Channel Coding for Joint Colour and Depth Segmentation

Marcus Wallenberg[1], Michael Felsberg[1], Per-Erik Forssén[1], and Babette Dellen[2]

[1] Linköping University, SE-581 83 Linköping, Sweden,
{wallenberg,mfe,perfo}@isy.liu.se,
[2] Institut de Robotica i Informatica Industrial (CSIC-UPC)
Llorens i Artigas 4-6, 08028 Barcelona, Spain

Abstract. Segmentation is an important preprocessing step in many applications. Compared to colour segmentation, fusion of colour and depth greatly improves the segmentation result. Such a fusion is easy to do by stacking measurements in different value dimensions, but there are better ways. In this paper we perform fusion using the channel representation, and demonstrate how a state-of-the-art segmentation algorithm can be modified to use channel values as inputs. We evaluate segmentation results on data collected using the Microsoft Kinect peripheral for Xbox 360, using the superparamagnetic clustering algorithm. Our experiments show that depth gradients are more useful than depth values for segmentation, and that channel coding both colour and depth gradients makes tuned parameter settings generalise better to novel images.

1 Introduction

Segmentation of a colour image into semantically meaningful regions is one of the oldest problems in computer vision. Purely colour-based segmentation is often problematic, due to colour changes on the surfaces of textured objects. It is thus often argued that without auxiliary information (such as prior knowledge obtained e.g. using object appearance learning) bottom up, image based segmentation is an ill-posed problem [15, 11].

In contrast to colour regions, homogeneous regions obtained from depth segmentation are more likely to correspond to what we intuitively perceive as objects. The reason for this is that we categorise objects, mainly according to what actions we can perform on them [14, 11]. An entity that is defined in 3D is more likely to be acted upon separately, than one that is defined only by colour. By fusing colour and depth we can however obtain an even better result, and here we investigate how to do so.

Our intended application is segmentation of individual leaves on growing plants, and we use data from the recently introduced Microsoft Kinect sensor[3]. As an operational problem definition, we make use of a set of hand-labelled images, where individual leaves have been assigned different labels.

[3] http://www.xbox.com/Kinect

1.1 Related Work

Much work on fusion of colour and depth has been done over the years. Such work has either used custom made sensors, as e.g. in [4], or more recently, time-of-flight sensors [7, 3, 5]. Another large body of similar work is stereo rig segmentation [1, 20]. Stereo rig research is however of a different nature, as the input is two RGB images, and thus best results are obtained when jointly estimating a segmentation and a depth map [20].

We use depth from structured light (the Kinect output), which gives us *quasi-dense* depth maps; values almost everywhere, but with thin missing-data shadows near occlusion boundaries.

Currently there exists no standard evaluation set for RGB+depth segmentation, instead only qualitative examples of success are shown, see e.g. [4, 1, 7, 3]. In [20, 5] only the depth map quality is evaluated. In colour image segmentation, good evaluation datasets exist, see e.g. [16], and these are of great use when selecting algorithms for particular applications. We have assembled a dataset with hand-labelled ground truth, and we use it to thoroughly verify the relative contributions of colour and depth, as well as the improvement offered by channel coding.

Our application is inspection and measurement of growing plants. As the scene is static, we cannot exploit either background modelling [7] or tracking [3]. Furthermore, purely colour-based segmentation is particularly brittle here, due to small reflectance variations, shadows, and in particular occlusions [6]. Segmentation for plant model registration is considered to be a hard problem that requires manual interaction even if colour and depth information is used [17].

We improve fusion of colour and spatial derivatives of depth, by using the channel representation [12]. By feeding the fused channel vectors to a state-of-the-art colour segmentation algorithm [2] we obtain a method that once tuned, will generalise well to new data.

2 Methods and Materials

2.1 The Microsoft Kinect

The Microsoft Kinect[1] is a peripheral device for the Xbox 360. It is used to obtain dense depth estimates using a structured light pattern. The device contains a colour camera, a *near-infrared* (NIR) camera and a laser projector, offset by a narrow baseline, see Fig. 1, **a**, **b**.

A structured light pattern is projected onto the scene, using a laser projector with a characteristic wavelength of 830 nm[4]. The structured light pattern is designed to have a negligible auto-correlation, and is imaged by the NIR camera. The displacement of the NIR camera relative to the laser projector allows the distance to objects in the scene to be computed using triangulation [18]. The device is capable of outputting RGB, NIR and depth images with 640×480

[4] `http://openkinect.org/wiki`

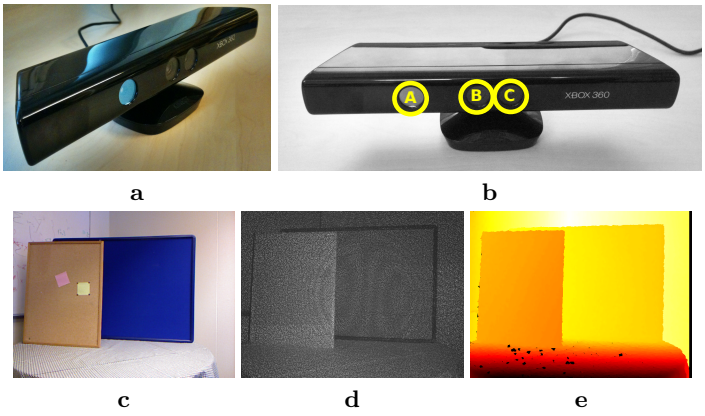

Fig. 1. The Kinect device: a, b (A) – laser projector, (B) – colour camera, (C) – NIR camera; **Images from the Kinect: c** – RGB image from colour camera; **d** – light pattern as imaged by NIR camera; **e** – resulting depth map.

pixels at 30 frames per second (Fig. 1, **c**–**e**). Open source drivers in the form of the `libfreenect`[5] library are available from the OpenKinect[6] community and can be used to interface with the Kinect device. Approximate formulae for converting the Kinect depth map to metric distances are also available[2].

We use the `libfreenect`[3] library to control the Kinect, and receive the colour and depth video streams. The two streams need to be aligned, since the position, orientation and *field of view* (FoV) of the cameras are different. We do this by first estimating the intrinsic camera parameters of the two cameras, using the widely used OpenCV[7] implementation of [22]. We then find the relative orientation and translation between the cameras, by minimising the transfer error in the image plane of the colour image, using manually selected corresponding points in the colour and NIR images. We do this using the non-linear least squares solver `lsqnonlin` in MATLAB.

Note that, as the Kinect cameras are rigidly mounted, the calibration described here only has to be performed once for each unit. In the following, we thus consider the RGB image $\mathbf{f}(x, y)$, and the depth map $h(x, y)$, transferred to the RGB camera as the input.

2.2 Fused Feature Vectors

The depth image $h(x, y)$ delivered by the Kinect is a) quantised and b) the quantisation levels are proportional to the absolute depth. This implies that segmentation based on the depth becomes more difficult if the respective part of

[5] `https://github.com/OpenKinect/libfreenect`
[6] `http://openkinect.org`
[7] `http://opencv.willowgarage.com`

the scene is located further away from the camera. Due to the constant spatial accuracy in the NIR camera, this behaviour is sensible. However, for leaves that touch each other and which are at distances of approximately one meter, the segmentation will bleed out between neighbouring leaves if the segmentation is based on regularising the gradient of the depth image:

$$E_{\text{smooth}} = \rho(|\nabla h(x,y)|^2) \ , \tag{1}$$

where E_{smooth} is the regularising term and $\rho()$ is a monotonic function.

Touching leaves are unlikely to have identical surface normals. Therefore, we choose to regularise the differences in the gradients of the depth map instead:

$$E_{\text{smooth}} = \rho(|\nabla h_x(x,y)|^2 + |\nabla h_y(x,y)|^2) \ , \tag{2}$$

where $h_x(x,y) = \frac{\partial}{\partial x} h(x,y)$ and $h_y(x,y) = \frac{\partial}{\partial y} h(x,y)$.

Thus, we have three requirements for assigning image points to the same segment: similar colour ($\mathbf{f}(x,y)$), similar x-derivative of the depth image ($h_x(x,y)$), and similar y-derivative of the depth image ($h_y(x,y)$). In the ideal case, the feature vector used for segmentation, called $\mathbf{g}(x,y)$ in what follows, should represent $\mathbf{f}(x,y)$, $h_x(x,y)$, and $h_y(x,y)$. In the experiments below, five different variants of the feature vector will be used:

$$\mathbf{g}(x,y) = \mathbf{f}(x,y) \tag{3}$$

$$\mathbf{g}(x,y) = h(x,y) \tag{4}$$

$$\mathbf{g}(x,y) = \begin{bmatrix} h_x(x,y) \\ h_y(x,y) \\ \sqrt{h_x(x,y)^2 + h_y(x,y)^2} \end{bmatrix} \tag{5}$$

$$\mathbf{g}(x,y) = \begin{bmatrix} (1-\lambda)\mathbf{w}(\mathbf{f}(x,y); b_1) \\ \lambda \mathbf{w}(h_x(x,y); b_2) \\ \lambda \mathbf{w}(h_y(x,y); b_2) \end{bmatrix} \tag{6}$$

$$\mathbf{g}(x,y) = \begin{bmatrix} (1-\lambda)\mathbf{f}(x,y) \\ \lambda h_x(x,y) \\ \lambda h_y(x,y) \\ \lambda\sqrt{h_x(x,y)^2 + h_y(x,y)^2} \end{bmatrix} \tag{7}$$

where $\lambda > 0$ is a weight factor between colour and depth and $\mathbf{w}$ is the channel vector computed using the basis function b_j, cf. sect. 2.4. The respective feature vector $\mathbf{g}(x,y)$ is then spatially clustered using superparamagnetic clustering.

2.3 Superparamagnetic Clustering

In the image, each pixel is characterised by a feature vector $\mathbf{g}(x,y)$. Our goal is to to group the image pixels into spatially connected areas of similar feature values. This defines a pixel labelling problem, where a label has to be assigned to every pixel i, which we call l_i. To find this label configuration, we use the method of superparamagnetic clustering of data [2]. In this method, each pixel i is assigned a spin variable σ_i (not to be confused with the label l_i), which can take q different states. The spins interact with each other such that spins having a similar feature value have the tendency to align. Here, we only consider nearest neighbour coupling, i.e., two pixels are i and j with coordinates (x_i, y_i) and (x_j, y_j) are only interacting if $|(x_i - x_j)| \leq 1$ and $|(y_i - y_j)| \leq 1$.

The spin states configuration is then determined by a Potts energy function

$$E = -\sum_{\langle ij \rangle} J_{ij} \delta(\sigma_i, \sigma_j) \ , \tag{8}$$

with $J_{ij} = 1 - \triangle / \bar{\triangle}$ and $\triangle_{ij} = |\mathbf{g}_i - \mathbf{g}_j|$, where $\mathbf{g}_i$ and $\mathbf{g}_j$ are the feature vectors of the pixels i and j, respectively. The mean distance $\bar{\triangle}$ is obtained by averaging over all bonds and scaling with a factor h. The Kronecker δ function is defined as $\delta(a, b) = 1$ if $a = b$ and zero otherwise.

The model is a statistical model, so the probability $P(S)$ of a spin configuration S is determined by the Boltzmann distribution through $P(S) \propto \exp(-E/T)$, where T is the temperature of the system. This implies that the energy is the logarithm of the probability of the spin configuration and can thus also be viewed as the log likelihood of the posterior distribution of a Markov Random Field [10].

The grouping problem is then solved by finding clusters of correlated spins in the low temperature equilibrium states of the energy function E, using a sigmoid of E as the link strength. The total number M of segments is then determined by counting the computed segments. It is usually different from the total number q of spin states, which is a parameter of the algorithm (here $q = 30$).

We solve this task by implementing a clustering algorithm. In a first step, "satisfied" bonds, i.e. bonds connecting pixels of identical spins $\sigma_i = \sigma_j$, are identified. Then, in a second step, the satisfied bonds are "frozen" with some probability P_{ij}. Pixels connected by frozen bonds define a cluster, which are updated by assigning the same value to all spins inside a cluster [19]. In the method of superparamagnetic clustering proposed by Blatt *et al.* [2] this is done independently for each cluster. The algorithm is controlled by the "temperature" parameter, and has been shown to deliver robust results over a large temperature range. After 100 iterations, clusters are used to define segments with labels l_i. As a consequence, two spins which are in the same spin state can carry different segment labels. This allows testing new spin combinations in the next iteration, while stabilising segments having similar feature values.

2.4 Channel Coding

Adding depth derivatives, $h_x(x, y)$, $h_y(x, y)$, and colour, $\mathbf{f}(x, y)$, as different components of a vector space (7) is not sensible due the different respective physical units. Instead, we will use smooth basis functions to generate probabilistic representations of colour and depth derivatives and combine those (6). The generated representations, called *channel representations* [12], are a special case of soft histograms, with the additional property that modes of the underlying density can be extracted with sub-bin accuracy [9].

Channel representations are also known as population codes [21]. They differ from GMMs and Parzen window (or kernel density) estimators, because positions of the basis functions are spread regularly across the domain. This has the advantage that signal processing methods can be used for manipulation, see e.g. [13] for the use of basis functions in the colour channels.

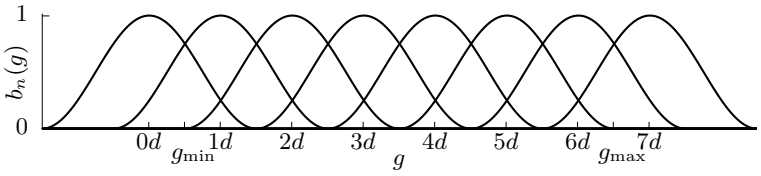

Fig. 2. Illustration of basis functions for $N = 8$. The basis functions are spaced with distance d and have a width of $3d$. The encoded values may lie between $g_{\min}$ and $g_{\max}$.

Given a feature component g, the basis functions are located on a grid with spacing d. The used kernel function $b(g)$ are compact and overlapping. Throughout this paper they have a support of size $3d$, see Fig. 2. In the remainder of this paper, $\cos^2$ kernel functions [12] are used:

$$b(g) \triangleq \frac{2}{3d} \begin{cases} \cos^2(\frac{\pi g}{3d}) & |g| < \frac{3d}{2} \\ 0 & |g| \geq \frac{3d}{2} \end{cases} . \tag{9}$$

The range of g together with d determine the number of basis functions $N = (g_{\max} - g_{\min})/d + 2$. The grid index is $n \in \{0 \dots N-1\}$. Using (9), we obtain the *channel vector* $\mathbf{w} = [w_0,\ w_1,\ \dots,\ w_{N-1}]^T$ from g using:

$$w_n(g; b) = b(g - nd - d/2 - g_{\min}) \ . \tag{10}$$

Usually, several feature components from a local neighbourhood are pooled in each vector by local averaging of channel vectors.

The distance of channel vectors behaves like a sigmoid function of the corresponding feature distance: Large distances become saturated [9]. Statistically independent channel vectors can be concatenated, as is done in (6), and still result in sensible distance measures. The RGB vector might be interpreted as a channel vector of the spectral density with length $N = 3$ and the colour matching function as basis functions. Applying spatial averaging, the resolution of the channel vector is increased [8]. Channel encoding is denoted $\mathbf{w}(\mathbf{f}(x, y); b_1)$ in (6).

3 Experiments

3.1 Data sets

Evaluation data consists of six pairs of images, each consisting of a colour image (640×480 pixels in 8-bit RGB), and an aligned depth map of equal resolution. An example of such an image pair is shown in figure 3. These image pairs (henceforth denoted *plant1* to *plant6*) were chosen to illustrate the challenges faced when performing segmentation based on colour and depth. The objective is to segment leaves on the plant from the background, and from each other. This causes problems when using colour-based segmentation due to the similarity in colour between one leaf and another. The complex structure with many

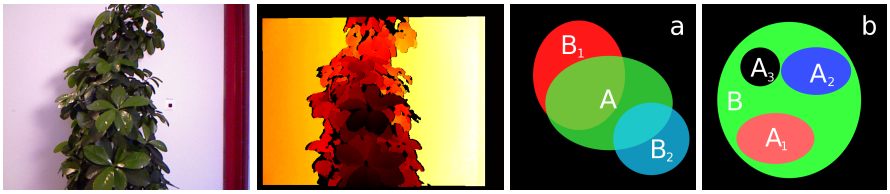

Fig. 3. Examples of evaluation images and segmentation evaluation. From left to right: Colour image and depth map for the *plant3* images. Illustrations (a) and (b) used to describe the segmentation evaluation procedure (see section 3.2).

occlusion boundaries where leaves overlap also causes problems for depth-based segmentation, as do the connections of leaves to one another.

3.2 Performance Evaluation

Performance evaluation was carried out using manually segmented ground-truth images. In these images, regions of the kind we wish to segment were manually separated and labelled. Examples of such images are shown in figure 5, first and third row. In [16], *precision* and *recall* measures are used to evaluate performance. While this is readily applicable to a binary problem, its generalisation to the multi-region segmentation case is not straightforward. We instead propose a *consensus score*, s, with which to score a particular segmentation of an image. The score s is computed as the sum of two terms, where one serves to reward coverage of ground truth segments and penalise over-segmentation, and the other serves to penalise under-segmentation and merging of ground-truth regions.

When calculating $s_{\mathbf{Y}}(X)$, X is the segment for which the score is calculated, and $\mathbf{Y} = \{Y_j\}_1^J$ are overlapping segments in the result being compared to. With regions as in figure 3(a) (with $X = A$ and $\mathbf{Y} = \{B_1, B_2\}$), A corresponds to a ground-truth segment, and B_1 and B_2 correspond to overlapping segmentation results. With $S(R)$ denoting the area of a particular segment, $s_{\mathbf{B}}(A)$ is:

$$s_{\mathbf{B}}(A) = \max_i (S(A \cap B_i) - \sum_{j \neq i} S(A \cap B_j)) \,. \tag{11}$$

For the example in figure 3(a) we get $s_{\mathbf{B}}(A) = S(A \cap B_1) - S(A \cap B_2)$.

When all ground-truth segments in an image have been scored in this way, the roles of ground-truth and segmentation results are reversed. With regions as in figure 3(b), with B corresponding to a segment in the segmentation result, and A_1, A_2 and A_3 corresponding to ground-truth regions, $s_{\mathbf{A}}$ is calculated as in (11), which in this case means $s_{\mathbf{A}}(B) = S(B \cap A_1) - S(B \cap A_2) - S(B \cap A_3)$.

The final consensus score s is then the sum over all K ground-truth regions and all J segments as:

$$s = \frac{\sum_{k=1}^{K} s_{\mathbf{B}}(A_k) + \sum_{j=1}^{J} s_{\mathbf{A}}(B_j)}{2 \sum_{k=1}^{K} S(A_k)} \,, \tag{12}$$

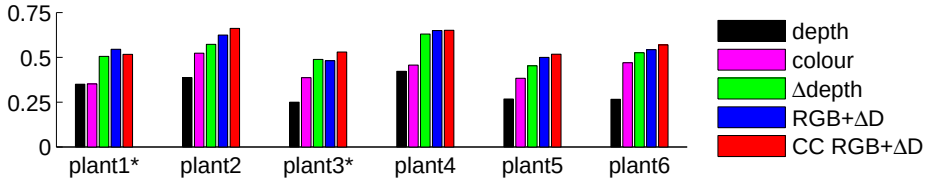

Fig. 4. Consensus scores on the *plant* data set. Images indicated with asterisks were used for parameter tuning of all methods. Note that although RGB+ΔD and channel coded RGB+ΔD have similar results on the tuning images, channel coded RGB+ΔD has a higher score on all evaluation images.

where $\sum_{k=1}^{K} S(A_k)$ is the total area of ground-truth regions in an image (this produces a score in the range $-1 < s \leq 1$). Note that this method differs from [16]. Since we cannot evaluate results in areas not covered by ground truth data, these will not affect the resulting score (12). We also use the entire regions instead of comparing boundaries as our goal is coverage, rather than precise location of boundaries.

3.3 Tested methods

The methods we evaluate all make use of superparamagnetic clustering, as described in section 2.3. The feature vectors used are those described in section 2.2. The depth map gradient was estimated using finite differences. A small amount of low-pass filtering (3×3 Gaussian kernel with $\sigma = 1.5$ px) was applied to each component of the feature vectors before clustering. This serves to reduce noise, and was found to improve the results for all tested methods.

For all methods, the temperature parameter was kept constant at $T = 0.05$. Scaling parameters were tuned by maximising consensus score on the *plant1* and *plant3* sets. For the methods using only colour or depth, the global scaling parameter was tuned individually for each method. In the cases when both depth and colour information was used, the global scale factor was optimised together with the relative weight λ for each method (see section 2.2, eq. (3) to (7)). The number of basis functions in channel coding was $N = 6$ for colour and $N = 7$ for each of $h_x(x, y)$ and $h_y(x, y)$ (resulting in a total of 20 channels).

3.4 Results

Results of the evaluation procedure are shown in figures 4 (consensus scores) and 5 (segmented images). Purely colour- and depth-based segmentation performs worst, as can be expected given the nature of the data. Depth gradient-based segmentation (Δdepth) performs better than either of these two. The concatenation of RGB colour and depth gradient (RGB + ΔD) performs well overall, but seems to show a slight tendency toward overfitting. The channel-coded variant (CC RGB + ΔD) shows similar results on the training data, but generalises better to the other image sets.

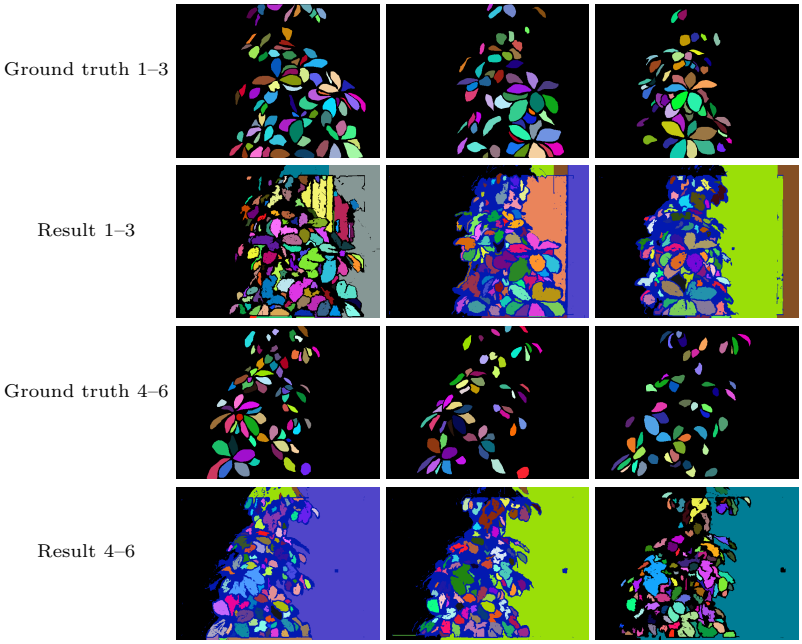

Fig. 5. Segmentation results on the *plant* data set, and corresponding ground truth.

4 Conclusions

We have evaluated a method for joint colour and depth-based segmentation using data gathered with the Kinect. The results show that it is indeed possible to obtain better results by fusing colour and depth, than using either one in isolation. The greater robustness of the channel-based segmentation indicates that this is a suitable approach for fusing these measurement modalities. Our experimental setup with consensus score tuning on two of the image pairs, and evaluation on all pairs also demonstrates that the parameters found by tuning generalise well to new data. Future work will include exploring the use of other colour spaces, as well as other ways to represent the depth maps, before feeding them to the channel encoding procedure.

Acknowledgements

This work was supported by the EC's 7th Framework Programme (FP7/2007-2013), grant agreement 247947 (GARNICS), the Swedish Research Council through a grant for the project *Embodied Visual Object Recognition*, and by Linköping University. B. Dellen also acknowledges support from the Spanish Ministry for Science and Innovation via a Ramon y Cajal fellowship.

References

1. M. Björkman and J.-O. Eklundh. Vision in the real world: Finding, attending and recognizing objects. *Int. J. of Imag. Sys. and Technology*, 5(16):189–209, 2006.
2. M. Blatt, S. Wiseman, and E. Domany. Superparametric clustering of data. *Physical Review Letters*, 76(18), 1996.
3. A. Bleiweiss and M. Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. In *DAGM Dyn3D Workshop*, LNCS 5742:58-69, 2009.
4. Pierre Boulanger. Simultaneous segmentation of range and color images based on bayesian decision theory. In *Computer and Robot Vision (CRV04)*, 2004.
5. B. Dellen, G. Alenya, and C. Torras. Segmenting color images into surface patches by exploiting sparse depth data. In *WACV*, pages 591–598, 2011.
6. A. Walter et al. Dynamics of seedling growth acclimation towards altered light conditions can be quantified via GROWSCREEN. *New Phytol*, 174:447–455, 2007.
7. J. Leens et al. Combining color, depth, and motion for video segmentation. In *ICVS*, volume LNCS 5815, pages 104–113, 2009.
8. M. Felsberg. Incremental computation of feature hierarchies. In *Pattern Recognition 2010, Proceedings of the 32nd DAGM*, 2010.
9. M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE TPAMI*, 28(2):209–222, 2006.
10. Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE TPAMI*, 6:721–741, 1984.
11. G. H. Granlund. Does vision inevitably have to be active? In *SCIA'99*, 1999.
12. G. H. Granlund. An associative perception-action structure using a localized space variant information representation. In *Proceedings AFPAC*, 2000.
13. M. Kass and J. Solomon. Smoothed local histogram filters. In *ACM SIGGRAPH 2010 papers*, pages 100:1–100:10, New York, NY, USA, 2010. ACM.
14. G. Lakoff. *Women, Fire, and Dangerous Things - what categories reveal about the mind*. University of Chicago Press, 1987.
15. B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference*, pages 759–768, 2003.
16. David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using brightness and texture. In *Advances in Neural Information Processing Systems (NIPS02)*. MIT Press, 2002.
17. L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang, and S. B. Kang. Image-based plant modeling. In *ACM SIGGRAPH 2006*, pages 599–604, New York, NY, USA, 2006.
18. A. Shpunt and B. Pesach. Optical pattern projection. US Patent Application Publication, US 2010/0284082 A1, November 2010.
19. R.H. Swendsen and S. Wang. Nonuniversal critical dynamics in monte carlo simulations. *Physical Review Letters*, 76(18):86–88, 1987.
20. Y. Taguchi and et al. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *CVPR08*, 2008.
21. R. S. Zemel, P. Dayan, and A. Pouget. Probabilistic interpretation of population codes. *Neural Computation*, 10(2):403–430, 1998.
22. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 1998.