DOI:10.1145/2629637

Pixels labeled with a scene's semantics and geometry let computers describe what they see.

BY STEPHEN GOULD AND XUMING HE

Scene Understanding by Labeling Pixels

PROGRAMMING COMPUTERS TO automatically interpret the content of an image is a long-standing challenge in artificial intelligence and computer vision. That difficulty is echoed in a well-known anecdote from the early years of computer-vision research in which an undergraduate student at MIT was asked to spend his summer getting a computer to describe what it "saw" in images obtained from a video camera.[35] Almost 50 years later researchers are still grappling with the same problem.

A scene can be described in many ways and include details about objects, regions, geometry, location, activities, and even nonvisual attributes (such as date and time). For example, a typical urban scene (see Figure 1) can be described by specifying the location of the foreground car object and background grass, sky, and road regions. Alternatively, the image could be summarized as a street scene. We would like a computer to be able to reason about all these aspects of the scene and provide both coarse image-level tags and detailed pixel-level annotations describing the semantics and geometry of the scene. Early computer-vision systems attempted to do just that by using a single unified model to jointly describe all aspects of the scene. However, the difficulty of the problem soon overwhelmed this unified approach, and, until recently, research into scene understanding has proceeded along many separate trajectories.

Along one of them, researchers aim to provide a high-level summary or categorization of a scene using a small number of tags (such as city and forest) without explicitly identifying the objects within it. Another, known as "object detection," aims to locate discrete objects (such as cars or pedestrians) in a scene by placing a bounding box around the objects. Face-detection algorithms in today's digital cameras and smartphones perform this task. However, these approaches do not provide detailed object outlines and fail to reason about the image as a whole.

Perhaps closest to the long-term scene-understanding goal is yet another trajectory that aims to produce annotations for the entire image, at the pixel level. Such a pixel-labeling, or semantic-segmentation, approach to scene understanding is our main focus

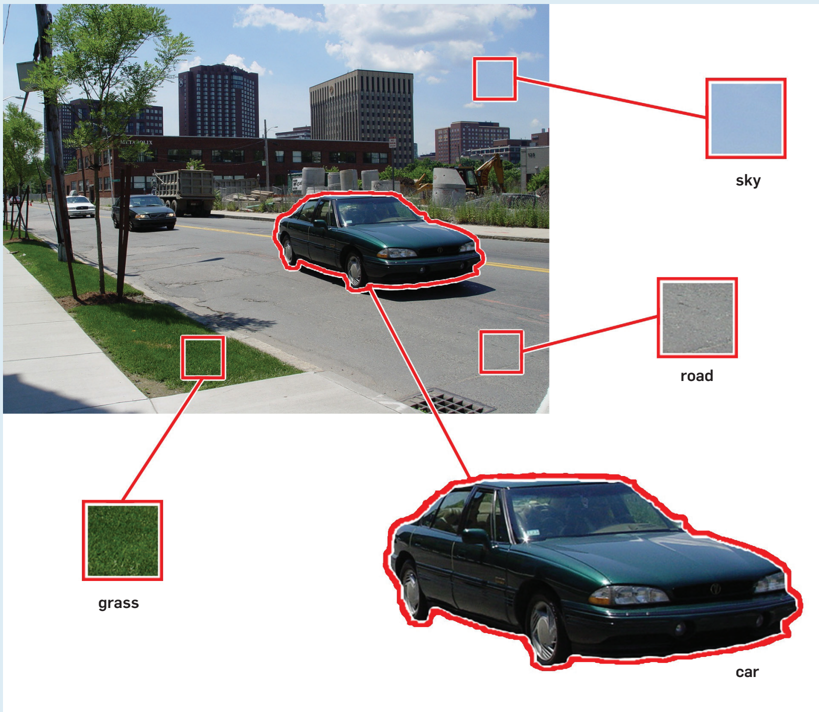
» key insights

- **Recent progress on image understanding, a long-standing challenge of AI, is enabling numerous new applications in robot perception, surveillance and environmental monitoring, content-based image search, and social-media summarization.**
- **One approach to image understanding is to label every pixel in an image with a category label using probabilistic models known as CRFs that can handle uncertainty and propagate contextual information across the image.**
- **Improved machine-learning techniques, more-powerful machines, and ever-growing volume of data are getting us closer to machines that are able to see and understand the world the way humans do.**

ILLUSTRATION BY ANDREW JOYCE

Figure 1. Understanding a typical urban scene in terms of foreground objects ("things") and background regions ("stuff").

The background regions are easily recognized from local color and texture cues, whereas foreground objects like the car require complex shape and appearance models.



here. Objects and background classes are segmented into discrete nonoverlapping regions and a label provided for each region, or, equivalently, all the pixels within it. In addition to labeling each pixel with a class label, different instances of each object can also be labeled with a unique identifier, so two adjacent cars are treated as, for example, disjoint objects. Such multiclass/multi-instance approaches are at the cutting edge of contemporary scene-understanding research. Hierarchical segmentation, so-called "scene parsing,"[37] can produce an even more refined view of the scene by breaking objects into component parts; for example, a car can be broken down into wheels, body panels, and windows.

These pixel-labeling approaches work with a predefined set of class labels that dictates the categories of objects and types of scenes the model can recognize. The labels can be semantic (such as grass, road, sky, and car) or geometric (such as horizontal, vertical, and slanted) and tuned for different scene types; for example, describing an indoor scene requires very different semantic and geometry classes from an outdoor scene. Scaling up the number and diversity of recognizable categories is a core thrust in contemporary research.

Most methods for pixel labeling use a probabilistic model known as a conditional Markov random field, or CRF, which provides a formal framework for encoding the complex relationship between the visual appearance of a scene and the underlying semantic (or geometric) labels. Moreover, the formalism admits efficient inference algorithms and allows model parameters to be learned from data. Some recent research (such as Heitz et al.,[15] Hoiem,[18] Li et al.,[28] and Yao et al.[40]) has sought to reunite the different research trajectories for scene understanding into a single coherent model incorporating high-level scene labeling, object segmentation, and geometric reasoning. This unified view allows, for example, constraints (such as object support and scale) to be expressed naturally by linking the semantic and geometric aspects of a scene. These state-of-the-art holistic models build on the pixel-labeling approach and employ the CRF framework to integrate the various aspects of the scene.

Here, we outline CRF model variants for scene understanding, showing how they exploit various assumptions (such as that cars typically appear on roads) about real-world scenes. Note, however, the problem of scene understanding remains wide open, with new innovations being introduced regularly. We also offer example results from a baseline CRF model on standard scene understanding datasets to demonstrate the capabilities and weaknesses of today's scene-understanding models.

Pixel Labeling

In pixel labeling, each pixel in an image is assigned a class label from a predefined set (such as grass, tree, road, car, and person). The assumption is that each pixel belongs to a single category of interest and that category can be identified unambiguously.

One approach is to classify each pixel individually without regard to the label assigned to other pixels in the image. However, as we show, treating each pixel independently can produce highly inconsistent results. A more sophisticated approach adopted by state-of-the-art scene-understanding algorithms is to consider the labeling of the pixels jointly by defining a random field over them.[13,34] Such an approach has enjoyed much success in recent years due to effective inference and learning algorithms.[3,22,31]

Conditional Markov random fields. CRFs were first introduced in the area of natural-language processing but have since found application in a range of machine-learning tasks.[25] Their key benefit is to provide a principled probabilistic framework for describing the relationship between related output variables (such as labels for pixels in an image) as a function of observed features (such as pixel colors). They are thus ideal for integrating multiple visual cues and combining related scene-understanding problems. Moreover, CRFs admit a compact representation and provide efficient (approximate) algorithms for inference and learning.

Formally, let $\mathbf{y} = (y_1,\dots,y_n)$ be a vector of discrete random variables (output variables) we are interested in predicting from some observed features $\boldsymbol{x}$. We assume each variable can take a label from a predefined finite set of labels $y_i \in L$. In pixel labeling the variable y_i denotes the label assigned to the i-th pixel in the image and n is the total number of pixels in the image. Typically, the features $\boldsymbol{x}$ are represented by real-valued vectors associated with the individual pixels or image regions. For example, $\boldsymbol{x}$ may encode color and texture cues, as discussed later.

The CRF model defines a probability distribution over the output variables, given the observed features via an energy function $E(\boldsymbol{y}; \boldsymbol{x})$ as follows

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\{-E(\boldsymbol{y};\boldsymbol{x})\} \qquad (1)$$

where $Z(\boldsymbol{x})$ is the so-called partition function that ensures the probability distribution is normalized correctly, or sums to one. Note, in general, computing the partition function is intractable since computation involves summing over the exponentially many assignments to $\boldsymbol{y}$. Fortunately, computation of the partition function is not necessary for inferring the most likely labeling, as we show here.

CRFs are compactly represented by decomposing the energy function $E(\boldsymbol{y}; \boldsymbol{x})$ as the sum of smaller clique potentials

$$E(\boldsymbol{y};\boldsymbol{x}) = \sum_c \psi_c(\boldsymbol{y}_c;\boldsymbol{x}). \qquad (2)$$

Here, each clique potential $\psi_c(\boldsymbol{y}_c; \boldsymbol{x})$ is a real-valued function defined over a subset of the random variables. We use the shorthand $\boldsymbol{y}_c$ to indicate the subset of variables in the clique, or the scope of the "clique potential."[a] Roughly, a clique potential encodes a numerical score for every joint assignment to the variables within its scope (ignoring all other variables in the model). Probabilistic influence propagates across the model via clique potentials that share variables, or have overlapping scope.

The number of variables within a clique potential defines the order of the model. Higher-order models can contain a very large number of variables within each clique. However, without appropriate structure, these models result in intractable inference problems. A common model for pixel labeling involves only unary and pairwise terms

$$E(\boldsymbol{y};\boldsymbol{x}) = \sum_{i=1}^{n} \psi_i^U(y_i;\boldsymbol{x}) + \sum_{ij \in \mathcal{E}} \psi_{ij}^P(y_i, y_j;\boldsymbol{x}). \quad (3)$$

There is one unary term associated with each pixel in the image, while the pairwise terms are defined over a set of pixel pairs ε. The set is usually sparse, containing only pairs of pixels that are neighbors in the image. Figure 2 is an example of a grid structured CRF defined over a four-connected neighborhood. In such a model, the label for each variable is influenced by some local features, as well as by the labels from surrounding variables. This influence can be captured by the unary and pairwise terms, respectively. In particular, the pairwise terms permit the encoding of smoothness assumptions; that is, a pixel in an image is likely to belong to the same object as its neighbors. Some recent research considers fully connected graphs in which a pairwise term exists between all pairs of pixels. This term between pairs of pixels allows long-range interactions to be captured but is tractable only when the pairwise terms take a specific form involving Gaussian kernels.[23]

Researchers are often interested in only the most likely interpretation of a scene; probabilistically, this is known as maximum a posteriori, or MAP, inference and requires solving

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmax}} P(\boldsymbol{y};\boldsymbol{x}). \qquad (4)$$

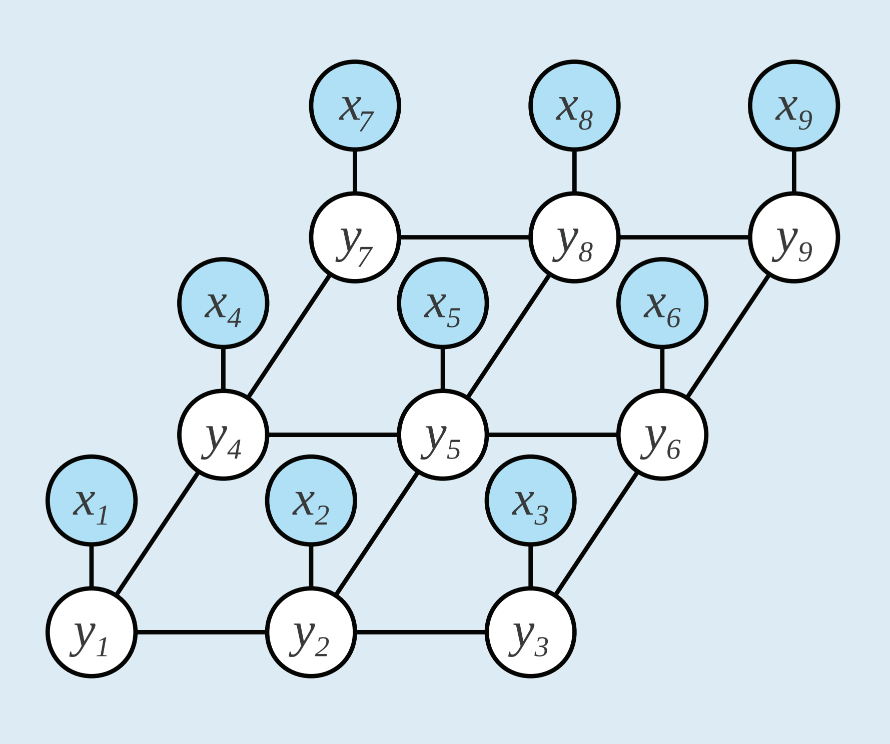
Since the partition function $Z(\boldsymbol{x})$ does not change with different assignments to $\boldsymbol{y}$, the most likely scene interpretation can be found by solving the equivalent energy-minimization problem

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y}}{\operatorname{argmin}} E(\boldsymbol{y};\boldsymbol{x}). \qquad (5)$$

For CRFs found in pixel-labeling problems, very fast algorithms have been developed for approximate energy minimization.[3,21,22] The most ubiquitous is a method known as "graph-cuts,"[3,21] a move-making algorithm that starts with an initial assignment to the variables and then iteratively solves a sequence of binary optimization problems that progressively improve on the solution at hand.

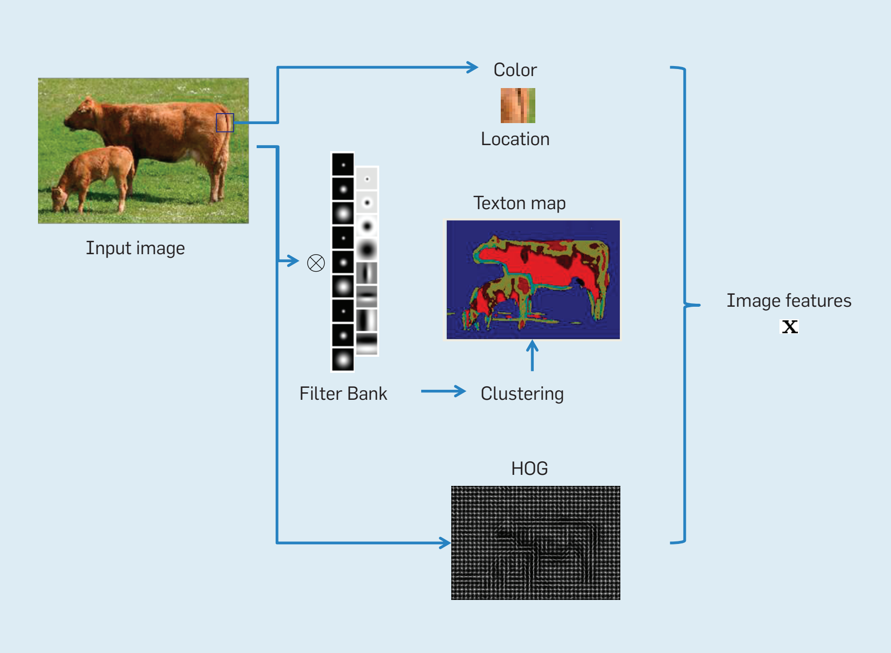
The clique potentials $\psi c(\boldsymbol{y}_c;\ x)$ can be specified in a variety of ways but often include parameters that control the value of the potential as a function of the variables and observed features; for example, the potential could

Figure 2. Graphical representation for a grid-structured pairwise CRF over nine random variables $\{y_1,\ldots,y_9\}$, with each variable represented by a node, direct pairwise correlations between variables by edges, and observed features $\{x_1,\ldots,x_9\}$ as shaded nodes.



[a] Formally, c indexes a sparse subset of elements from the power set of $\{1,\ldots,n\}$.

Figure 3. Typical image features used in the potentials of CRF models, including color, image location, filter outputs, and HOG; the filter outputs can be used to generate "texton features," or texture prototypes.



be defined as a linear combination of feature values and parameterized by a weight vector. Alternatively, the potential could be defined as the output of a decision-tree classifier and parameterized by node splits and leaf probabilities. The main difficulty in designing CRF models for scene understanding is in learning these parameters. Maximum-likelihood approaches cannot be applied due to the intractability of calculating the partition function. Many approximate learning techniques have been attempted. Popular among them are "piecewise learning with cross-validation"[34] and "pseudo-likelihood learning."[12] In recent years, max-margin learning approaches have shown success.[31] However, the most effective strategy for CRF parameter learning is still an open research question.

Features. One advantage of the CRF formulation is that a variety of image features can be used in the clique potentials and treated as observed values, or fixed for each image. Treating image features as observed values provides flexibility in scene representation and simplifies the model structure, as the model does not need to explicitly capture the probability distribution over these features. Different types of image features are tailored to the visual properties of the object classes of interest.

Computer-vision researchers affectionately divide object classes into two broad categories: "things" and "stuff." Background classes (stuff) are easily recognized from local appearance; objects (things) need additional cues (such as shape). Figure 1 reflects this distinction with three examples of background classes—grass, road, and sky—and a foreground object—car.

The local appearance features are color and texture cues. Two commonly used color spaces are RGB for its simplicity and CIE-Lab for color similarity closer to human perception.[35] For image features associated with regions, color histograms or summary statistics are often adopted to describe their overall appearance. The texture cues aim to capture repetitive local patterns in images and are usually extracted by filter banks and the distribution of their outputs. One widely used example is the texton, or small texture prototype, feature[30,34] that first clusters the filter bank outputs into a texton codebook, then represents an image patch as a histogram over the texton code words; see Figure 3 for several types of commonly used image features.

For foreground objects, however, local-appearance features usually have much greater variation than background classes. More distinctive and stable features are needed for reliable recognition. These features look beyond individual pixels and encode structural information; for example, object-shape information can be represented using the histogram of oriented gradient, or HOG, feature.[5] In addition, other higher-level features derived from top-down processing, discussed later, can be used to represent object properties. Moreover, global appearance features (such as average image color) can provide context for certain objects; for example, sheep tend to appear in greenish images, whereas cars tend to appear in grayish ones.

Image features can be concatenated as a feature vector $\boldsymbol{x}$ for each pixel and incorporated into the potential functions in CRF models. In particular, the unary potentials, which play a critical role in mapping image cues into labels, can be defined like this

$$\psi_i^U(y_i = k; \boldsymbol{x}) = \boldsymbol{w}_k^T \phi_i(\boldsymbol{x}) \qquad (6)$$

where w_k is a weight coefficient for label class k and $\phi_i(\boldsymbol{x})$ is a (non-)linear mapping of the features for pixel i. The weights w_k are determined from training data using, for example, structured prediction learning algorithms.[31]

The role of context. Context plays an important role in natural human recognition of objects and scene understanding. Consider the urban scene in Figure 4; we instantly recognize the street and somewhat surprisingly each of the cars in it despite the cars being fewer than 32 pixels high. This recognition is because the weak local evidence for the cars is compensated by strong contextual cues based on the scene's spatial layout.

Contextual information can be incorporated into a scene-understanding model in a number of ways. The simplest form of context is the statistical correlation between features and class labels; for example, blue pixels are more likely to be sky or water than grass or trees. Likewise, green pixels are more likely to be grass or trees than sky or water. This is exactly the type of context captured through the unary potentials in a CRF model. However, much more sophisticated contextual assumptions can also be encoded.

The co-occurrence of object classes (such as that cars often co-occur with road) can also be included in scene-understanding models.[32,40] Here, the relative positioning of the two objects is irrelevant; all that is important is they often appear together. This sort of context can be incorporated into a CRF by including binary variables z_k that indicate the presence or absence of each object class. Pairwise potentials linking these object variables z_k to the pixel variables y_i enforce consistency; that is, if an object is present, then some pixels must be labeled as that object and vice versa. Pairwise potentials between two object variables z_k and z_l then capture co-occurrence preferences. However, note that the resulting CRF graph is no longer a regular grid over pixels. More important, the very efficient graph-cut technique cannot be used for inference, so methods must resort to slower inference algorithms; see, for example, Yao et al.[40]

Class co-occurrence is a crude measure of context and does not capture the relative position of the objects. It can thus result in false detections placed inconsistently with respect to one another (such as a car floating in the trees). The spatial location of objects relative to the adjacent background[16] or relative location of objects across the scene[12] allows such errors to be corrected automatically by designing potential functions that encode this information in the model. Unfortunately, modeling the relative location of objects directly requires non-local reasoning that can make inference intractable. A global entity (such as the location of the horizon for outdoor scenes[19] or box-structure layout of a room for indoor scenes[14,39]) can be used to link objects indirectly and simplify reasoning. Here, additional variables (such as v_{hz} for the location of the horizon) are introduced into the CRF formulation (Equation 2) and linked via pairwise potentials to the pixel label variables y_i.

Pixels versus superpixels. An image is represented within a computer as a rectangular array of pixels, so associating labels with individual pixels for scene understanding is a natural choice. Moreover, the regular image structure is convenient for constructing CRFs with pairwise terms over adjacent pixels. However, pixels themselves are an artifact of the imaging process and do not necessarily reflect the underlying structure or complexity of a scene; for example, a one-megapixel photograph and an eight-megapixel photograph convey essentially the same information for the purpose of scene understanding despite the latter containing eight times as many pixels, hence requiring a bigger model. Pixels are noisy indicators of class; for example, in Figure 5, two adjacent pixels can have very different colors despite belonging to the same object.

A potential way to avoid an unnecessarily large number of variables is to model the image in terms of superpixels, or small contiguous regions of consistent appearance. Here, y_i in Equation 3 denotes the label assigned to the i-th superpixel, and n is the number of superpixels, as in Fulkerson et al.[9] A superpixel representation of an image can be generated by clustering adjacent pixels with similar color to produce numerous small regions.[4,8,33] Since superpixels are often much smaller than the objects of interest, the process is called "oversegmentation." Traditional oversegmentation methods are often slow and can result in highly irregular regions that do not honor object boundaries. A number of recent research efforts have attempted to produce superpixels with a more regular shape[1] and better alignment to true boundaries in the scene.

Many algorithms provide a way to control the number of superpixels. Figure 5 includes three different oversegmentations of an image into superpixels. The first (Figure 5b) produces too few superpixels, and much detail in the image is lost. However, the last (Figure 5d) maintains all essential structure in the image and achieves a 46 times reduction in the number of entities needed to describe the scene, from 68,160 pixels to 1,476 superpixels.

In addition to reduced model size, superpixels provide spatial support for computing features,[9,17] making these features less susceptible to noise. However, there are also a number of drawbacks to representing an image with superpixels rather than pixels. First, superpixels do not conform to a regular neighborhood structure, making it difficult for a programmer to weight the influence of each pairwise term in a CRF model. Second, superpixels commit to region boundaries as a pre-processing step, and these boundaries may not coincide with the true object boundaries.

The best of both worlds—pixels and superpixels—can be achieved by representing the image in terms of pixels but using superpixels to provide additional features or enforce higher-order consistency constraints.[20] For example, the model can penalize labelings where pixels within a superpixel disagree on their class label. Such penalties can be implemented in a CRF model through a higher-order Potts potential

$$\psi_c^H(\boldsymbol{y}_c) = \begin{cases} 0 \text{ if } \exists \ell \in \mathcal{L} \text{ s.t. } y_i = \ell \text{ for all } i \in c \\ \lambda \text{ otherwise} \end{cases} \tag{7}$$

where a cost of $\lambda > 0$ is paid if not all variables within clique c take the same label. In this way, the algorithm prefers boundaries defined by the superpixels, but, given enough evidence to the contrary, this assumption can be overridden.

Superpixels can also be used within iterative inference procedures (such as move-making algorithms[10,11]) or data-

Figure 4. Example low-resolution image showing the importance of context for object detection; recognizing the highlighted object is difficult in isolation but in the context of a street scene is easily identified as a car.

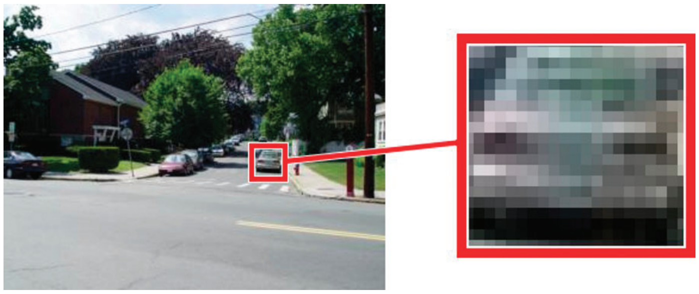
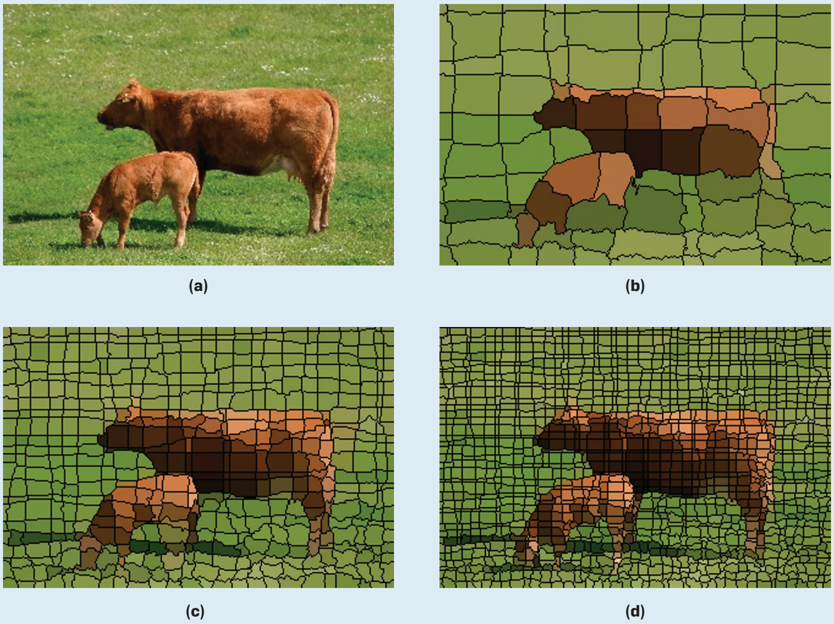


Figure 5. An image and various oversegmentations of the image into superpixels; the number of superpixels controls the trade-off between model complexity and representation accuracy.



driven Markov chain Monte Carlo algorithms.[38] Here, the superpixels guide the inference procedure by proposing changes to the boundaries of objects for the current interpretation of the scene at hand.

Top-down vs. bottom-up. CRF models provide a unified framework for integrating image cues and prior assumptions over scene-label configurations. Image cues used for defining unary clique potentials or forming superpixel representations are usually referred to as "bottom-up information." These bottom-up cues describe the similarity between pixels and can be used to generate informative proposals for the labeling task (such as larger homogeneous regions). We view the class-dependent unary potentials as part of the bottom-up process, though they also include the label information. The most common prior assumptions over labels are defined on neighboring (super)pixels (via pairwise terms in the CRF) and represent a soft smoothness constraint on region labels. While this constraint is an effective prior on region labels, it lacks the expressive power needed for object-specific information (such as global shape and pose). We refer to the priors on these latter properties as "top-down information," as they encode knowledge at a level beyond pixels and simple image regions.

Object shape is probably the most important type of object-level information. Shape priors are usually represented as either rigid or deformable masks that assign each pixel a cost of belonging to the foreground object corresponding to the mask[2,27] (see Figure 6). Deformable masks can capture pose variation suitable for object classes (such as people and animals) and are often implemented through a part-based model where individual components of the mask can move relative to one another. As shape is a global property of foreground regions, shape priors naturally lead to large cliques and therefore more complex inference algorithms. In practice, some mask representations decompose into a sum of local unary or pairwise terms; for instance, given a matched shape mask, pixels lying within the mask can be encouraged to take specific labels by modifying costs within the corresponding unary potentials.

One common strategy for incorporating object priors is to use object detection to generate object-instance proposals (such as bounding boxes) and define object-shape potentials on those regions.[11,24] The resulting energy function usually has higher-order terms, as in Equation 2, in which the clique c is defined by pixels within the object-support regions, and the term favors coherent labeling of those pixels.

Higher-order energy functions can be optimized by generalized graph-cut inference algorithms[20] or transformed into pairwise models by adding auxiliary variables. Alternatively, both bottom-up and top-down information can be used in the design of the move-making algorithms that iteratively propose energy-reducing changes to variable assignments; for example, object-detection outputs can generate powerful top-down moves, overcoming fragmented object-labeling results from pure bottom-up-driven segmentation.[11]

As discussed earlier, some recent approaches take a holistic view toward scene understanding by integrating scene segmentation with multiple scene-analysis tasks (such as scene classification, object detection, and depth/layout estimation[15,18,40]). In such a framework, several levels of top-down information, including geometry, candidate object bounding boxes, and scene category, are incorporated into a single CRF energy function. Concretely, variables are introduced to represent these quantities, and their relationships to pixel labels are modeled through pairwise or higher-order potential functions. The overall problem can be decomposed into smaller tasks and solved in an alternating fashion or treated as a unified objective to be optimized jointly.

Datasets and Software

Many software implementations of scene-understanding algorithms are freely available. Two notable examples are the Darwin software framework[b] and the Automatic Labeling Environment (ALE),[c] providing infrastructure for scene understanding via pixelwise CRF models, as described earlier. The most basic models consist of unary and pairwise potentials. The unary potentials are constructed using local and global appearance features. The pairwise clique potentials are defined on either four-connected

[b] http://drwn.anu.edu.au
[c] http://cms.brookes.ac.uk/staff/PhilipTorr/ale.htm

or eight-connected neighborhoods around each pixel and often involve a contrast-sensitive smoothness prior[34] that discourages adjacent pixels from taking different labels when the pixels are similar in color.

Moreover, many well-labeled datasets are readily available, with many researchers using them to develop and compare scene-understanding algorithms. To give a flavor of the results that can be achieved, consider the following results on two standard datasets from the Darwin software framework:

The Stanford Background Dataset (SBD)[10] *consisting of 715 images of rural, urban, and harbor scenes.* Images are labeled from two different label sets: the first captures semantic class and includes seven background classes (sky, tree, road, grass, water, building, and mountain) and a single foreground object class; the second captures scene geometry (sky, vertical, and horizontal). Each image pixel is allocated two labels, one semantic and one geometric; and

The Microsoft Research Cambridge (MSRC) dataset[34] *consisting of 591 images.* Pixels are labeled with one of 23 different classes. However, due to the rare occurrence of the horse and mountain class, they are often discarded. Pixels not belonging to one of the remaining 21 categories are ignored during training and evaluation. One drawback of this dataset is the ground-truth labeling is rough and often incorrect near object boundaries. Nevertheless, the dataset contains a diverse set of images and is widely used.

As scene-understanding research matures, larger and more diverse datasets are becoming more important for applying existing scene-understanding algorithms and inspiring new ones. The PASCAL Visual Object Classes (VOC) dataset[6] is a very large collection of images annotated with object-bounding boxes and pixelwise segmentation masks for 20 different (foreground) object categories. It contains approximately 20,000 images organized into numerous challenges, with training, validation, and evaluation image sets pre-specified. Another large dataset of interest to scene-understanding researchers is the SIFT Flow dataset,[29] a subset of outdoor images from the LabelMe image repository (http://labelme.csail.mit.edu), which contains 2,688 images annotated using 33 diverse object and background categories. Performing well on both these datasets requires a combination of many of the techniques described earlier.

Accuracy of scene-understanding algorithms can be evaluated by many measures, including sophisticated boundary-quality metrics and intersection-over-union (Jaacard) scores. The simplest measure computes the percentage of pixels that were correctly labeled by the model on a "hold out," or separate, set of images, referred to as the "test set" or "evaluation set." As is standard practice when evaluating machine-learning algorithms, these images should not be viewed during training of the model parameters. Formally, we can write

$$\text{acc} = \frac{\sum_{i=1}^{n} [\![\hat{y}_i = y_i^*]\!]}{n} \qquad (8)$$

where $\hat{y}_i$ is the label for pixel i predicted by the algorithm, y_i^* is the ground-truth label for pixel i, and $[\![\cdot]\!]$ is the indicator function taking value one when its argument is true and zero otherwise. An alternative evaluation metric that better accounts for performance on rare categories is class-averaged accuracy, defined as

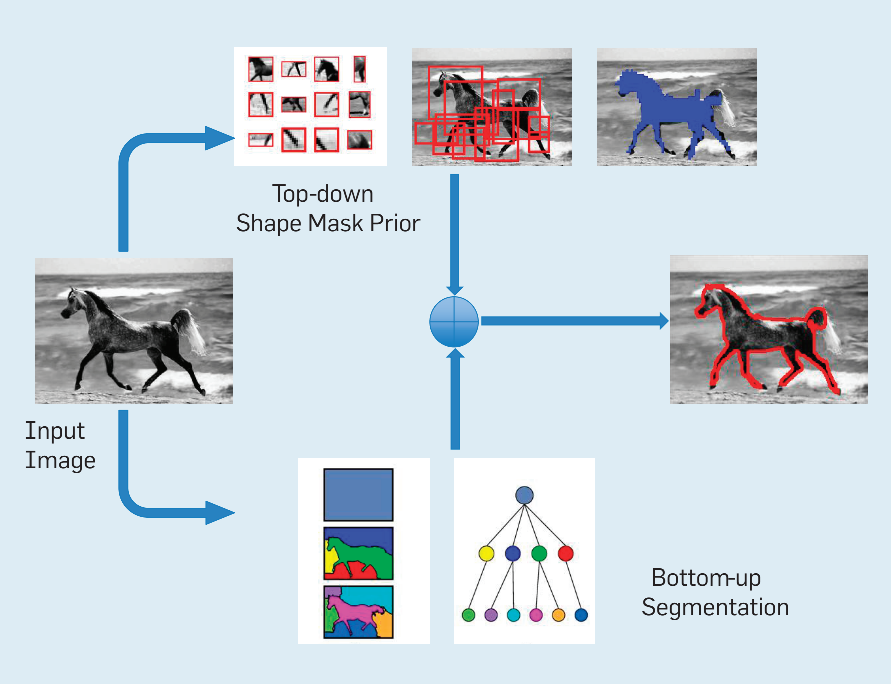
$$\text{acc}^{\text{class–avg}} = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{\sum_{i=1}^{n} [\![\hat{y}_i = \ell) \wedge (y_i^* = \ell)]\!]}{\sum_{i=1}^{n} [\![y_i^* = \ell]\!]} \qquad (9)$$

The different accuracy measures defined by Equation 8 and Equation 9 are often referred to in statistics as "micro averaging" and "macro averaging," respectively.

State-of-the-art performance on the semantic categories of the MSRC and Stanford Background datasets is approximately 86% and 77% pixelwise accuracy, respectively; class-averaged accuracies are typically 5%–10% less. On larger datasets, performance can be quite poor without top-down and contextual cues, especially on the less frequently occurring classes.

Illustrating the effects of different aspects of a scene-understanding model, Figure 7 includes results on an example image from the MSRC dataset. Classifying pixels independently (left results column) produces very noisy predictions, as shown. Adding a pairwise smoothness term helps remove the noise (right side). However, when the features are weak (top row), the algorithm cannot correctly classify the object in the image, though the background is easily identified using local features. Stronger features, including local and global cues, as discussed, coupled with the pairwise smoothness term, produce the correct

Figure 6. Combining bottom-up and top-down information in a CRF framework.[2] The bottom-up process forms region hypotheses and maps them to semantic labels. The top-down process represents part-based shape constraints on object instances.



labeling result (bottom right).

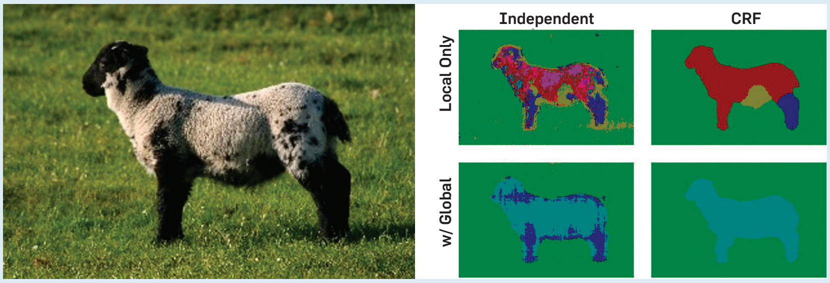
These examples of semantic segmentation are indicative of more general trends in scene-understanding algorithms. More sophisticated features that incorporate contextual information (such as pixel location and global and shape-based features) perform much better than local appearance features, in general. Moreover, CRF models, with their pairwise smoothness priors, improve performance over independent pixel classification, but the benefit decreases as the sophistication of the features used by the independent classifiers increases. This trade-off is to be expected, as these features allow both the baseline performance to increase and the features to encode contextual information that can act as a surrogate for the smoothness assumption.

The qualitative results from the Darwin software framework (see Figure 8) also highlight a few points; as shown, the accuracy of the predictions is generally good, and the model is able to identify the boundary between object categories quite well. The labeling of foreground objects occasionally leaks into the background. This leakage is more prominent in the MSRC results and can be attributed to, in part, rough ground-truth labeling in that dataset. In models that use superpixels, these boundary errors can also be caused by inaccurate over-segmentations.

An interesting result is the labeling of the ducks in Figure 8 (MSRC, left column, third row down). Here, the water is classified correctly, but both ducks are labeled incorrectly. The white duck is mislabeled as water by the model due to both confusion of its local appearance with that of water and a strong smoothness assumption preferring to label it with the same class as the surrounding background. The second duck is mislabeled as a boat. While this may seem absurd to humans, it is a reasonable mistake for an algorithm when considering the context is correct (boats also co-occur with water) and the model was trained on only a handful of images containing ducks.

Figure 7. Example semantic segmentation for an image from the MSRC dataset.

Shown are the original image (left) and color-coded pixel labels (right) from different scene-understanding models. The models vary by features (local appearance versus local and global appearance) and model complexity (independent pixel classification versus a CRF model with pairwise term); see Figure 8 for the related color legend.

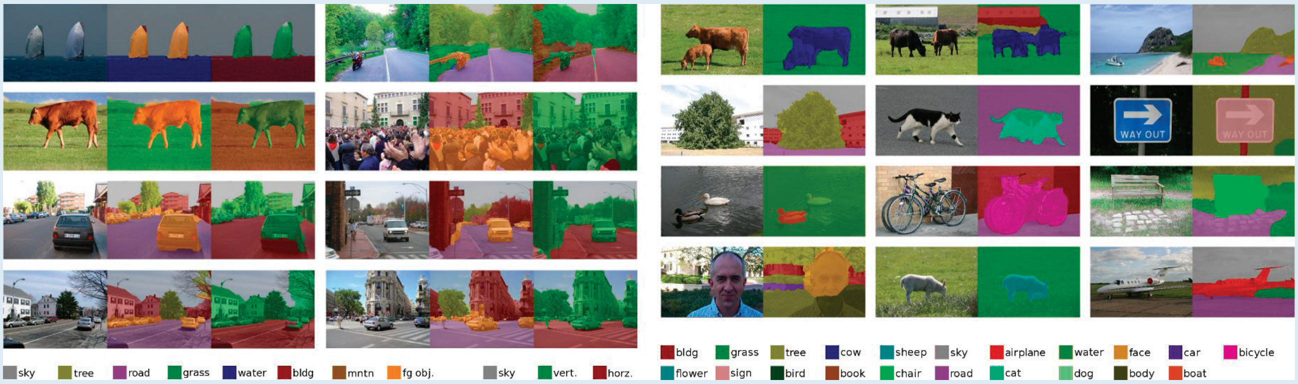


Conclusion

We have explored scene understanding as a pixel-labeling task, including a number of technical challenges facing scene-understanding algorithms and a glimpse at current trends toward addressing them. Active research along these lines and the growing availability of high-quality datasets reflect current scene-understanding research; for example, better low-level feature representations are being learned automatically from large volumes of data rather than engineered by hand.[26] Researchers are also looking toward mid-level visual cues (also called "attributes") to overcome some of the limitations of scarce training data; for example, knowing an object has feathers narrows the range of possible labels for that object.[7]

Moreover, improved learning algorithms based on structured-prediction models[31] means large numbers of parameters can be tuned simultaneously. This results not only in more optimal parameters but enables use of richer models (such as those with parameterized higher-order terms). Other models being studied are hybrid models (such as grid-structured

Figure 8. Representative results on two standard scene-understanding datasets produced by the Darwin software library; shown are the original image and predicted class labels overlay; best viewed in color.



CRFs) combined with a shape-constraining counterpart (such as restricted Boltzmann machines, or RBMs, and multi-scale patterns). Here, the CRFs capture appearance and smoothness, while the RBM and its variants encourages global consistency over the shape of objects. The models are still in their infancy but show great promise.

Another exciting direction is the use of non-parametric label-transfer techniques to allow for greater scalability in terms of the size of the dataset and the diversity of object categories.[29,36] These techniques overcome (somewhat) the assumption of a closed-world set of labels implicit in the CRF formulation but introduce other complications (such as more-expensive test-time computation) and the need to resolve language ambiguities (such as whether "water" and "river" are semantically equivalent and, more important, refer to the same object).

One may ask whether it is necessary to label every pixel in an image. Indeed, for some scene-understanding tasks (such as face detection), a rough bounding box may suffice. However, for a detailed description of a scene, along the lines envisaged by early computer-vision researchers, pixel-level labeling seems inevitable. Advances on this front (such as those discussed here) and their integration into coherent holistic models are getting us closer to when a computer is indeed able to describe what it sees.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (Nov. 2012), 2274–2282.
2. Borenstein, E., Sharon, E., and Ullman, S. Combining top-down and bottom-up segmentation. In *Proceedings of the IEEE Workshop on Perceptual Organization in Computer Vision at the IEEE Conference on Computer Vision and Pattern Recognition* (Washington, D.C., June 27–July 2). IEEE Computer Society Press, 2004, 46–46.
3. Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (Nov. 2001), 1222–1239.
4. Comaniciu, D. and Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619.
5. Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (San Diego, CA, June 20–25). IEEE Computer Society Press, 2005, 886–893.
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., and Zisserman, A. The Pascal visual object classes challenge. *International Journal of Computer Vision 88*, 2 (June 2010), 303–338.
7. Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. Describing objects by their attributes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Miami, FL, June 20–25). IEEE Computer Society Press, 2009, 1778–1785.
8. Felzenszwalb, P.F. and Huttenlocher, D.P. Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (Sept. 2004), 167–181.
9. Fulkerson, B., Vedaldi, A., and Soatto, S. Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the 12th International Conference on Computer Vision* (Kyoto, Japan, Sept. 29–Oct. 2). IEEE Computer Society Press, 2009, 670–677.
10. Gould, S., Fulton, R., and Koller, D. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the 12th International Conference on Computer Vision* (Kyoto, Japan, Sept. 29–Oct. 2). IEEE Computer Society Press, 2009, 1–8.
11. Gould, S., Gao, T., and Koller, D. Region-based segmentation and object detection. In *Advances in Neural Information Processing Systems 22* (Vancouver, B.C., Canada, Dec. 6–11). Curran Associates, Inc., 2009, 655–663.
12. Gould, S., Rodgers, J., Cohen, D., Elidan, G., and Koller, D. Multi-class segmentation with relative location prior. *International Journal of Computer Vision 80*, 3 (Dec. 2008), 300–316.
13. He, X., Zemel, R.S., and Carreira-Perpinan, M. Multiscale conditional random fields for image labeling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Washington, D.C., June 27–July 2). IEEE Computer Society Press, 2004, 695–702.
14. Hedau, V., Hoiem, D., and Forsyth, D. Recovering the spatial layout of cluttered rooms. In *Proceedings of the International Conference on Computer Vision* (Kyoto, Japan, Sept. 29–Oct. 2). IEEE Computer Society Press, 2009, 1849–1856.
15. Heitz, G., Gould, S., Saxena, A., and Koller, D. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems 21* (Vancouver, B.C., Canada, Dec. 8–13). Curran Associates, Inc., 2008, 641–648.
16. Heitz, G. and Koller, D. Learning spatial context: Using stuff to find things. In *Proceedings of the European Conference on Computer Vision* (Marseille, France, Oct. 12–18). Springer, Berlin, Heidelberg, 2008, 30–43.
17. Hoiem, D., Efros, A.A., and Hebert, M. Recovering surface layout from an image. *International Journal of Computer Vision 75*, 1 (Oct. 2007), 151–172.
18. Hoiem, D., Efros, A.A., and Hebert, M. Closing the loop on scene interpretation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Anchorage, AK, June 23–28). IEEE Computer Society Press, 2008, 1–8.
19. Hoiem, D., Efros, A.A., and Hebert, M. Putting objects in perspective. *International Journal of Computer Vision 80*, 1 (Oct. 2008), 3–15.
20. Kohli, P., Ladicky, L., and Torr, P.H. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision 82*, 3 (May 2009), 302–324.
21. Kolmogorov, V. and Zabih, R. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (Feb. 2004), 147–159.
22. Komodakis, N., Paragios, N., and Tziritas, G. MRF optimization via dual decomposition: Message-passing revisited. In *Proceedings of the International Conference on Computer Vision* (Rio de Janeiro, Oct. 14–21). IEEE Computer Society Press, 2007, 1–8.
23. Krahenbuhl, P. and Koltun, V. Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in Neural Information Processing Systems 24* (Granada, Spain, Dec. 12–17). Curran Associates, Inc., 2011, 109–117.
24. Ladicky, L., Russell, C., Kohli, P., and Torr, P.H. Graph cut-based inference with co-occurrence statistics. In *Proceedings of the 11th European Conference on Computer Vision* (Crete, Greece, Sept. 5–11). Springer, Berlin, Heidelberg, 2010, 239–253.
25. Lafferty, J.D., McCallum, A., and Pereira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning* (Williamstown, MA, June 28–July 1). Morgan Kaufmann, San Francisco, 2001, 282–289.
26. Le, Q.V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., and Ng, A.Y. Building high-level features using large scale unsupervised learning. In *Proceedings of the International Conference on Machine Learning* (Edinburgh, Scotland, June 26–July 1). Morgan Kaufmann, San Francisco, 2012.
27. Levin, A. and Weiss, Y. Learning to combine bottom-up and top-down segmentation. *International Journal of Computer Vision 81*, 1 (Sept. 2008), 105–118.
28. Li, L.-J., Socher, R., and Fei-Fei, L. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Miami, FL, June 20–25). IEEE Computer Society Press, 2009, 2036–2043.
29. Liu, C., Yuen, J., and Torralba, A. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 12 (Dec. 2011), 2368–2382.
30. Malik, J., Belongie, S., Shi, J., and Leung, T. Textons, contours and regions: Cue integration in image segmentation. In *Proceedings of the International Conference on Computer Vision* (Corfu, Greece, Sept. 20–25). IEEE Computer Society Press, 1999, 918–925.
31. Nowozin, S. and Lampert, C.W. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision 6*, 3–4 (May 2011), 185–365.
32. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. Objects in context. In *Proceedings of the International Conference on Computer Vision* (Rio de Janeiro, Oct. 14–21). IEEE Computer Society Press, 2007, 1–8.
33. Ren, X. and Malik, J. Learning a classification model for segmentation. In *Proceedings of the International Conference on Computer Vision* (Nice, France, Oct. 13–16). IEEE Computer Society Press, 2003, 10–17.
34. Shotton, J., Winn, J., Rother, C., and Criminisi, A. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision* (Graz, Austria, May 7–13). Springer, Berlin, Heidelberg, 2006, 1–15.
35. Szeliski, R. *Computer Vision: Algorithms and Applications*. Springer, Berlin, Heidelberg, 2011.
36. Tighe, J. and Lazebnik, S. SuperParsing: Scalable nonparametric image parsing with superpixels. In *Proceedings of the European Conference on Computer Vision* (Crete, Greece, Sept. 5–11). Springer, Berlin, Heidelberg, 2010, 352–365.
37. Tu, Z., Chen, X., Yuille, A.L., and Zhu, S.-C. Image parsing: Unifying segmentation, detection and recognition. *International Journal of Computer Vision 63*, 2 (July 2005), 113–140.
38. Tu, Z. and Zhu, S.-C. Image segmentation by data-driven Markov chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 657–673.
39. Wang, H., Gould, S., and Koller, D. Discriminative learning with latent variables for cluttered indoor scene understanding. In *Proceedings of the 11th European Conference on Computer Vision* (Crete, Greece, Sept. 5–Sept. 11). Springer, Berlin, Heidelberg, 2010, 497–510.
40. Yao, Y., Fidler, S., and Urtasun, R. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (Providence, RI, June 16–21). IEEE Computer Society Press, 2012, 702–709.

Stephen Gould (stephen.gould@anu.edu.au) is a fellow (senior lecturer) in the Research School of Computer Science in the College of Engineering and Computer Science at the Australian National University, Canberra, and visiting researcher in the Machine Learning Research Group at the National ICT Australia, Canberra.

Xuming He (xuming.he@nicta.com.au) is a senior researcher in the Computer Vision Research Group at the National ICT Australia, Canberra, and an adjunct fellow (senior lecturer) in the Research School of Engineering in the College of Engineering and Computer Science at the Australian National University, Canberra.