Linköping University Post Print

Extending Graph-Cut to Continuous Value Domain Minimization

Michael Felsberg

N.B.: When citing this work, cite the original article.

©2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Michael Felsberg, Extending Graph-Cut to Continuous Value Domain Minimization, 2007, Canadian Conference on Computer and Robot Vision, 274.

<http://dx.doi.org/10.1109/CRV.2007.29>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-38252>

Extending Graph-Cut to Continuous Value Domain Minimization*

Michael Felsberg
Computer Vision Laboratory
Linköping University, Sweden

E-mail: mfe@isy.liu.se

Abstract

In this paper we propose two methods for minimizing objective functions of discrete functions with continuous value domain. Many practical problems in the area of computer vision are continuous-valued, and discrete optimization methods of graph-cut type cannot be applied directly. This is different with the proposed methods. The first method is an add-on for multiple-label graph-cut. In the second one, binary graph-cut is firstly used to generate regions of support within different ranges of the signal. Secondly, a robust error minimization is approximated based on the previously determined regions. The advantages and properties of the new approaches are explained and visualized using synthetic test data. The methods are compared to ordinary multi-label graph-cut and robust smoothing for the application of disparity estimation. They show better quality of results compared to the other approaches and the second algorithm is significantly faster than multi-label graph-cut.

1. Introduction

In this paper we address the issue of non-linear optimization of discrete functions with *continuous* value domain. The wide-spread and popular graph-cut method originally addresses only binary labeling problems and various algorithms from the literature use graph-cut to find local minima of multiple-label problems. To find the global minimum of a multiple-label non-convex objective function is NP-hard and hence not tractable. The continuous problem is therefore not solvable in polynomial time either, but we propose a good approximation.

We adapt to the mathematical notation from [1]. Although we only show examples with 1D value domain,

*This work has been supported by EC Grant IST-2003-004176 COSPAL. This paper does not represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

i.e., we remove noise, results generalize to higher dimensions and a more general clustering method is obtained, see also [7].

1.1. Problem Formulation

Consider the general minimization problem

$$\hat{f} = \arg \min E(f) = \arg \min E_{\text{smooth}}(f) + E_{\text{data}}(f) , \quad (1)$$

where f is a discrete function from a set $\mathcal{P}$ into the continuous vector space $\mathbb{R}^n$. The operators E_{smooth} and E_{data} measure the deviation of f from a smoothness assumption and from the measured data, respectively.

In practice and due to finite machine accuracy, continuous vector spaces on a computer become bounded and quantized, e.g., for color images we consider functions $f : \mathcal{P} \rightarrow \{0, \dots, 255\}^3$. In many computer vision problems we face however much more quantization levels than 256, e.g. in depth or disparity estimation.¹ Hence, our problem formulation is changed to find pseudo-continuous f with machine accuracy.

Concerning the data term, we consider a point-wise non-linear penalty for deviation from the measurements (cf. [1]):

$$E_{\text{data}}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) . \quad (2)$$

For the smoothness term, we consider pairwise interactions between pixels (in contrast to e.g. [12], where even triplets of pixels are considered):

$$E_{\text{smooth}}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q) , \quad (3)$$

where $\mathcal{N}$ denotes the set of interacting pixels – typically the neighborhood structure.

¹Although many successful methods according to taxonomy from [13] use quantized disparity maps, we believe that the authors of those methods solve the wrong problem: depth as such is continuous and so is the disparity.

Figure 1. Synthetic example: linear gradient with noise (Gaussian and Salt & Pepper) is labeled with 8 labels using the swap algorithm.

1.2. Graph Cut Methods

According to [12], *binary* problems of type (1), i.e., $f : \mathcal{P} \rightarrow \{0, 1\}$, can be solved exactly by graph cut methods. The global minimum of (1) for binary f is obtained by the minimum cut of an appropriate graph. The method to construct the appropriate graph for (1) is explained in [12] and will not be repeated here. A typical fast implementation for finding the minimum cut is given by maximum flow algorithms using the push-relabel technique [3]. In the experiments described further below we use the mex-implementation from [4].

A generalization of the non-convex binary problem to multiple labels is NP hard and only approximative solutions can be obtained by algorithms with tractable computational effort, e.g. the swap and the expansion algorithm [1]. The approximation character of these algorithms is no problem in practice as the solutions are typically 'good' [1]. However, having discrete labels is a strong limitation per se.

Assume that the function f to be estimated is linearly increasing. This could be, for instance, a linearly increasing disparity or the velocity of a quadratically moving point, i.e., we assume a model and the parameter within the model changes linearly. Similarly to computing a histogram with rectangular bins, using discrete labels leads to quantization errors, since we lose the accuracy within the bin / label range, see Fig. 1.

In many practical problems of computer vision, as for instance disparity estimation, we would like to have machine accuracy representation for the estimate. Therefore, graph cut has to be extended to work on a continuous domain. Exactly this is the purpose of this paper: to propose a continuous extension for graph cut. Note that such a method can only be of approximative character as the NP hardness of the non-convex multiple label problem implies NP hardness of the non-convex continuous problem.

2. Methods

The proposed method is based on three concepts from the literature: binary graph cut (see above), channel smoothing [6], and normalized convolution [11]. Whereas the latter is a well known concept which is explained elsewhere in great detail, channel smoothing is a not so well known technique which is reviewed below.

2.1. Channel Representations

The channel representation is a high-dimensional, biologically motivated information and signal representation [10, 15]. It is based on the idea of placing local functions, the *channels*, in the (co-)domain² and to project the data onto the channels. It resembles somehow vector quantization of the co-domain, but without the drawback of losing accuracy, as sub-quantization information is compensated in the channel representation by knowledge about the algebraic relation between the channels.

The projections onto the channels result in tuples of numbers which - although often written as vectors - do not form a vector space in the strict sense. In particular the value zero (in each component) has a special meaning, *no information*, and need not be stored in the memory.

Formally, the channel representation is obtained from a finite set of *channel projection operators* F_n . These are applied to the function f in a point-wise way to calculate the *channel values* c_n :

$$c_n = F_n(f) \quad n = 1, \dots, N. \quad (4)$$

Each function value f_p is mapped to a vector $\mathbf{c} = (c_{1,p}, \dots, c_{N,p})$, the *channel vector*. This process is called *channel encoding*. For instance, if f is a grey-value image, channel encoding leads to a stack of images, c.f. Fig. 2, left.

The projection operators can be of various form, e.g., $\cos^2$ functions, B-splines, or Gaussian functions [8]. In what follows, we will apply quadratic B-spline basis functions B_2 . The B-spline of order k is a piece-wise defined polynomial of order k , defined as

$$B_0(f) = \Pi(f) \quad (5)$$

$$B_k(f) = \frac{k+2f+1}{2k} B_{k-1}(f + \frac{1}{2}) + \frac{k-2f+1}{2k} B_{k-1}(f - \frac{1}{2}) \quad (6)$$

where $\Pi(f)$ is the rectangle function [2], pg. 52. Assuming that f is scaled such that $f \in [1.5; N - 0.5]$, (4) becomes

$$c_n = B_2(f - n) \quad n = 1, \dots, N. \quad (7)$$

²Actually it is possible to project onto the joint input-output space, but in context of channel smoothing it is more reasonable to just consider co-domain projection.

Figure 2. The simple synthetic example from Fig. 1 smoothed by channel smoothing, using a gaussian filter with $\sigma = 10$. No quantization effects are visible but note the rounding of the corner. On the top right: robust error of the decoding.

Decoding of channel representations is discussed in detail in [8]. A *channel decoding* can be considered as a functional inverse to the encoding (4). Different decoding methods show different behavior in terms of computational complexity and quantization errors. In what follows, we will apply the approximately quantization-free decoding scheme proposed in [6], which performs an implicit B-spline interpolation resulting in an efficient semi-analytic decoding method. The method is based on the local linear decoding

$$\begin{aligned}\hat{f} &\approx \frac{\sum_{n=n_0-1}^{n_0+1} n c_n}{\sum_{n=n_0-1}^{n_0+1} c_n} \\ &= \frac{c_{n_0+1} - c_{n_0-1}}{c_{n_0-1} + c_{n_0} + c_{n_0+1}} + n_0, \quad (8)\end{aligned}$$

for n_0 being the closest integer to f . The index n_0 is chosen by minimizing the residual error of the reconstruction, see [6]. For the case of grey-value images, channel decoding condenses a stack of images to a reconstruction image and a residual image, c.f. Fig. 2, right.

2.2. Channel Smoothing and Robust Smoothing

It has been shown that summing B-spline channel vectors of samples from a stochastic variable ξ results in a sampled kernel density estimate of the underlying distribution $d(\xi)$ [6]:

$$\langle \mathbf{c} \rangle_\xi = \langle [F_n(\xi)] \rangle_\xi = (B_2 * d)(n), \quad (9)$$

where $\langle \cdot \rangle_\xi$ denotes the expectation value with respect to variable ξ and $*$ denotes the convolution operator. The global maximum of d is the most probable value for ξ and for locally symmetric distributions, it is equivalent to the maximum of $B_2 * d$. The latter is approximately extracted from the channel vector $\mathbf{c}$ using channel decoding.

In [6] it has been shown that smoothing of B-spline channels approximates the minimization of the robust error norm

$$\rho(f) = -B_3(f + \frac{1}{2}) - B_3(f - \frac{1}{2}) + 2B_3(\frac{1}{2}), \quad (10)$$

such that

$$\hat{f}_p \approx \arg \min_f \langle \rho(F_p - f_p) \rangle_{F_p} \quad (11)$$

for a particular pixel position p and the stochastic variable F_p . The reconstruction error (see e.g. Fig. 2, right) is the residual of this robust minimization.

Typically we do not have several instances of the same data (e.g. image). Hence, we have to replace the expectation value in (9), (11) with a local spatial averaging, for instance with a gaussian kernel g_σ . This results in the channel smoothing algorithm, see Alg. 1.

Algorithm 1 Channel smoothing algorithm.

Require: $f \in [1.5; N - 0.5]$

- 1: **for** $n = 1$ to N **do**
 - 2: $c_n \leftarrow B_2(f - n)$
 - 3: $c_n \leftarrow g_\sigma * c_n$
 - 4: **end for**
 - 5: $[\hat{f}, \hat{\rho}] \leftarrow \text{decode}(\mathbf{c})$
-

Channel smoothing works well if we process data that follows a spatially constant distribution, see Fig. 2 and at straight edges. In the latter case the distribution changes, but the maximum remains the same due to the balanced support on both sides of the edge (although the predicted error increases). This is different at the corner, where the inner part has smaller support. At some point, the maximum changes to the background, and thus the corner becomes rounded, cf. Fig. 2, center image.

2.3. Channel Decoding using Graph-Cut

The rounding of corners can only be avoided if the ergodicity assumption is restricted to the domain where the respective channels are active. This means in practice: channel values should be averaged on bounded domains. Before looking into the issue of determining the active region of a channel, we have to modify the averaging in order to apply it to a bounded domain. Filtering of (uncertain) data f from a bounded domain is well modeled in terms of normalized convolution [11] of 0th order (normalized averaging):

$$\hat{f}_p = \frac{(a * (bf))_p}{(a * b)_p} = \frac{\sum_q a_{p-q} b_q f_q}{\sum_q a_{p-q} b_q}, \quad (12)$$

where a denotes the applicability function, i.e., a suitable averaging kernel (typically a Gaussian function) and b_p is the certainty function, which describes the bounded domain Ω :

$$b_p = \begin{cases} 1 & p \in \Omega \\ 0 & p \notin \Omega \end{cases}. \quad (13)$$

The co-domain of normalized convolution is however unbounded, and therefore, we cannot apply it directly to the case of channels c_n that are active in a bounded region. To remain within the active region of each channel, we mask the result from normalized convolution to the *same bounded domain* of each channel by means of the certainty function $b_{n,p}$:

$$\hat{c}_{n,p} = b_{n,p} \frac{(a * (b_n c_n))_p}{(a * b_n)_p} = b_{n,p} \frac{\sum_q a_{p-q} b_{n,q} c_{n,q}}{\sum_q a_{p-q} b_{n,q}}. \quad (14)$$

Note that the same kind of bounded domain averaging can be used to modify the multi-label graph-cut results: b_n

is replaced with the label function $l : \mathcal{P} \rightarrow \{1, \dots, L\}$ and c_n with the signal f . Artifacts caused by the quantization to the discrete label set are reduced in this way, but not entirely removed. This method is referred to as *modified multi-label graph-cut* below and is summarized in Alg. 2.

Algorithm 2 Modified multi-label graph-cut algorithm.

```

1:  $l \leftarrow \text{multi\_label\_graph\_cut}(f, \mathcal{N}, \lambda, L)$ 
2: for  $i = 1$  to  $L$  do
3:    $m \leftarrow (l = i)$ 
4:    $\hat{f} \leftarrow m \frac{g_{\sigma^*}(mf)}{g_{\sigma^*}m}$ 
5: end for

```

What remains to be considered is the estimation of the active region, or equivalently, the certainty function for each channel. We have to find those regions where the image was sufficiently stationary in order to produce *similar* channel vectors c_p . Similar channel vectors have the same active components, where we classify the activity by a simple threshold θ . As stationarity does not make sense without spatial context, we require the active region to be as connected as possible. This is where graph-cut comes in.

For each channel n we formulate the following objective function:

$$E(a_n) = \sum_{p \in \mathcal{P}} b_{n,p}(\theta - c_{n,p}) + \lambda \sum_{\{p,q\} \in \mathcal{N}} |b_{n,p} - b_{n,q}|, \quad (15)$$

where we use the following parameters throughout the remainder of this paper:

- $\mathcal{N}$ is the four-neighborhood
- $\lambda = 0.3$ is the penalty for discontinuities in b_n
- $\theta = \frac{1}{16}$ is the threshold for active channels.

All parameters were chosen ad-hoc (as is the width of the Gaussian filter $\sigma = 10$ and the number of channels $N = 10$), but remain constant throughout the paper. The more systematic choice of parameters based on signal statistics is out of the scope of this paper. The interested reader is referred to [5, 9] for estimation of the channel averaging filter and the number of channels. The threshold θ can be derived from classical decision theory (see e.g. [16], Chapt. 3) and depends also on the number of channels. The meta parameter λ depends on the neighborhood structure and the signal statistics. It should be at least one third of the maximum channel value (minus θ) to fill in one-pixel gaps in a contour with four-neighborhood. Too large values will remove structural details from the active region. For quadratic B-spline channels and $\theta = \frac{1}{16}$ this happens for $\lambda \geq \frac{11}{32}$.

We summarize the graph-cut channel smoothing in Alg. 3. The synthetic example from Fig. 1 shows that graph-cut channel smoothing does not suffer from the drawback

of multi-label graph-cut (coarse quantization), nor does it suffer from rounding of corners as pure channel smoothing does, see Fig. 3. The computational complexity of the proposed method is somewhat higher than that of pure channel smoothing and it is dominated by the N binary graph-cut computations. However, it is by at least two orders of magnitude lower than the complexity for multi-label graph-cut (original variant and gcm), as no iterations are required and as no pair-wise cuts have to be computed.

Algorithm 3 Graph-cut channel smoothing algorithm.

Require: $f \in [1.5; N - 0.5]$

```

1: for  $n = 1$  to  $N$  do
2:    $c_n \leftarrow B_2(f - n)$ 
3:    $b_n \leftarrow \text{binary\_graph\_cut}(c_n, \mathcal{N}, \lambda, \theta)$ 
4:    $c_n \leftarrow b_n \frac{g_{\sigma^*}(b_n c_n)}{g_{\sigma^*} b_n}$ 
5: end for
6:  $[\hat{f}, \hat{\rho}] \leftarrow \text{decode}(c)$ 

```

2.4. The Combined Objective Function

Apparently, the whole process of *graph-cut channel smoothing* presented in the previous section contains two objective functions: one point-wise robust error minimization (11) and one channel-wise region labeling (15). In order to show that the proposed method as a whole solves approximately a continuous labeling problem according to (1), one has to reformulate the whole algorithm as a single objective function.

At first glance, the smoothness constraint in (15) corresponds to a Potts model [1] by interpreting the channel indices as labels. This is however not correct for two reasons: a) We do not choose a single index out of N , but several of them, where the cardinality is determined by the distribution of the data. b) We are interested in estimating a value in the original continuous value domain of f , i.e., we have to consider the smoothness term in terms of deviations in f .

The issue a) is not crucial as we later select the index with the minimum robust residual, see below. Hence, we can equivalently consider only the one with minimum data error. b) As f is channel encoded using a continuous function (B-spline) that is monotonically increasing with increasing deviation in f , the smoothness term is a piece-wise constant function of differences in f :

$$V_{p,q}(f_p, f_q) = \begin{cases} 0 & |f_p - f_q| < \theta_f \\ \lambda & |f_p - f_q| \geq \theta_f \end{cases}. \quad (16)$$

The relation between θ in (15) and θ_f above is given by the channel basis function, i.e., the quadratic B-spline in this case. We hence obtain $\theta_f = s \frac{6-\sqrt{2}}{4}$, where s is the spacing between channels.

Figure 3. Simple synthetic example from Fig. 1 respectively Fig. 2 smoothed by graph-cut channel smoothing, using a gaussian filter with $\sigma = 10$. Top right: result from graph-cut channel smoothing. Below: actual absolute error compared to noise-free gradient image and reconstruction error. Note the absence of rounding of the corner.

The data term which corresponds to channel smoothing with bounded support normalized convolution is given by

$$D_p(f_p) = \sum_{q \in \mathcal{A}(f_p)} a_{q-p} \rho(f_q - f_p) , \quad (17)$$

where $\mathcal{A}(f_p)$ is the set of all pixels q where $E_{\text{smooth}}(f_q - f_p) = 0$. The applicability function a_q is a normalized Gaussian function used for channel averaging. If we accept the assumption that the image generating process is ergodic in the region $\mathcal{A}(f_p)$, we can replace the weighted averaging over this region with the expectation value for the stochastic variable F_p , and hence

$$D_p(f_p) = \langle \rho(F_p - f_p) \rangle_{F_p} . \quad (18)$$

Finally, we formulate the following *conjecture*³: Graph-cut channel smoothing as described in the previous sections establishes an algorithm for finding an approximation to a minimum of the objective function

$$E(f) = \sum_{p \in \mathcal{P}} \langle \rho(F_p - f_p) \rangle_{F_p} + \sum_{\{p,q\} \in \mathcal{N}: |f_q - f_p| \geq \theta_f} \lambda . \quad (19)$$

The approximation of the minimum is in a sense of approximating the expectation value. The minimum approximation of the first term is global, i.e., under mild assumption the method approximates the global minimum [6]. However, since we do the minimization in two steps based on quantized representations, we cannot guarantee that we exclude the global solution during the first step due to quantization errors.

3. Experiment

In the experiment we used the ground truth data from the disparity evaluation described in [13, 14], added Gaussian noise ($\sigma = 10\%$ of total range) and salt & pepper (5% of all pixels) to the disparity maps. We applied multi-label (10 labels) graph-cut (gcut), modified multi-label graph-cut (gcm), pure channel smoothing (bscs), and graph-cut channel smoothing (gccs) to the disparity maps. The resulting maps were then evaluated through the web-site, the results are collected in Tab. 1. For comparison, we even included the noisy disparity maps in the table (noise). Fig. 4 shows the different results for the cases of the 'Cones' and 'Teddy' experiments.

The results clearly show that graph-cut channel smoothing improves results compared to multi-label graph-cut and pure channel smoothing. Only modified multi-label graph-cut is a few times superior to graph-cut channel smoothing, in particular near to discontinuities and occlusions. All experiments were performed with the parameters reported earlier.

³We call this a conjecture as we do not prove it formally by computing approximation error bounds.

Cones

Teddy

Figure 4. Example images from the disparity evaluation, 'Cones' and 'Teddy'. Row-wise, from top left to bottom right: ground truth disparity with added Gaussian noise and salt & pepper, result from multi-label graph-cut, from modified multi-label graph-cut, and from channel smoothing with binary graph-cut.

4. Conclusion

We have presented two algorithms to find approximate solutions to minimization problems that are typically addressed by multiple-label graph-cut, but which avoid a hard quantization of the value domain and thus has a much broader spectrum of possible applications. One algorithm

Table 1. Results of disparity evaluation from [13, 14]. L: Tsukuba, V: Venus, T: Teddy, C: Cones. non: non-occluded, all: includes half-occluded regions, dis: near discontinuities and occlusions.

alg.	L/non	L/all	L/dis	V/non	V/all	V/dis	T/non	T/all	T/dis	C/non	C/all	C/dis
noise	53.9	53.9	54.0	74.9	74.9	74.6	86.7	86.7	86.9	86.8	86.8	86.9
gcut	1.51	1.57	7.07	55.7	56.0	62.4	79.0	76.0	75.6	59.5	60.6	61.9
gcm	0.90	1.06	4.70	3.00	3.15	5.92	23.3	23.4	41.0	25.0	26.3	34.7
bscs	6.70	7.30	21.9	3.00	3.64	41.8	15.3	16.3	47.1	17.0	20.0	46.8
gccs	2.11	2.36	10.3	2.10	2.72	29.4	12.6	13.5	38.6	14.5	17.0	40.8

is a modified multi-label graph-cut method, the other one is based on channel smoothing. We have explained the effects of the algorithms in detail, we have illustrated them with a synthetic experiment, and formulated a combined energy function for the channel-based method. We have also evaluated the algorithms quantitatively against ordinary multiple-label graph-cut and pure channels smoothing. The proposed methods show results superior to ordinary channel smoothing and multi-label graph-cut for a mild increase of complexity compared to the ordinary algorithms. We consider graph-cut channel smoothing as the best approach as it is two orders of magnitude faster than multi-label graph-cut and slightly superior to modified multi-label graph-cut. What remains to do in future work is to formalize the proof of the stated conjecture concerning the combined objective function.

References

- [1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [2] R. N. Bracewell. *The Fourier transform and its applications*. McGraw Hill, 1986.
- [3] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.
- [4] N. H. A. Deschamps. Better foreground segmentation through graph cuts. Technical report, <http://arxiv.org/abs/cs.CV/0401017>, 2004.
- [5] M. Felsberg. Wiener channel smoothing: Robust Wiener filtering of images. In *DAGM 2005*, volume 3663 of *LNCS*, pages 468–475. Springer, 2005.
- [6] M. Felsberg, P.-E. Forssén, and H. Scharf. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, 2006.
- [7] M. Felsberg and G. Granlund. P-channels: Robust multi-variate m-estimation of large datasets. In *International Conference on Pattern Recognition*, Hong Kong, August 2006.
- [8] P.-E. Forssén. *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University, Sweden, 2004.
- [9] W. Förstner. Image preprocessing for feature extraction in digital intensity, color and range images. In *Proceedings of the International Summer School on Data Analysis and the Statistical Foundation of Geomatics*, Lecture Notes on Earth Sciences. Springer, 1998.
- [10] G. H. Granlund. An Associative Perception-Action Structure Using a Localized Space Variant Information Representation. In *Proceedings of Algebraic Frames for the Perception-Action Cycle (AFPAC)*, Kiel, Germany, September 2000.
- [11] H. Knutsson and C.-F. Westin. Normalized convolution: Technique for filtering incomplete and uncertain data. In *Proceedings of the 8th Scandinavian Conference on Image Analysis*, volume II, pages 997–1006, 1993.
- [12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
- [13] D. Scharstein and R. Szelisky. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.
- [14] D. Scharstein and R. Szelisky. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, volume 1, pages 195–202, 2003.
- [15] H. P. Snippe and J. J. Koenderink. Discrimination thresholds for channel-coded systems. *Biological Cybernetics*, 66:543–551, 1992.
- [16] C. W. Therrien. *Decision, estimation, and classification: an introduction into pattern recognition and related topics*. John Wiley & Sons, Inc., 1989.