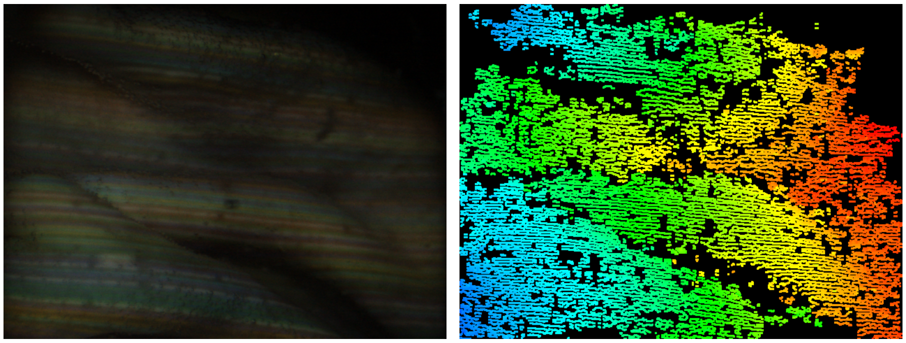
Belief Propagation for Improved Color Assessment in Structured Light

Christoph Schmalz[1,2] and Elli Angelopoulou[1]

[1]Pattern Recognition Lab, University of Erlangen-Nuremberg, Germany
[2]Siemens CT T HW 2, Munich, Germany

Abstract. Single-Shot Structured Light is a well-known method for acquiring 3D surface data of moving scenes with simple and compact hardware setups. Some of the biggest challenges in these systems is their sensitivity to textured scenes, subsurface scattering and low-contrast illumination. Recently, a graph-based method has been proposed that largely eliminates these shortcomings. A key step in the graph-based pattern decoding algorithm is the estimation of color of local image regions which correspond to the vertex colors of the graph. In this work we propose a new method for estimating the color of a vertex based on belief propagation (BP). The BP framework allows the explicit inclusion of cues from neigboring vertices in the color estimation. This is especially beneficial for low-contrast input images. The augmented method is evaluated using typical low-quality real-world test sequences of the interior of a pig stomach. We demonstrate a significant improvement in robustness. The number of 3D data points generated increases by 30 to 50 percent over the plain decoding.

1 Introduction



(a) Input image taken inside a pig stomach

(b) Result of the proposed decoding method with color enhancement

Fig. 1: Example input image and color coded depthmap result. Range is 142mm to 164mm.

Structured Light is a general term for many different methods for measuring 3D surface data. The main idea is to project a known illumination pattern on the scene. Shape information is then extracted from the observed deformation of the pattern (figure 1). The most basic hardware setup consists of one camera and one projector, but the details of the implementations vary widely. A survey of the various pattern designs that have been proposed in the literature can be found in [1]. Range data is generated by triangulation of the camera ray and the projection ray to a point in the scene. This requires solving the correspondence problem: determining which pairs of rays belong together.

One way to address this issue is temporal encoding, like the Gray Code and the Phase Shifting techniques (e.g. [2]). In these methods, the correspondences are determined by illuminating the scene with a sequence of patterns which is unique for each single pixel. This imposes the limitation that the object may not move while the sequence is acquired. Another approach to solve the correspondence problem is through the use of spatial encoding, where the necessary disambiguating information is encoded in a spatial neighborhood of pixels. This requires that the neighborhood stays connected, which means that the object must be relatively smooth. Nonetheless, spatial encoding has the advantage that only one pattern suffices to generate 3D data. This makes it particularly suitable for moving scenes. It also allows the use of simpler hardware, which in turn results in high scalability from millimeter to meter range. Miniaturization is of high interest in order to build Structured Light-based 3D video endoscopes for medical as well as industrial applications.

We recently proposed a graph-based Single-Shot Structured Light method [3], which shows considerable improvement in robustness in the presence of texture and noise. An essential component of the algorithm is the extraction of the representative color for local regions in the striped image. Ideally, such a color descriptor should be relatively invariant to cross-channel effects and surface reflectivity. The original algorithm used simply the median color of all the pixels in a local region. Though statistically robust, such a measurement does not explicitly address the properties that the region-color descriptor should satisfy. Thus, we propose a belief propagation-based color enhancement step that specifically tries to infer the illuminant color for the particular region using cues from neighboring regions. Thus the influence of the object color is minimized and contrast is enhanced. We evaluate the method with real-world example image sequences, and show an increase of 30% to 50% in the number of data points generated.

2 Single-Shot Structured Light

The performance of a Structured Light-based sensor depends crucially on the pattern that is used. Many different single-shot pattern designs have been proposed. Most of them are based on pseudorandom sequences (1D) or arrays (2D) [4],[5]. They have the property that a given window of size N or NxM occurs at most once. This is known as the *window uniqueness property*. Observing such a window suffices for deducing its position in the pattern. Pattern design involves two trade-offs. One concerns the size of the building blocks of the pattern, the so-called primitives. To achieve a high resolution, small pattern primitives are needed, but the smaller the primitives, the harder it is to reliably detect them. The other one is the alphabet size of the code, i.e. the number of different symbols that are used. Ideally, one wants to use a large alphabet for a long code with a small window size. However, the smaller the differences between individual code symbols, the less robust the code.

A well known single-shot 3D scanning system is the one by Zhang et al. [6]. The color stripe pattern used in that system is based on pseudorandom De Brujin sequences [7]. The decoding algorithm works per scanline and is based on dynamic programming. Its largest drawback seems to be the high processing time of one minute per frame. Koninckx and van Gool [8] present an interesting approach based on a black-and-white line pattern with one oblique line for disambiguation. It runs at about 15Hz and the resolution is given as 10^4 data points per frame, which is also relatively low. A recent paper by Kawasaki et al. [9] uses a pattern of vertical and horizontal lines. It is one of the few articles containing quantitative data about the accuracy of the methodology, which is given as 0.52mm RMS error on a simple test scene. The runtime is 1.6s per frame, but there is no information on the number of points reconstructed per frame. Forster [10] uses a color stripe pattern with scanline-based decoding, running at 20Hz and with up to 10^5 data points per frame. The RMS error of a plane measured at a distance of 1043mm is given as 0.28mm. Our system [3] is also based on color stripes but uses a graph-based decoding algorithm which offers superior robustness. With our method we can generate up to 10^5 data points per frame at 15 frames per second. The accuracy is 1/1000 of the working distance. We improve upon this method by including an additional graph-based inference step to enhance the observed colors of the pattern.

3 Graph-Based Pattern Decoding

In [3] we describe a series of steps for decoding the observed pattern. They are:

1. Finding a superpixel representation of the image: This is achieved with a watershed transform [11].
2. Building the region adjacency graph: Each basin of the watershed segmentation corresponds to one vertex of the graph. Neighboring basins are connected by edges.
3. Assigning edge symbols and probabilities: Edges usually connect vertices of different color. Given the knowledge about the projected pattern, there is only a finite number of possibilities of how color can change between adjacent vertices. The probabilities for each color change are computed.
4. Find a unique path of edges: Use the window uniqueness property of the pattern to solve the correspondence problem.
5. Recursively visit all neighbors in a best-first-search: Once the stripe number of a start vertex is known, propagate this information to all its neighbors, as long as the connecting edges are in accordance with the pattern.

We improved this method by adding an optional additional step 2b: Recover the projected color for each vertex. In the original algorithm the colors of the vertices are determined with a median filter over all image pixels belonging to the corresponding watershed basin. However, this observed color is not the original projected color. There are many alterations introduced by the object texture, scattering, blurring, camera crosstalk and so on. To recover the projected color, we can use an inference algorithm. The output of step 2 and thus the input for the new step 2b is a set of vertices and a set of edges. The color of the vertices is to be re-estimated by explicitly incorporating information about the color changes C across the edges to all neighbors.

$$C = [c_r\, c_g\, c_b]^T \in R^3 \tag{1}$$

The elements of the vector are equalized and normalized so that the maximum absolute value is 1. For details refer to [3]. Each edge also has a scalar edge weight w, which is defined as the L_∞-norm of the original unnormalized color change $\hat{C}$.

$$w = ||\hat{C}||_\infty = max(|\hat{c}_r|, |\hat{c}_g|, |\hat{c}_b|) \tag{2}$$

In the pattern used in [3] there are only eight projected colors (the corners of the RGB cube). This means that the number of labels is rather low and the inference can be performed in real time. Furthermore, the three color channels are independent. Thus we can perform a per-channel inference with binary labels. At a given vertex, a given color channel can only be either on or off.

4 Color Enhancement

We use Belief Propagation [12,13] to implement the color enhancement. BP is an iterative message passing algorithm. Each node of the graph receives messages from its neighbors containing their current beliefs about their state. This incoming information is combined with local evidence and passed on. Assuming pairwise cliques, the update equation for the message from node i to node j at time t is

$$m_{ij}^{t+1}(x_j) = \sum_i f_{ij}(x_i, x_j) g_i(x_i) \prod_{k \in N_i \setminus j} m_{ki}^t(x_i) \tag{3}$$

Here $f_{ij}(x_i, x_j)$ is the smoothness term for assigning labels x_i and x_j to nodes i and j, $g_i(x_i)$ is the data term for assigning x_i to node i and $N_i \setminus j$ is the neighborhood of node i excluding node j. After convergence, the final belief b_i is

$$b_i \propto g_i(x_i) \prod_{k \in N_i} m_{ki}^t(x_i) \tag{4}$$

In our case there is no data term, i.e. $g_i(x_i) = 1$, as we do not judge the absolute color values but only the color changes. Since we split the inference into three binary problems (one per RGB channel), there are only two labels, namely channel on or off. The smoothness term $f_{ij}(x_i, x_j)$ can therefore be written as a 2x2 compatibility matrix **F**.

$$\mathbf{F} = \begin{bmatrix} p_{constant} & p_{rising} \\ p_{falling} & p_{constant} \end{bmatrix} \tag{5}$$

On the diagonal we have the probability of the channel state being constant, as both nodes have the same label. If node i is in “off” state (index 0) and j in “on” state (index

1) the channel must rise, or fall in the opposite case. The values are computed with a truncated linear model:

$$p = \begin{cases} max(0, 1 - h_{bp} - c_l h_{bp}) & if \quad falling \\ max(0, 1 - |c_l| h_{bp}) & if \quad constant \\ max(0, 1 - h_{bp} + c_l h_{bp}) & if \quad rising \end{cases} \tag{6}$$

Here c_l is an indiviual normalized channel of the color change vector from eq.1 and h_{bp} is the slope of the probability function. This means the that the probability for a falling channel is 1 if c_l is -1 and goes down to 0 as the deviation of c_l from the ideal value approaches $\frac{1}{h_{bp}}$. For the cases of constant and rising channels, the ideal values are 0 and +1. A typical value for h_{bp} is $\frac{3}{2}$.

The most widely used message update equation in BP is the sum-product formula shown in 3. It is used when the focus is to approximate marginals. Other variants, however, have been used. For example, a popular variant is the max-product (or max-sum in the log domain) [12]. In this latter formulation, beliefs no longer estimate marginals. Instead, they are scoring functions whose maxima point to most likely states. In our application, we want a belief estimate that can capture the fact that not all messages are equally reliable. A belief that better captures the influence of the neighboring nodes, and is at the same time more robust to outliers, is one based on weighted sums. Thus we replace the products in eqs. 3 and 4 by weighted sums.

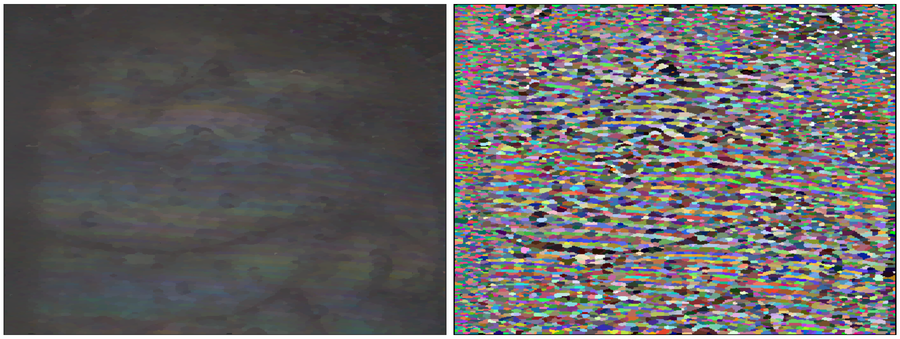
$$m_{ij}^{t+1}(x_j) = \sum_i f_{ij}(x_i, x_j) \sum_{k \in N_i \setminus j} m_{ki}^t(x_i) w_k \tag{7}$$

$$b_i \propto \sum_{k \in N_i} m_{ki}^t(x_i) w_k \tag{8}$$

This has two advantages. The first is that we can include our information about the edge weights (eq.2). High weight edges are more reliable as they are less likely to be distorted by noise. The second is that when using products to combine the incoming messages, one “bad” message with a probability of zero makes the product zero. This is especially the case for the so called null edges (see [3]). For this type of edges many entries in f_{ij} are zero and we end up with all-zero messages. In the weighted sum this does not happen. In fact, null edges have only a small influence because of their low weight.

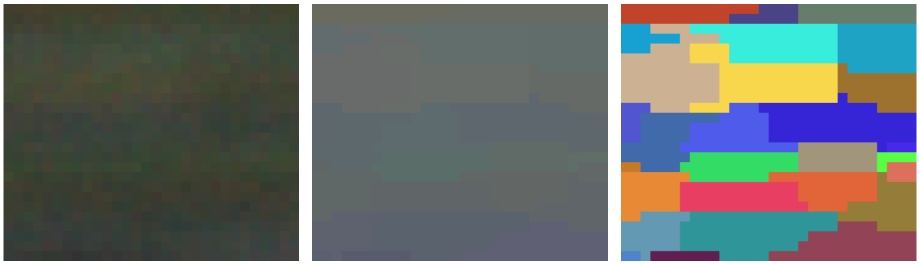
We initialize the messages as $m_{ij}^0(x_j) = 1$, i.e. we make no assumptions whether a given channel is on or off in the beginning. Because of the low number of labels the message passing converges in only two iterations. Afterwards, we can form a belief vector B that gives the probability of each of the color channels being in “on” state. This belief vector can be interpreted as a color again. We call this the enhanced color. Note that since the estimation of B is based solely on the color changes between the superpixels it ideally contains only information on the projected light. The result is shown in figure 2 and 3.

Although the image looks confusing, the contrast between the new colors is better and they are better suited for decoding in the following steps. We therefore compute new



(a) Original colors (b) BP-enhanced colors

Fig. 2: Color enhancement example



(a) Original image (b) Median filtered colors (gamma adjusted) (c) Color belief scaled to [0;255]

Fig. 3: Detail view of color enhancement results

color change vectors C for each edge in the graph, this time representing the change in our belief in the colors. Table 1 shows the improvement numerically on the basis of the blue and green regions shown in the center of figure 3c. The enhanced values are much closer to the ideal.

	ideal	median color	BP-enhanced color
region color a	(0,0,1)	(15,22,25)	(0.31,0.36,0.92)
region color b	(0,1,0)	(14,24,21)	(0.20,0.87,0.40)
color change	(0,+1,-1)	(-0.20,+0.50,-1.00)	(-0.21,+0.97,-1.00)

Table 1: Example of enhanced region colors. Note that the range of the original colors is [0;255] and the range of the enhanced colors is [0;1]. The color changes are normalized.

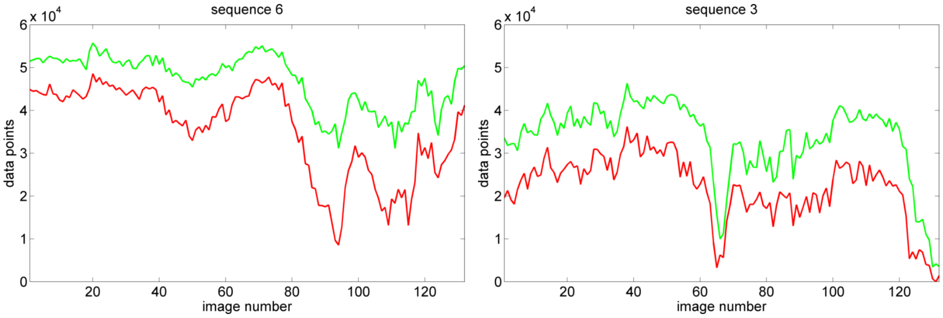
Note that this inference approach can also be applied to patterns with more colors. In that case there are more than two labels per channel and $\mathbf{F}$ is larger. We also experimented with a Graph-Cut optimization [14] to find the optimal labeling. This also works, but the results were inferior, as the output was a "hard" labeling as opposed to the "soft" labeling produced by BP.

5 Results

The proposed decoding method inherits the robustness of [3], which is due to: a) the rank filtering used to assign the superpixel colors, and b) the graph decoding algorithm that allows it to sidestep disruptions in the observed pattern. The BP-based enhanced color deduction further increases the robustness, especially in low-contrast situations, where single edges may be unreliable. In that case integrating the information from all neighbors before making a decision is especially helpful. This is a crucial improvement for medical purposes, where it can counteract the sub-surface scattering in skin and other tissue.

The biggest benefit of the proposed method is the increase in the number of recovered pixels. The image gradients used for the final triangulation of the distance are unaffected. Experiments with ground truth data have confirmed that the accuracy of the recovered depth map does not change. A detailed analysis on the accuracy of the overall methodology can be found in our prior work [3]. Here we focus on the improvement in the number of data points achieved with the additional color enhancement step. Figure 4 shows the number of data points that could be generated for each frame of two example videos. In sequence 6, which has the better image quality, the overall improvement was 32%, in sequence 3 with very poor image quality we gained 51%. The sequences were recorded with a handheld prototype scanner submersed in a liquid-filled pig stomach.

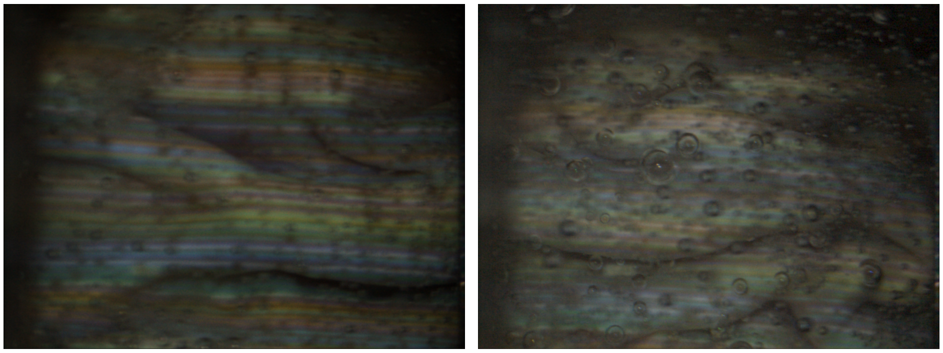
Figure 5 shows the results for two individual frames from each test sequence. For further comparison we also include the output of a reimplementation of the classic dynamic programming decoding method [6]. As can be seen, [6] is more susceptible to low image quality. Another example image with results is displayed in figure 7. Without



(a) Results for pig stomach sequence 6 with higher image quality.

(b) Results for pig stomach sequence 3 with poor image quality.

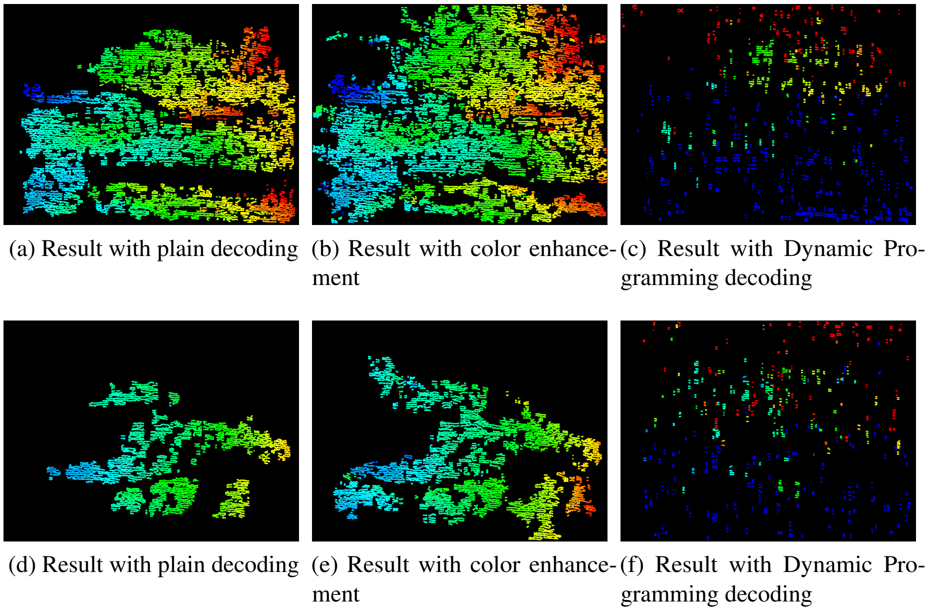
Fig. 4: The number of recovered pixels without (red) and with (green) the color enhancement



(a) Frame 90 of sequence 6

(b) Frame 90 of sequence 3

Fig. 5: Single frames from sequence 6 and sequence 3


(a) Result with plain decoding (b) Result with color enhancement (c) Result with Dynamic Programming decoding

(d) Result with plain decoding (e) Result with color enhancement (f) Result with Dynamic Programming decoding

Fig. 6: Color coded depthmap results for the image in figure 5a (first row, range is 137mm to 146mm) and the image in figure 5b (second row, range is 148mm to 166mm)

the color enhancement [3] generated 29391 data points, with color enhancement we get 40107 points. This is again an improvement of 36%.

6 Conclusion and Future Work

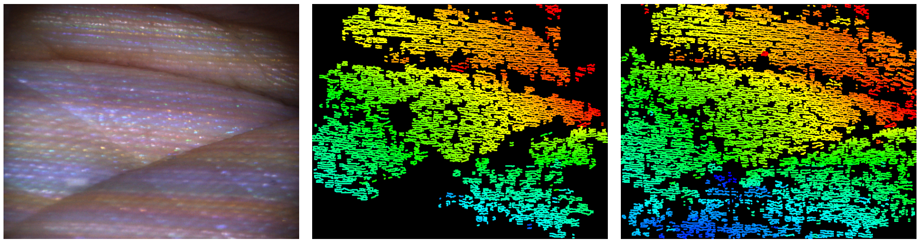
We presented an extension to the robust decoding algorithm for Single-Shot Structured Light patterns presented in [3]. It works even under very adverse imaging conditions, where previous methods like Dynamic Programming fail. It improves the data yield in the test sequences by 30% respectively 50% over the plain graph-based decoding. The method can tolerate low contrast, high noise as well as other artifacts and can run at 10 Hz with input images of 780x580 pixels on a 3Ghz machine, generating up to 10^5 data points per frame. As before, the typical accuracy is 1/1000 of the working distance. We have also demonstrated the miniaturization potential with the pig stomach images.

Acknowledgements

We would like to thank Philip Mewes for the chance to record the pig stomach data.

References

1. Salvi, J., Fernandez, S., Pribanic, T., Llado, X.: A state of the art in structured light patterns for surface profilometry. Pattern Recognition **In Press, Corrected Proof** (2010) – 2



(a) Input image (b) Result with plain decoding (c) Result with color enhancement

Fig. 7: “Palm” image and color coded depthmap result. Range is 119mm to 134mm.

2. Sansoni, G., Carocci, M., Rodella, R.: Three-dimensional vision based on a combination of gray-code and phase-shift light projection: Analysis and compensation of the systematic errors. Appl. Opt. **38**(31) (1999) 6565–6573 2
3. Schmalz, C., Angelopoulou, E.: Belief propagation for improved color assessment in structured light. In: 7th IEEE International Workshop on Projector-Camera Systems (PROCAMS), CVPR 2010. (2010) 2, 3, 4, 5, 7, 9
4. Paterson, K.G.: Perfect maps. IEEE Transactions on Information Theory **40**(3) (May 1994) 743–753 2
5. Mitchell, C.J.: Aperiodic and semi-periodic perfect maps. IEEE Transactions on Information Theory **41**(1) (Jan. 1995) 88–95 2
6. Zhang, L., Curless, B., Seitz, S.M.: Rapid shape acquisition using color structured light and multi-pass dynamic programming. In: Proc. First International Symposium on 3D Data Processing Visualization and Transmission. (19–21 June 2002) 24–36 3, 7
7. Annexstein, F.: Generating de bruijn sequences: An efficient implementation. IEEE Transactions on Computers **46**(2) (1997) 198–200 3
8. Koninckx, T.P., Van Gool, L.: Real-time range acquisition by adaptive structured light. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(3) (March 2006) 432–445 3
9. Kawasaki, H., Furukawa, R., Sagawa, R., Yagi, Y.: Dynamic scene shape reconstruction using a single structured light pattern. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR ’08. (june 2008) 1 –8 3
10. Forster, F.: A high-resolution and high accuracy real-time 3d sensor based on structured light. In: 3D Data Processing Visualization and Transmission, International Symposium on, Los Alamitos, CA, USA, IEEE Computer Society (2006) 208–215 3
11. Roerdink, J.B.T.M., Meijster, A.: The watershed transform: definitions, algorithms and parallelization strategies. Fundam. Inf. **41**(1-2) (2000) 187–228 3
12. Wainwright, M.J., Jordan, M.I.: Graphical Models, Exponential Families, and Variational Inference. Now Publishers Inc., Hanover, MA, USA (2008) 4, 5
13. Felzenszwalb, P.F., Huttenlocher, D.R.: Efficient belief propagation for early vision. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004. Volume 1. (27 June–2 July 2004) I–261–I–268 4
14. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(11) (Nov. 2001) 1222–1239 7