A Bayesian Approach to Image Understanding: From Images to Virtual Forests *

Terry Caelli, Li Cheng and Qiongyan Fang
Department of Computing Science
University of Alberta, Edmonton, Alberta, Canada
T6G 2E1

Abstract

In this paper we consider the full cycle of Image Understanding (IU): the generation of 3D object hypotheses (inverse model) from images and their projections back onto image data (forward model) in terms of Bayesian inference processes. Each subprocess is framed as a local optimization problem based on a component model and observations. The end result is an IU system that not only provides a symbolic description of scenes but also generates fully 3D versions of the scene being sensed so providing validation criteria for the image annotation.
Keywords: Image Understanding, Bayesian methods, hierarchical hidden Markov random fields, CAD-based Vision, Forestry Photo Interpretation, Stochastic L-systems

1 Introduction

Image Understanding (IU), by definition, is concerned with the interpretation of visualized data in terms of the structures and processes that generated them. These structures and processes can be exclusively 2D, as in documents or handwriting, or, more often than not, 3D, as occurs with human vision in navigation or scene understanding. Over the past twenty years or so 2D IU systems have have focused on statistical and structural pattern recognition technologies, while the 3D systems have more or less focused on Photogrammetry or CAD-based approaches. The result of this has been the rapid development of stochastic models for image processing, feature extraction, Visual Learning and Pattern Recognition in 2D. At the same time, in a parallel community, there has been significant developments in areas such as Photogrammetry, improved CAD-based vision and model matching algorithms (see, for example, Hartley and Zisserman[14]). Only rarely do these two areas come together in the development of fully 3D systems that can be readily trained and accommodate the variabilities that human perception can. In this paper we explore the integration of these two approaches into a fully trainable, adaptive 3D system that exploits the ability of probabilistic models and Bayesian inference methods to perform IU. Of particular importance is the exploration of how to formulate the solution of the system components in terms of common optimization principals.

In addition to the above aims, we also claim that IU is particularly relevant when it can assist or replace humans in performing specific talks - particularly as they occur in a circumscribed, well-defined setting. In this case we have considered the task of photo interpretation as used in the forestry industry where professional photo interpreters are certified and employed on a regular basis as past of the forestry inventory process. Fortunately, in this case, there is a significant amount of expertise and knowledge about what makes for reliable photo interpretation in this context which provides an excellent platform for investigating the proposed approach.

Perhaps the best way of documenting how expert photo interpreters interpret forestry images is to examine what they are taught when being trained. Summarizing Hall[13] the photo interpretation (PI) procedure has the following components:

1. Terrain and forestry (stand and individual tree characteristics) prior knowledge is critical.
2. Depth, stereo information about trees is required.
3. PI involves combining and reasoning about image clues as to how they fit with prior knowledge.
4. Important image clues include: Shadow, Tone(spectra/contrast), Texture (for example, types of canopies), Pattern(distributions of trees over a region), Shape (canopy shape), Size(canopy size), Location and Association - related to terrain knowledge.

*This project was funded by a grant from the National Science and Engineering Research Council of Canada and the Alberta Research Council.

5. PI involves reasoning about what is perceived with what is known: integrating 2D image clues with 3D models and vice-verse.
6. PI progresses from what is clearly interpretable to the less clear.

Expert PI accuracy is informally reported to be in the $70 - 80\%$ range. On the other hand, an automated individual tree inventorization process via airborne imaging faces a number of challenges including the need for high spatial resolution images, accurate image orthorectification and geo-referenced images. Further, such systems are quite sensitive to changing weather, sun position and seasonal variations in addition to the basic difficulty of segmenting overlapping canopies of similar colour and texture - conditions far more complex that aerial IU systems focused on building, roads and manmade structures. In spite of these challenges a number of systems have already been investigated over the past twenty years and they fall into two types of systems.

Image-Based Systems. Such approaches use simplifying assumptions about forest images. For example, [12, 19] use a token-based recognition approach which assumes a high-level of contrast between the tree crown and the surrounding area [10, 11, 12]. Tree canopies are detected and classified by a mixture of "valley-finding" (low intensity iso-contours), peak intensity detection [28, 29] and texture, structure and contextual image features. The underlying idea is to match (cross-correlate) pre-specified tree crown image(s) with the image at hand[18].

Tree Model-Based Approaches. Tree model approaches use an explicit model of 3D tree crowns to match trees in the supplied images. The STCI system [21, 20] uses a template matching approach. However, unlike the example-based approaches discussed above, the crown templates are synthesized from a tree crown model. The upper part of a tree crown (so-called "sun crown") is modeled as a generalized ellipsoid of revolution and then ray-tracing techniques are used to generate templates [16]. These systems still end up in cross-correlating crown templates with image data.

Though useful in some areas like conifer plantation inventories, all these methods fail with native forests where there is significant canopy overlap and mixed species, variations in pattern, shape and sizes. Further, they do not offer methods for validating the predicted PI nor fit with expert training procedures. For these reasons we have explored a different type of IU/PI model that does fit more closely to the PI training model where the treatment of uncertainty and inference processes play key roles in the interplay between what is known and what is perceived.

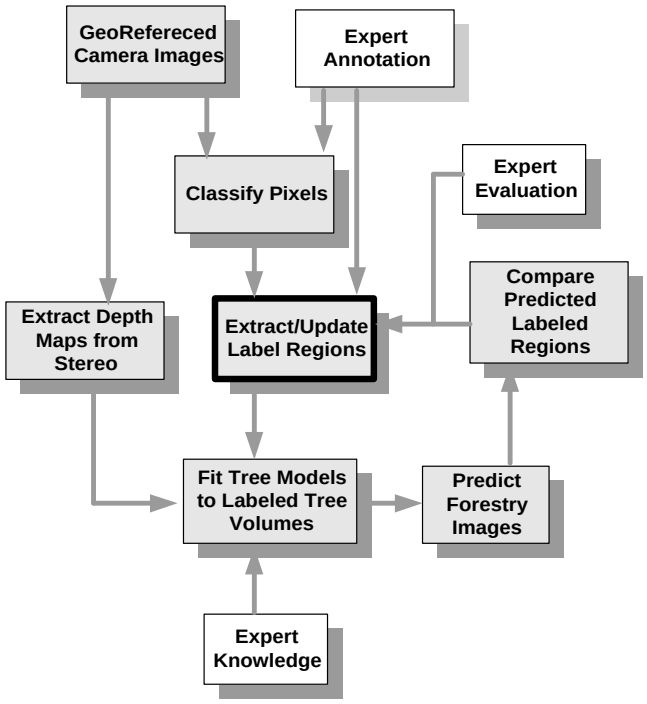

Figure 1: Proposed Photo interpretation model based on principles from training professional Photo Interpreters.

We have encapulated this in the model shown in Fig. 1 where both 2D and 3D aspects of the IU process are embedded and integrated into a unified probabilistic inference context. Again, the key two ideas behind this system are: one, to iterate through the cycle of inferring 3D scenes and objects from image segmentation, labeling and stereo, and then, in turn, predicting images from such 3D information which, again can be used by experts or computer systems to infer more consistent segmentation and labeling till there is consistency between forward and inverse modeling processes. Two, to define each component process in terms of an optimization problem whereby the desired outcomes are derived in terms of MAP (maximum posterior probabilities), given a component model and observations.

2 The Model Components

2.1 Segmentation and Labeling

In recent years there has been significant developments in formulating image segmentation and labeling as dependent and parallel processes using hierarchical hidden Markov models. In such models image annotation is viewed as defining labeling operators over sets of hierarchical (multiscaled) Markov Ran-

dom Fields(MRF) with the use of Maximum Likelihood (ML) and MAP criteria [8, 27, 2, 17, 1, 3]. Our model is closest, but not identical to, that of Cheng et. al. [3] using supervised learning to help estimate the important features within and between observation and label hierarchies. The basic model is shown in Fig. 2. The aim is to derive the optimal labelling of pixels given a model and observations. The labels, in turn, define regions: annotated segmented regions. What differentiate our approach from the others is that we explore a form of hierarchical constraint propagation using bijective operators and tuning of the model parameters to best fit expert annotation, and the use of colour images.

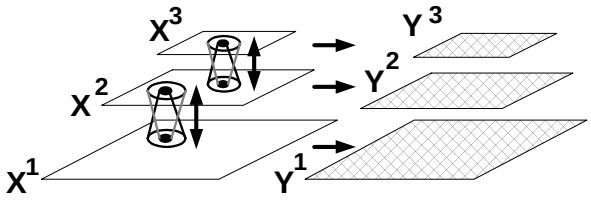

Figure 2: The basic hierarchical hidden Markov tree (HHMT) model. Here only three levels ($l-1$, l,$l+1$) of the multi-scale representation are shown. At each level there are two random fields corresponding to observed pixels, Y, image region labels defined by X. The bijection operations shown in grey and black colours represent upward and downward support kernel operators and associated region label transitions over movements upwards and downwards in scale encapsulating two types of contextual constrains.

In this model (see Fig. 2), each layer l consists of two random fields to encode the hidden labelling process (X^l) and the observed image (Y^l). The ith node (pixel) for the hidden and observation random fields are denoted by x_i^l and y_i^l, respectively. Since the posterior (label) probability at any given layer l, $p(X^l|Y^l)$, is computational intractable we introduce further assumptions which enable us to perform model parameters estimation in a feasible way. These are:

1. In any given layer l, the observed random field Y^l is solely depended on the hidden states at the same level, X^l:

$$p(Y^l|X^{ALL}, Y^{ALL}) = p(Y^l|X^l) \quad (1)$$

where X^{ALL}, Y^{ALL} refer to the complete labelling/observation hierarchy. Further, each observed data pixel is dependent only on its corresponding state (label):

$$p(Y^l|X^l) = \prod_i p(y_i^l|x_i^l). \quad (2)$$

2. The intra-layer hidden states are only dependent on their adjacent layers, i.e.:

$$p(X^l|X^{ALL}) = p(X^l|X^{\partial l}) \quad (3)$$

where X^{ALL} refers to the whole hidden hierarchy, and $X^{\partial l}$ corresponds to the neighborhood layers of X^l, i.e., X^{l-1}, and X^{l+1}.

3. In each layer l, the inter-layer hidden states (labels) are independent of each other, that is:

$$p(X^l|Y^l) = \prod_i p(x_i^l|y_i^l). \quad (4)$$

Consequently, the proposed hierarchical hidden Markov tree (HHMT) model (Fig. 2) is defined as:

$$\lambda = \{(\pi_l, A^+_{l-1,l}, A^-_{l,l-1}, B_l); l = 1, .., L\} \quad (5)$$

where the upward state transition matrices A^+ are defined by:

$$A^+_{l-1,l}(\vec{i}, j) = p(x_s^{(l)} = j|\hat{\partial s}^{(l-1)} = \vec{i}). \quad (6)$$

The downward transition matrices A^- are defined by:

$$A^-_{l,l-1}(\vec{i}, j) = p(x_s^{(l-1)} = j|\hat{\partial s}^{(l)} = \vec{i}). \quad (7)$$

For position j in layer l, $\hat{\partial s}^{(l+1)} = \vec{i}$ defines its contextual parents (clique, kernel), and $\hat{\partial s}^{(l-1)} = \vec{j}$ corresponds to its contextual children. In both cases, these kernels are defined by indexed histogram (see below). For each layer the prior probabilities of each label is defined by:

$$\pi_i^l = p(X^l = i). \quad (8)$$

derived from the relative frequencies of expected image labels at a given scale, l. The observation matrices B^l are defined by $B \equiv \{B^l, l \in [1..L]\}$, where $B^l(o, c) \in B^l$ characterize the likelihood of the observed pixel values (o) at level l, given label $c \in [1..C]$, which selected from a set of K Gaussian mixtures (defining the observation "symbols"; $G_K = \{G_k; k = 1, .., K\}$) in the label-dependent cluster space and corresponds, after normalization, to

$$B^l(o, c) = p(y^l = o|x^l = c). \quad (9)$$

The clustering method for extracting discrete numbers of "observations" used here follows Bouman's Minimum Description Length (MDL [24]) criterion based mixture of Gaussian method [1] for choosing the number of clusters.

Optimization problem I: Initial pixel labelling: Select pixel labels that maximize the following MAP criterion

$$\boxed{X^l = \arg\max_{X^l, k}(p(Y^l|X^l, G_k)p(X^l))}$$

Optimization problem II: Segmentation and Region Labelling. Given a model estimation procedure, initial estimates of pixel labels, image segmentation and labeling reduces to that of determining the most likely pixel labeling over scales, given the model and observed images

$$\boxed{X^l = \arg\max_{X^l,k}(p(Y^l|X^l,G_k)p(X^l|X^{\partial l})p(X^l))}$$

Brief details of the estimation and prediction procedures follow.

Estimation. The initial estimates of the model are obtained from expert annotated training images as follows:

1. For each training image, construct a Gaussian pyramid (where layer 1 to layer L correspond to the finest to coarsest levels, respectively).
2. The finest layer is manually labelled, and then subsampling (consistent with the upper frequency of each pyramid layer) is performed to obtain initial labels for the upper layers.
3. In the downward scanning phase, the transition matrix $A^-_{l+1,l}$ is determined by moving the kernels over the complete image layers $l+1$ and l, and continues in this direction until the finest layer is reached. A similar method is used in the upward scanning phase to determine $A^+_{l-1,l}$.
4. The prior vector π and the state-dependent observation matrices B's are determined from the Gaussian mixture model for each class c and layer l, using the MDL mixture of Gaussians clustering model.

Once this initial model is obtained from the training images, the process of image segmentation and labelling involves instantiating (and updating) the model from new image data as follows.

To interpret (annotate) new images, upward-downward recursion is used to propagate scene labels across layers until they are as compatible as possible with respect to the support kernel statistics. That is, we use an MAP criterion:

$$X^l = \arg\max_{x\in X} p(X^l|Y^l). \tag{10}$$

Prediction. Consequently,the upward-downward segmentation-annotation reduces to:

1. For a candidate image, generate the Gaussian pyramid.
2. Naive Bayesian classifier: Use the MAP principle, based purely on the B matrix and π, to obtain an initial labelling of layer L (the coarsest layer):

$$\hat{x_i^L} = \arg\max_{x_i^L \in X^L} p(x_i^L|y_i^L)$$

$$p(x_i^L|y_i^L) = p(y_i^L|x_i^L)p(x_i^L).$$

3. Downward recursion using π, A^- and B from the coarsest layer L to the finest layer 1:

$$\hat{x_i^l} = \arg\max_x p(x_i^l|y_i^l)$$

$$p(x_i^l|y_i^l) = p(y_i^l|x_i^l)p(x_i^l|\hat{\partial s_i}^{(l+1)}).$$

4. Upward recursion using π, A^+ and B, from the finest layer 1 to the coarsest layer L:

$$\hat{x_i^l} = \arg\max_{x\in X} p(x_i^l|y_i^l)$$

$$p(x_i^l|y_i^l) = p(y_i^l|x_i^l)p(x_i^l|\hat{\partial s_i}^{(l-1)}).$$

5. Iterate from step 3 to 4, and compute the log-likelihood score:

$$Q^l = \sum_i \log \sum_c p(x_i^l = c|y_i^l, \hat{\partial s_i}) \tag{11}$$

$$Q^* = \sum_{l=1}^{L} \gamma^l Q^l$$

where

$$\gamma^l = D^{2\times(l-1)}/P;$$

D is the size of the kernel(clique), P is the number of pixels, Q^l is the log-likelihood score for current layer l, and Q^* is for the image over all layers.

6. Repeat 1-5 until no label changes over all layers.

The convergence properties of such hierarchical relaxation operations have been studied before [6] and prove convergence to local minima. Consequently the initial pixel labeling plays an important role in the robustness of this procedure.

Performance of segmenter-annotator. We have used a region-based percentage overlap (PO) measure between corresponding expert and predicted segmented and annotated regions using a split-half design (half training and half unseen test images). That is, for all predicted regions R_j labelled i in the image, we compute:

$$PO(i) = max_j \frac{R_i \vee R_j}{R_i \wedge R_j} \tag{12}$$

where $A \wedge B$ and $A \vee B$ refer to the set intersection and union of regions A, B, respectively. This provides an objective measure of how well the predicted annotated regions fitted the observed ones. Our forestry image database consisted of 24 , 1024×1536 24-bit RGB images of experimental forestry plots, with an average of 65 trees per image. We have evaluated our model on 6 of these images where we had acquired expert annotated of $6 \times 64 = 384$ trees. These latter images were used for training and objective assessment of the model. The plots varied in terms of the degree of mixture densities of spruce and

Plot type	Spruce	Aspin
Pure Spruce	80.60	—
Mixture	69.08	46.37
Fall mixture	75.60	50.24

Table 1: Shows Percentage Overlap (PO) scores for: column 1 - Spruce, column 2 - Aspen over the three different plot mixtures: pure spruce and mixture of Aspen and Spruce over one 1024x1536 representative test image per type.

aspen and results for one such example are shown in Fig. 3. The images are taken from Alberta Research Council's experimental farm at Vegreville, Alberta. Throughout the simulations in this paper, four different classes (scene labels) were used as annotation: Aspen, Spruce, Shadow and Ground. Gaussian pyramid were used with σ values of $\{2, 4, 8, 16\}$, and the image colour intensities were directly used as observed features. For each such label, we have found that the parameters which offer the best PO scores occurred with the smallest kernels over all scales and largest or near largest number of scales, i.e., 3x3 and 6, respectively. Table 1 summarizes performance in terms of the PO scores and detection detection scores for each type of condition.

2.2 The stereo component

As with segmentation and labeling, here we frame the stereo problem as that of determining the most likely depth (disparity) map given the disparity space (a vector-valued image corresponding to differences between left and right images pairs as a function of, in this case, movements along the epipolar line[26] - but not necessarily). Further, we assume that the depth map is a Markov Random Field.

Optimization problem III: Select the optimal disparity map, d^*, associated with the reference view that will maximize the posterior probability of the disparity map given the observed disparity space Y and intrinsic parameters θ, as

$$\boxed{d^* = \arg\max_d p(d|\theta, y)}$$

The Stereo Model. Let $i = 1, \cdots, n$ index a 2D lattice of image pixels. let $y_l = \{y_{li}\}$ denote the left (reference) view pixel intensities and $y_r = \{y_{ri}\}$ denote the right view pixel intensities. Let $y = \{y_i\}$ denote a vector-valued image the same size as y_l where each vector component at each pixel corresponds to the intensity difference between the left and right images for a given disparity: $ds = [d_{min}, d_{max}]$. That is, for each pixel position i of the reference view there

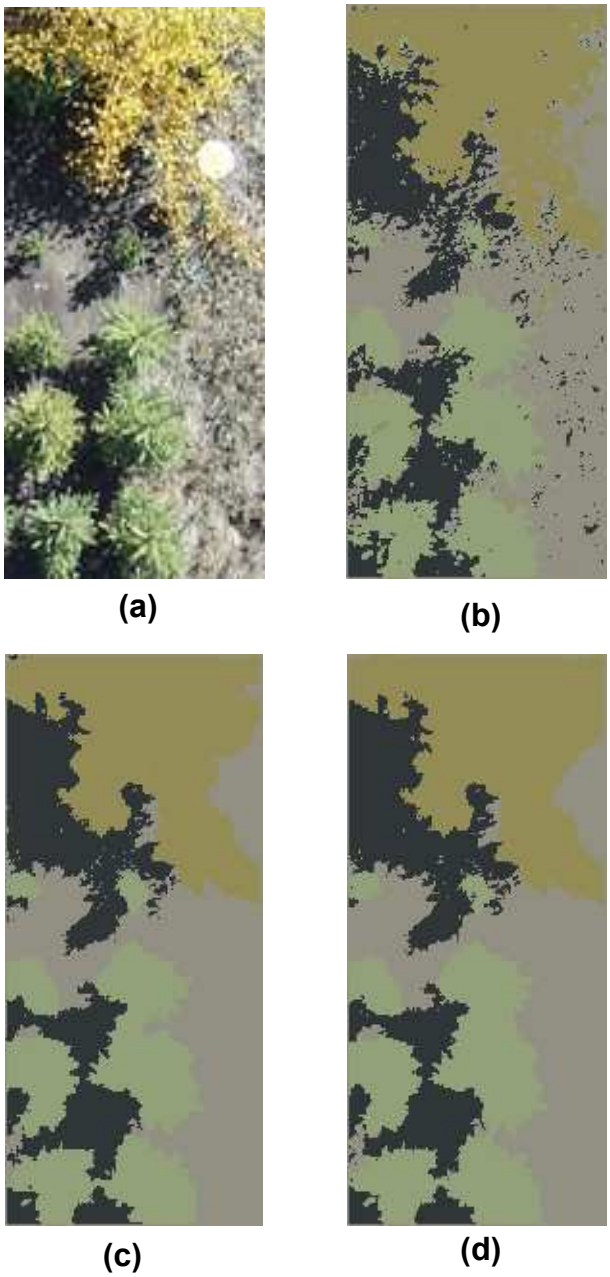

Figure 3: Shows (a) input image, (b) initially labeled pixels, (c) final output of segmenter compared to (d), expert annotation. Regions correspond to Ground, Aspen, Spruce and Shadows.

is a vector $\vec{v} = y_i$ whose components, $\vec{v_k}(k \in ds)$, encode the image intensity difference value with respect to disparity k.

Let $d = d_i, d_i \in ds$ denote a Markov Random Field (MRF) of disparity valued random variables, one per pixel, where a possible configuration of d defines a single disparity map derived from the left and right images. Clearly, the Markov property of

the MRF places constraints on a pixel's disparities as a function of its neigbourhood disparities. From this perspective, the stereo problem reduces to that of searching for an optimal configuration of d which minimizes a cost function defined over the left and right image pairs, intensity difference distributions, and associated MRF constraints using a Bayesian formulation as schematically shown in Fig. 4. Similar to the area of image reconstruction [15] [25], we derive a search algorithm using sampling techniques [9] for learning the hyper-parameters and use "Loopy Belief Propagation" [30] for deriving the optimal d.

Bayesian Analysis. As shown in Fig. 4, there are three components in the model: the disparity space Y; the MRF disparity map d which indexes, one site for one vector in $y \in Y$, a distribution of the disparity of the given site; and the hyper-parameters, σ's $\equiv \theta$. Because of the uncertainty of θ for different image pairs, Bayesian analysis treats θ as unknown and assigns prior densities for θ. Since we can establish the likelihood $p(y|d)$ and the priors $p(d|\theta)$ and $p(\theta)$, we can define the posterior as:

$$p(d, \theta|y) \propto p(y|d)\ p(d|\theta)\ p(\theta).$$

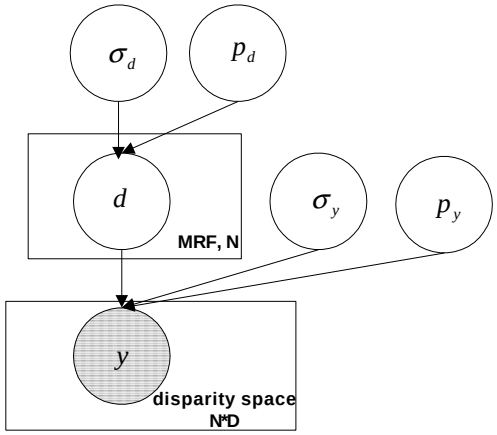

Figure 4: The stereo model. Here the grey circle denotes the observable variable (y: the disparity-indexed intensity difference maps, also called disparity space, are between the left and right images and forming a 3D volume, the same size of the MRF d with each site containing a size D vector) while the white circles correspond to unobservable variables. The d node represents the inferred disparity MRF relative to the neighbour scale size, σ_d, which is sampled from a scaled $Inv-\chi^2$ distribution with the scale σ_{d_1} with ν_1 degree of freedom. Similarly, the RF (Random Field) parameters apply to the observable image intensity difference volume with parameters σ_{y_0} and ν_0.

Our task is now two-fold. First, we need to infer the MAP disparity map d^* from:

$$d^* = \arg\max_d p(d|\theta^*, y)$$

where θ^* denotes the optimal estimate.

For computational efficiency we only calculate the mode (MAP) instead of sampling the whole posterior distribution (full conditional probability) of d. Loopy belief propagation (BP) [30] has been theoretically and eperimentally proven to suffice for this purpose (see below and for further detail, and refer to [30]). Second, we need to derive the adaptive estimation of hyper-parameters, θ,

$$\begin{aligned} \theta^* &= \int_d p(\theta|d)p(d|y,\theta)dd \\ &= E_{d|y,\theta}(p(\theta|d)) \end{aligned}$$

with either direct sampling or MCMC (Markov Chain Monte Carlo [9]), which draws dependent samples from the posterior of θ. That is, using MCMC we can actually evaluate the whole posterior distribution which incorporates our uncertainty of θ expressed in the prior, $p(\theta)$.

The data and prior models. Here we consider the data (likelihood) and prior models as the following [25] unified functional form:

$$p(x|\sigma, p) = \frac{1}{\sigma^n\ z(p)}\ \exp\left\{-\frac{1}{p}\ u(x|\sigma, p)\right\}$$

where:

$$u(x|\sigma, p) = \sum_i\ \rho(\frac{x_i}{\sigma}, p)$$

is the energy function; x represents a random field; $\rho(\cdot,\cdot)$ is the potential function with scale parameter $\sigma \in (0, \infty)$ and shape parameter $p \in [1, 2]$.

One reason for choosing this function is that the potential function $\rho(\cdot,\cdot)$ unifies many existing functional forms, both convex and non-convex, into one general representation[15]. It includes the Generalized Gaussian distribution:

$$\rho(\frac{x_i}{\sigma}, p) = |x_i/\sigma|^p \tag{13}$$

and when $p = 2$, the Gaussian distribution:

$$\rho(\frac{x_i}{\sigma}, 2) = (x_i/\sigma)^2. \tag{14}$$

As shown in Fig. 5, the Generalized Gaussian density function covers a spectrum of density functions with the shape varies significantly, from a more edge-preserving function with $p = 1$ (also called "double exponential density function"), to a smoother function with $p = 2$ (gaussian density function).

Here we model the probability of the disparity space (the data model) given the disparity map (related to the hyper-parameters) as:

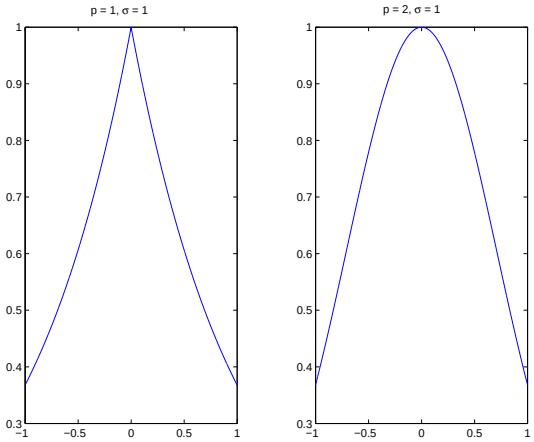

Figure 5: Examples of the Generalized Gaussian distribution. Left: double exponential distribution when taking the shape value $p = 1$ and the scale value $\sigma = 1$; Right: normal distribution when taking the shape value $p = 2$ and the scale value $\sigma = 1$.

$$p(y|d, \sigma_y, p_y) = \frac{1}{\sigma_y^n} \exp\left\{-\frac{u(y|\sigma_y, p_y)}{p_y}\right\} \qquad (15)$$

where σ_y corresponds to the "scale" parameter and p_y corresponds to the "shape" parameter for the data model.

Similarly, the Gibbs prior $p(d|\theta)$ of the MRF is modeled by:

$$p(d|\sigma_d, p_d) = \frac{1}{z(p_d)\ \sigma_d^n} \exp\left\{-\frac{u(d|\sigma_d, p_d)}{p_d}\right\} \qquad (16)$$

where the energy function is slightly different due to the MRF:

$$u(d|\sigma_d, p_d) = \sum_{j \sim k} \rho(\frac{d_j - d_k}{\sigma_d}, p_d) \qquad (17)$$

for $\{j, k\} \in n$ corresponding to all possible configurations of neighbouring pixels, j and k. Notice that $z(p_d)$ in Eq.16 is the partition function (normalizing constant) which is, in general, computational intractable.

Update algorithm. Again, although the sampling order can be chosen at random, a fixed sampling order is used here. The update procedure simply involves selecting an initial depth map. From this we can then compute the full conditional distributions of the depth dependent intensity distribution function from which, using MAP, we determine the optimal depth value at each pixel. Given these values, and the assumed MRF model, we can then update the depth values and, again, in turn, the intensity disparity distributions - till convergence. More formally, we have

1. Initialization. The hyper-parameters σ_y, σ_d, p_y and p_d are initialized to fixed values.
2. The MAP disparity map d is inferred via loopy belief propagation.
3. Update the distribution of σ_y from its full conditional probability. This is done by directly sampling from the $Inv - \chi^2$ density function.
4. Update the distribution of p_y from its full conditional probability. This step is done via Metropolis sampling (refer to Sec..
5. Update the distribution of σ_d from its full conditional probability by directly sampling from a $Inv - \chi^2$ density.
6. Update the distribution of p_d from its full conditional probability. Because an MRF is involved, we need to estimate the partition function up to a constant value, and use the Metropolis algorithm to update.
7. repeat steps 2 through 6 till reach a predefined number of iterations.

For detail description of this stereo component, please refer to [4].

Experimental Results Fig. 6 show an exemplar result of the stereo module. Since no stereo pairs were available for this project as yet, (a) and (b) are the left/right view of the input image pair for a synthetic forest consisting of 8 spruce trees and 8 aspen trees, with the terrestrial view (c). The stereo module outputs the initial disparity map (e) and consequently converge to the final disparity map (f). (d) shows corresponding ground truth disparity/depth values with an RMS difference to the predicted (f)of 0.78 using 8 discrete disparity values for the canopy regions alone (approximately 1 in 8 error rate).

2.3 Tree Generation and Fitting to 3D Labeled Regions

Here we develop a stochastic L-systems model. The concept behind L-Systems is that complex biological objects can be computer generated by successively replacing parts of a simple initial object using a set of rewriting or production rules each consisting of a specific geometric operation on an object. They are widely used to create plants and fractal objects [22]. Each L-System object is ultimately defined as a string generated from an initial string, the *axiom*, and a set of rewriting rules called *productions*. In each step of rewriting the axiom symbols are replaced by these productions. The number of rewriting steps is called recursive depth.

A simple example of L-Systems is given below:

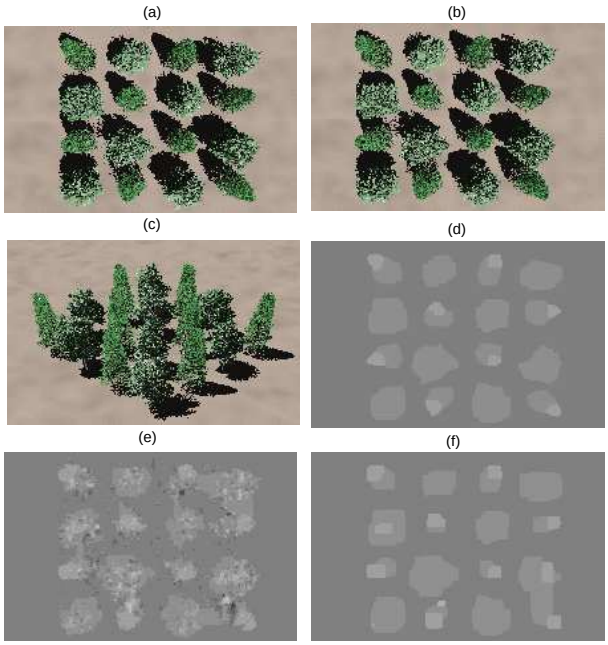

Figure 6: Experiment on one of the synthetic forest image pairs using the Generalized Gaussian model. (a) and (b) show the left/right view of the trees: eight aspens and eight spruces, with the side view (c). (d) is the ground-truth disparity map of (a); (e) is the disparity map of (a) at $t = 0$ step, and (f) is the disparity map at $t = 10$ step.

axiom: ab
rule 1: a -> bb
rule 2: b -> a
Final strings generated after different steps:
step 1: bba
step 2: $aabb$
step 3: $bbbbaa$

Upon completion of these rewriting processes a final string, termed L-String, is generated that corresponds to a set of geometric operations described by "turtle" graphics [22]. The turtle interpretation converts the resulting L-Strings into meaningful 3D geometrical models derived from turtle graphics. In 3D, the state of a turtle is defined by its position and three mutually perpendicular orientation vectors $\overrightarrow{H}$, $\overrightarrow{L}$, and $\overrightarrow{U}$ that indicate the turtle's heading, the direction to the left and up direction[22]. Imagining that a turtle in 3D space moves forward, rotates, etc. The trail of the turtle can be considered as geometrical shapes and it can be recovered by repeating its movements in order. Meaningful symbols in L-Strings are associated with these kinds of movements, or transformations.

The interpretations of some symbols are given below. Some symbols are associated with numerical parameters, and these parameters are also very important to the appearances of the final objects:

.Axiom
$NNSNNSNNSNNSNNSNNSNNSNNSNNB$
.Rules 1: 5
S=$'(.9)!(.9)$
N=$tF[\&'(0.8)!LBL\|zL] > (137)[z\&'(.7)!LBL\|zL] > (137)$
B=$tF[-'(.8)!(.9)\$LCL\|zL]'(.9)!(.9)C$
C=$tF[+'(.8)!(.9)\$LBL\|zL]'(.9)!(.9)B$
L=$[\ +f-f-f-(120)f-f-f]$

Table 2: A deterministic L-System rule description for generating a conifer

- "F" : Move forward a step of unit length and connect the new position with the last position by a line segment.
- "F(α)" : Move forward a step of length α and connect the new position with the last position by a line segment.
- "+" : Turn left by a default angle(counter clockwise) along the up axis.
- "+(α)" : Turn left by angle(counter clockwise) α along the up axis.
- "–" : Turn right by a default angle (counter clockwise) along the up axis.
- "–(α)" : Turn right by angle (counter clockwise) α along the up axis.

Figure 7: Shows the conifer generated by the deterministic L-System defined in Table 2.

Stochastic L-Systems. By rewriting operations L-Systems can generate very complicated objects as shown in Fig.7. These objects are always fractal in the sense that the rewriting rules used are the same. The results of L-Systems are predictable,

since all the operations are determinate. Stochastic L-Systems have evolved to overcome this problem by the introduction of probabilistic transitions between symbols (turtle operators), thus provide a biologically inspired statistical model[1] For example, the productions :

- p1 : a $\rightarrow$ (0.7) b a
- p2 : a $\rightarrow$ (0.3) c a

are 2 rewriting rules for the letter a. In one derivation step, either $p1$ or $p2$ are applied to each occurrence of a according to the given probabilities: $(0.7, 0.3)$.

This concept of probability-based selection of turtle operators needs to be formulated in terms of the rewriting form of L-Systems and for this reason, in the following section, we show how this can be accomplished via the use of a Hierarchical Hidden Markov Model(HHMM).

Hierarchical Hidden Markov Model L-Systems. Stochastic L-systems can generate objects with different structures, but cannot change the parameters of turtle operations. While HHMM L-Systems allow us to implement similar changes in the probabilistic formulation of Stochastic L-Systems with varied numerical parameters of the turtle operations. These are inherit from HMMs' transition matrices and observation matrices respectively.

We use the HMM nomenclature of Rabiner [23] where there exists a statistical model, a set of states, S_i, and a set of observation symbols, O_j. Each statistical model, a discrete HMM, λ, consists of three components $\lambda = \{A, B, \pi\}$ having N states and M distinct observation symbols; where $A = \{a_{ij}\}$ is an $N \times N$ state transition probability matrix and

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N. \quad (18)$$

$B = \{b_j(k)\}$ is an $N \times M$ matrix which is the probability distribution of observation symbol, o, given state j, where

$$b_j(k) = P[o = k | q = S_j], \quad 1 \leq j \leq N, 1 \leq k \leq M, \quad (19)$$

and $\pi = \{\pi_i\}$ is the initial state distribution where

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N. \quad (20)$$

In HMM L-Systems the states include turtle operations, productions operations: $F, +, -, G, E, R, \epsilon$, etc.

[1] Ideally we can learn this statistical model from both observed images and prior knowledge, but currently we set the model solely based on the prior knowledge from forestry science.

Symbols set: { N a M E L B D ‘ ! t F z & + – > $\mathcal{E}$ }

Rules:

- **E**=LBL|zL $\mathcal{E}$
- **L**={+f-f-f-(120)f-f-f} $\mathcal{E}$
- **N** = aM'! $\mathcal{E}$

B matrix

State	Means	STD
‘	0.95	0.05
!	0.95	0.05

Probability vector of super-state M

Observations	0	1	2	3
Probabilities	0.05	0.1	0.75	0.1

- **a** = tF $\mathcal{E}$

B matrix

State	Means	STD
F	1	0.1

- **M** = z&'!E> $\mathcal{E}$

π Vector

z	&	‘	!	E	>	$\mathcal{E}$
0.5	0.5	0	0	0	0	0

B matrix

State	Means	STD
&	40	10
‘	0.7	0.01
!	0.7	0.01
>	137	10

- **B**=aD'!B $\mathcal{E}$

B matrix

State	Means	STD
‘	0.9	0.02
!	0.9	0.02

- **D**= {L + - ' ! E $\mathcal{E}$ }

π vector

L	+	–	‘	!	E	$\mathcal{E}$
0.8	0.1	0.1	0	0	0	0

A matrix

	L	+	–	‘	!	E	$\mathcal{E}$
L	0	0.5	0.5	0	0	0	0
+	0	0	0	1	0	0	0
–	0	0	0	1	0	0	0
‘	0	0	0	0	1	0	0
!	0	0	0	0	0	0.9	0.1
E	0	0	0	0	0	0	1

B matrix

State	Means	STD
‘	0.9	0.02
!	0.9	0.02
+	35	10
–	35	10

Figure 8: An HHMM L-System models developed from L-System rules shown in Table.2. Transition matrices which are mainly deterministic are not shown in the tables.

The observations of the HMM are rewrite variable combinations of these states $F, +, -, G, E, R$

with numerical parameter values. Recursively, the $F, +, -, G, E, R$ combined operators encoded within these combinations also include the productions rules of the L-System and so more HMMs are used to model these rules and so on. This structure is defined by a hierarchy and so it is appropriate to call it a type of Hierarchical Hidden Markov Model(HHMM) [2]. All the production symbols from the axiom (the first level of the hierarchy) are replaced by their HMMs which generate (stochastic) L-Strings by Monte Carlo sampling of the underlying probability densities. Once computed, on the next level, production symbols in this L-String will be further replaced by observations inferred from the HMMs, again. This rewriting procedure will iterate till no HMM can be applied or the specified iteration depth is reached. There are three different states in this HHMM:

- Type 1. States whose observations are turtle operation symbols and not replaceable, such that these states can only be leaf nodes of HHMM.
- Type 2. States which are replaceable. These states are equivalent to *productions*. There is a special type of state termed "super-states".
  Each super-state is followed by an integer number n, which indicates this super-state will be repeatedly replaced by its sub-HMM n times. For example, X is a super-states, and $X(2)$ means the state X will repeat itself twice, and it is the same as XX.
  The benefit of super-states is that more constraints can be applied to control the generation procedure. For example, an HMM B is used to create a branch. Using the HMM, it is difficult to control the number of B in the observations by adjusting the transition matrix parameters. However, by super-states, the number of branches can be easily controlled by adjusting their observations vectors. These observation vectors can be gaussian models, probability vectors, etc.
- Type 3. Terminal states, which indicate the end of HMM sampling processes.

To this stage we have developed an HHMM for the generation of 3D objects. To fit these statistical models to labeled range map regions we have explored two procedures: one, the fitting of conical models to labeled segmented regions and, two, generating the Visual Hull (VH) of objects constructed from multiple images.

Optimization problem IV. Here we only consider the first approach using least squares methods. The aim is to maximize the fitting of the selected 3D tree model T_t from a pool of prototyped cone-like tree models $T(\phi(t)) \in \{T(\phi(1)), ..., T(\phi(T))\}$ [3], given both the 3D inferred convex hull (VH) of the region j (R_j) which is determined by the disparity map of this region, and the label(tree type t) of this region, as

$$\boxed{T_{R_j}(\phi(t)) = \arg\max_{\phi(t)}(p(T(\phi(t))/VH(R_j))}$$

After fitting the cones to the tree canopies we can then fit a tree generated by the HHMM L-system models into each cone. Branches are generated in order from roots up. Whenever a branches is accepted, its distance error is minimized, i.e. the branch is fully inside the cone, and it is as close as possible to the cone's boundary.

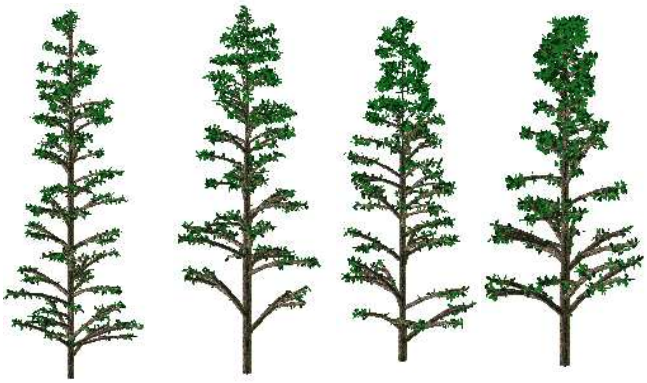

Figure 9: Shows four examples of a conifer generated by the HHMM-L System defined in 8 and to be compared with the deterministic version shown in Fig.7

2.4 Results: Validation/Update

The end result of these four optimization processes is the generation of a fully 3D forestry model from the image data so enabling different views and possible verification of the original labeling solution. As implied in Fig. 1 the aim of this type of approach to IU is to provide objective methods for verifying image annotation by using methods that can infer the underlying 3D model and, where appropriate, actually predict the initial image and segmentation used to generate the 3D model. Fig. 10 shows examples of this where equivalent images are generated from simple CAD-cone models with conifer and aspen texture maps and also S-LS models for the same data. At this stage we have not fully developed the integration model for forward and backward predictions

[2] There are many different formulations of hierarchical hidden Markov models and this is most similar to the one discussed in Singer[7]

[3] Here the 3D transformation $\phi(t)$ is the super-ellipsoid function, the same 3D tree model as used in [16]

but, rather, compare regions using expert annotations of both images. These resulted in near perfect overlap.

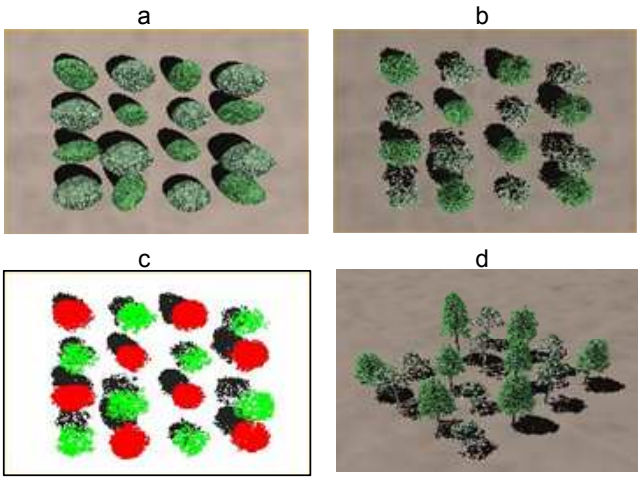

Figure 10: The 3D estimated results of the synthesized forest image (Fig. 6 (a)), (a): aerial view of CAD models, (b): aerial view of HHMM L-System trees fitted inside each CAD models, (c): labeled regions, (d): side view of the predicted HHMM L-System trees.

3 Conclusion and Future Work

In this paper we have considered the cycle of IU in terms of a common Bayesian inference approach where components are estimated at each stage by solving a Maximum Likelihood (ML) or MAP problem. We have also shown how this type of approach can generate 3D models and image predictions that can be compared either manually or automatically to original data for validation purposes. As to whether it is useful to fully and automatically "close-the-loop" is still an open question let alone the appropriate algorithm to use. For example, the predicted image could well be segmented by the same algorithm and the predicted regions compared to those predicted from the original image in order to update evidence for region labeling and tree identification. However, this may only propagate errors in initial segmentation so it is not fully clear that it would be generally useful. In the case of areas like forestry, cartography, medical image interpretation, where there is need to human checking of results the current ability to generate the 3D scene models and images from any view angle for human comparison with the predicted annotated data are the more useful outcomes of this work.

References

[1] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, Mar. 1994.

[2] Charles Bouman and Bede Liu. Multiple resolution segmentation of textured images. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(2):99–113, 1991.

[3] H. Cheng and C. A. Bouman. Multiscale bayesian segmentation using a trainable context model. *IEEE Transactions on Image Processing*, 10(2):460–474, April 2001.

[4] Li Cheng and Terry Caelli. Markov chain monte carlo stereo vision. In preparatioin, 2003.

[5] Li Cheng, Terry Caelli, and Victor Ochoa. A trainable hierarchical hidden markov tree model for colour image segmentation and labeling. In *Proceddings of the 16th International Conference on Pattern Recognition*, pages 192–195, Quebec city, Canada, 2002. IEEE Press.

[6] W.J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764, August 1995.

[7] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[8] Basilis Gidas. A renormalization group approach to image processing problems. *IEEE Trans. Patt. Anal. Mach. Intell.*, PAMI-11(2):164–180, 1989.

[9] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, 1996.

[10] F.A. Gougeon. Individual tree identification from high resolution meis images. In *Proceedings of the International Forum on Airborne Multispectral Scanning for Forestry and Mapping, D.G. Leckie and M.D. Gillis, editors*, pages 117–128, Forestry Canada, Chalk River, Ontario, 1993.

[11] F.A. Gougeon. A valley following approach to the automatic delineation of individual tree crowns in high spatial resolution meis images. *Unpublished manuscript*, 1994.

[12] F.A. Gougeon. Comparison of multispectral classification schemes for tree crowns individually delineated on high spatial resolution meis images. *Canadian Journal of Remote Sensing*, 21(1):1–9, 1995.

[13] R. Hall. *photogrammetry and photo interpretation (lecture notes of Forestry Engineering 201)*. Univ. of Alberta, 1996.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] D. Higdon, J. Bowsher, V. Johnson, T. Turkington, D. Gilland, and R. Jaszczak. Fully bayesian estimation of gibbs hyperparameters for emission computed tomography data. *IEEE Transactions on Medical Imaging*, 16:516–526, 1997.

[16] M. Larsen and M. Rudemo. Using ray-traced templates to find individual trees in aerial photos. In *Proceedings of the 10th Scandinavian Conference on Image Analysis, volume 2*, pages 1007–1014, Lappeenranta, Finland, 1997.

[17] M. Luettgen, W. Karl, A. Willsky, and R. Tenney. Multiscale representations of markov random fields. *IEEE Trans. Signal Process.*, 41(12):3377–3395, 1993.

[18] D. Murgu. Individual tree detection and localization in aerial imagery. *Department of Computer Science, University of British Columbia*, 1996. Master's thesis.

[19] A.J. Pinz. A computer vision system for the recognition of trees in aerial photographs. In *J.C.Tilton (ed.), Multisource Data Integration in Remote Sensing*, pages 111–124, Maryland: NASA, 1991.

[20] R.J. Pollock. A model-based approach to automatically locating tree crowns in high spatial resolution images. In *Image and Signal Processing for Remote Sensing. Jacky Desachy, editor*, pages 526–537, Rome, Italy, 1994.

[21] R.J. Pollock. The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown image model. *University of British Columbia, Vancouver, Canada*, 1996. PhD Thesis.

[22] P. Prusinkiewicz and A. Lindemayer. *The Algorithmic Beauty of Plants*. Springer Verlag, New York, 2 edition, 1996.

[23] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, Feb. 1989.

[24] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11:416–431, 1983.

[25] S. Saquib, C. Bouman, and K. Sauer. Ml parameter estimation for markov random fields, with applications to bayesian tomography. *IEEE Transactions on Image Processing*, 7(7):1029–1044, 1998.

[26] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002.

[27] M. Unser and M. Eden. Multiresolution feature extraction and selection for texture segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11:717–728, 1989.

[28] M. Wulder, K.O. Niemann, , and D. Goodenough. Local maximum filtering for the extraction of tree locations and basal area from high spatial resolution imagery. *Remote Sensing of Environment*, 73:103–114, 2000.

[29] M. Wulder, K.O. Niemann, and D. Goodenough. Error reduction methods for local maximum filtering. In *The Proceedings of the 22nd Symposium of the Canadian Remote Sensing Society*, Victoria, British Columbia, 2000.

[30] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *International Joint Conferences on Artificial Intelligence*, 20, 2001. Distinguished Lecture track.