Inokuchi, K. Sato, and F. Matsuda. Range imaging system for 3-D object recognition. In *Int. Conf. on Pattern Recogn.*, pages 806–808, 1984.

[12] J. Kamm and J. G. Nagy. Kronecker product and SVD approximations in image restoration. *Linear Algebra and its Applications*, 284(1-3):177–192, 1998.

[13] T. P. Koninckx, I. Geys, T. Jaeggli, and L. Van Gool. A graph cut based adaptive structured light approach for real-time range acquisition. In *Int. Symp. on 3D Data Process., Vis. and Transmiss.*, 2004.

[14] T. P. Koninckx, P. Peers, P. Dutre, and L. Van Gool. Scene-adapted structured light. In *IEEE Conf. on Comput. Vision and Pattern. Recogn.*, pages 611–618, 2005.

[15] T. Lauer. Deconvolution with a spatially-variant PSF. In *Astronomical Data Analysis II*, volume 4847, 2002.

[16] S. Li. *Markov random field modeling in computer vision*. Springer-Verlag, 1995.

[17] Y. F. Li and S. Y. Chen. Automatic recalibration of an active structured light vision system. *IEEE J. Robotics and Automation*, 19(2), 2003.

[18] Y. Liu, W. Huang, J. Johnson, and S. Vaidya. GPU accelerated Smith-Waterman. In *Int. Conf. on Comput. Science*, pages 188–195, 2006.

[19] L. Lucchese and S. Mitra. Color image segmentation: A state-of-the-art survey. *Image Processing, Vision, and Pattern Recognition*, 67(2), 2001.

[20] S. R. McNown and B. R. Hunt. Approximate shift-invariance by warping shift-variant systems. In *The Restoration of HST Images and Spectra II*, pages 181–187, 1994.

[21] M. Mignotte. A segmentation-based regularization term for image deconvolution. *IEEE Trans. Image Process.*, 2006.

[22] J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing*, 19(4):1063–1082, 1998.

[23] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recogn.*, 37(4), 2004.

[24] A. A. Sawchuk. Space-variant image restoration by coordinate transformations. *J. Opt. Soc. Am. 64*, 1974.

[25] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electron. Imaging*, 13(1), 2004.

[26] J. Tardif and S. Roy. A MRF formulation for coded structured light. In *3-D Digit. Imag. and Model.*, 2005.

[27] H. J. Trussell and B. R. Hunt. Image restoration of space-variant blurs by sectional methods. *IEEE Trans. Acoust. Speech, Signal Processing 26*, pages 608–609, 1978.

[28] J. Yedidia, W. Freeman, and Y. Weiss. *Exploring Artificial Intelligence in the New Millennium*. 2003.

[29] J. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Int. Symp. on 3D Data Process., Vis. and Transmiss.*, pages 24–36, 2002.