

# Context and Hierarchy in a Probabilistic Image Model

Ya Jin

Division of Applied Mathematics  
Brown University  
yajin@dam.brown.edu

Stuart Geman

Division of Applied Mathematics  
Brown University  
geman@dam.brown.edu

## Abstract

*It is widely conjectured that the excellent ROC performance of biological vision systems is due in large part to the exploitation of context at each of many levels in a part/whole hierarchy. We propose a mathematical framework (a “composition machine”) for constructing probabilistic hierarchical image models, designed to accommodate arbitrary contextual relationships, and we build a demonstration system for reading Massachusetts license plates in an image set collected at Logan Airport. The demonstration system detects and correctly reads more than 98% of the plates, with a negligible rate of false detection. Unlike a formal grammar, the architecture of a composition machine does not exclude the sharing of sub-parts among multiple entities, and does not limit interpretations to single trees (e.g. a scene can have multiple license plates, or no plates at all). In this sense, the architecture is more like a general Bayesian network than a formal grammar. On the other hand, unlike a Bayesian network, the distribution is non-Markovian, and therefore more like a probabilistic context-sensitive grammar. The conceptualization and construction of a composition machine is facilitated by its formulation as the result of a series of non-Markovian perturbations of a “Markov backbone.”<sup>1</sup>*

## 1. Introduction

By the theory of nonparametric inference, essentially any classification or estimation problem can be solved, more or less automatically, from a sufficiently rich and sufficiently lengthy sequence of examples. This was already well known within the statistics community in the early 1980’s, by which time several elegant approaches had been explored, including, for example, kernel estimation (e.g. [28]), k-nearest-neighbor classification (e.g. [10]), and

Grenander’s method of sieves [22]. The problem of over-fitting (a.k.a. controlling variance, model selection) was already front and center, resulting in the development of various analytic and practical methodologies (e.g. bounds derived from the Vapnik-Červonenkis dimensionality [42], cross validation and generalized cross validation [35, 45], and information-based measures of complexity [30]).

The availability of these remarkable tools invites a *tabula rasa* approach to the problem of computer vision. In principle, it is possible to formulate the object recognition problem in terms of a search for one or more decision surfaces in a high-dimensional image representation, and in principle it is possible to solve the problem, as well as it can possibly be solved, by nonparametric estimation. Indeed, recent advances in learning theory (e.g. Vapnik [43], Freund & Schapire [14]), coupled with relentless advances in computing technology, have rendered this approach practical for certain applications, such as the recognition of isolated hand-written digits.

More ambitious vision problems require more in the way of *a priori* structure, dictated by the need to control variance (over-fitting) and practical limitations in the size of any manageable set of training data. Yet *a priori* structure means *a priori* bias (cf. [18]), and the search for an *appropriate* class of models, embodying *the right structure* for unconstrained vision problems, is therefore critically important.

Here we propose a structure based upon the dual principles of hierarchy and reusability. Several observations argue for this general approach:

**Feature and Part Sharing.** Reusability is a common theme in computer vision. Indeed, the notion of a feature itself, such as a Gabor filter, or a locally invariant Sift feature [27], is already based upon the assumption that, from scene to scene, the same feature will participate in the representation of any one of a multitude of different entities. Biederman [4] and others have argued that a possibly small number of reusable parts might be sufficient to compose the large ensemble of shapes and objects that are in the repertoire of human vision. Empirical evidence for sharing

<sup>1</sup>Supported by Army Research Office contract DAAD19-02-1-0337, National Science Foundation grant DMS-0427223, and National Science Foundation grant IIS-0423031 as part of the NSF/NIH Collaborative Research in Computational Neuroscience Program.

comes from studies of the diminishing numbers of new parts that are needed to represent objects in a sequential learning task (Krempp et al. [24]), as well as from the successes of multiple-object recognition systems built on a common substrate of lower-level parts [2, 37].

**Context.** It is often observed that segmentation can be ambiguous, if not impossible, in the absence of the contextual information provided through recognition. Similarly, reliable edge and boundary detection is notoriously difficult when attempted in a purely bottom-up framework, without more global contextual constraints that help to disambiguate, for example, texture, shadow, and occlusion boundaries (cf. [7, 38]). By little more than their nature, hierarchical models (as in [15, 16, 21, 29, 33]), embody multi-level contextual constraints.

**Efficient Representation.** Barlow [3] proposed *suspicious coincidences* as a possible principle for discovering meaningful groupings, such as the grouping of features into parts, parts into objects, or objects into scenes. A new label, “tree”, “telephone”, “desk”, makes for a more efficient representation by virtue of “explaining” an otherwise “suspicious coincidence” in the arrangement of features and parts. Much earlier, Laplace [25] made a similar observation, arguing that a likelihood principle was sufficient to provide a gradient towards meaningful grouping. These notions of grouping are closely related to the notion of efficient representation, in that the introduction of a label for an otherwise unlikely grouping of parts amounts to an enhanced encoding and a shorter description length (as discussed for example by Bienenstock et al. [5]). By this connection, hierarchical description is a close cousin of Rissanen’s Minimum Description Length principle [30].

**Biology.** Fodor and Pylyshyn [12] have questioned the biological relevance of the (nonparametric-type) learning algorithms employed in most neural network models. They argue that these models lack a fundamental feature of human cognition – they are not compositional. The principle of compositionality holds that humans perceive and organize information as a syntactically constrained hierarchy of reusable parts. The prototypical formulation was introduced by Chomsky [8] as a system of formal grammars. Indeed, language itself is the prototypical compositional system, with evident hierarchy, syntax, and reusability. In the visual world, physical objects and scenes decompose naturally into a hierarchy of meaningful and generic parts, and it is perhaps no coincidence that there is an apparent hierarchical structure in the ventral visual pathways of the more highly evolved visual systems [32, 39, 41].

In §2, following the formulation proposed in [17], we develop a prior probability model on hierarchically organized image interpretations (“composition machine”). We begin with a Markov structure, in the spirit of a Bayesian net-

work, and later perturb this distribution in order to achieve greater selectivity. An application to licence-plate reading is explored in §3, and some conclusions and speculation are offered in §4.

## 2. Model: Composition Machine

Composition systems are generative, probabilistic, image models that embody a hierarchy of part/whole relationships. Generative probabilistic models include Bayesian networks [13, 23, 34, 36], linear and nonlinear filtering [11], Markov random fields [9, 31, 46], and probabilistic context-free grammars [19]. Compositional systems are distinct from these models in that they are non-Markovian. On the one hand this makes computation substantially more difficult, but on the other hand, non-Markovian models are more selective and thereby, in principle, capable of smaller type II error probabilities (probabilities of false alarms).

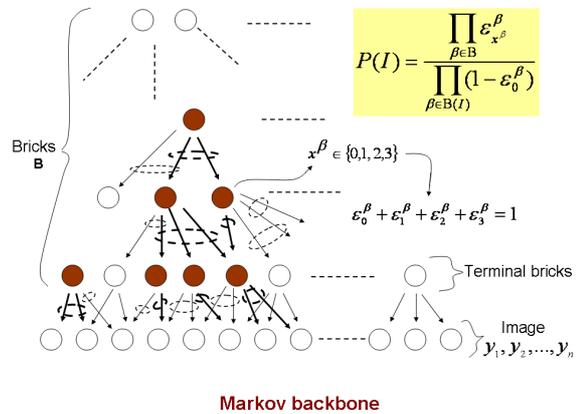


Figure 1. Vertical slice through a “composition machine”. Each row extends to a two-dimensional sheet of “bricks”. See text for details.

**Markov Backbone.** Figure 1 depicts the ‘Markov backbone’, which is a generative, hierarchical model equipped with a Markov structure on a directed acyclic graph. Starting at the bottom, the image pixels are represented by a one-dimensional string of nodes, corresponding to a one-dimensional slice through the two-dimensional pixel array. Hidden (model) variables are associated with two-dimensional sheets of nodes that sit “above” the image array; these variables are called *bricks* (as in Lego bricks) to emphasize their re-usability across legitimate configurations. The layer of bricks that sit immediately above the image array are called *terminal bricks*, and as we shall see, are associated with local image filters.

Bricks represent semantic variables, like edges, strokes, junctions, shapes, and various parts and objects. Assignments will vary from application to application; Figure 2 indicates the assignments for the application to license-plate

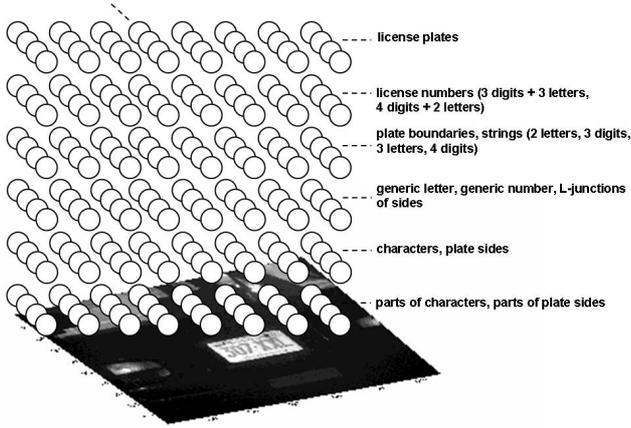


Figure 2. Semantic hierarchy for plate-reading application. All told, there are about 500,000 bricks. See text for details.

reading. Each sheet of bricks comprises a regular sublattice of each type of variable, so that the third layer above the image, in Figure 2, is interspersed with “generic-letter bricks”, “generic-number bricks”, and “L-junction bricks”.

A brick can be *on* or *off*. An *on* brick selects one subset of “children bricks” from an allowed collection of such subsets drawn from the layers below. The possible sets of children are depicted with broken ovals in Figure 1.<sup>2</sup> We refer to these selections as “compositions”, which is exactly what they are in the sense that the selected children are composed, as “parts”, into the “object” represented by the *on* brick. Put another way, the *on* brick is *instantiated* by the selected children. An image *interpretation*, corresponding to a semantic labeling of a scene, is a subgraph of *on* bricks, each substantiated by an allowed set of children bricks, which themselves must be *on*. Such subgraphs are called *complete*. See Figure 1, where an example is highlighted with colored nodes and bold arrows. The set of all interpretations is denoted  $\mathcal{I}$ .

The state of a brick, say the brick  $\beta \in \mathbf{B}$ , is a random variable,  $x^\beta \in \{0, 1, \dots, n^\beta\}$ , with  $x^\beta = 0$  representing *off*, and  $x^\beta = 1, 2, \dots, n^\beta$  representing the selected set of children in Figure 1. The pixels themselves (actually, their grey levels) are represented by a vector of intensities,  $\vec{y}$ .

The Bayesian framework has two components: a prior distribution, here on the set of interpretations,  $\mathcal{I}$ , and a conditional data model, meaning a probability distribution on  $\vec{y}$  for each  $I \in \mathcal{I}$ . As mentioned earlier, we start with a

<sup>2</sup>For simplicity of the figure, children are depicted as residing exclusively in the layer immediately below. In fact, children can reside at any level below a parent. As an example – see Figure 2 – a plate-boundary brick from the third layer from the top composes with a license-number brick from the second layer from the top to instantiate a license-plate brick in the top layer.

Markovian distribution on  $\mathcal{I}$ . Each brick  $\beta \in \mathbf{B}$  is assigned a probability vector  $(\epsilon_0^\beta, \epsilon_1^\beta, \dots, \epsilon_{n^\beta}^\beta)$ . In terms of these parameters, the probability  $P(I)$  of an interpretation (i.e. a complete subgraph)  $I$  is

$$P(I) = \frac{\prod_{\beta \in \mathbf{B}} (\epsilon_{x^\beta}^\beta)}{\prod_{\beta \in \mathbf{B}(I)} (1 - \epsilon_0^\beta)} \quad (1)$$

where  $\mathbf{B}$  is the set of all bricks and  $\mathbf{B}(I)$ , the “below set”, is the set of all *on* bricks that are not roots of the (directed) subgraph  $I$ . One way to verify that  $\sum_{I \in \mathcal{I}} P(I) = 1$  is by a thought experiment: choose, independently and according to the respective probability vectors, the states of the bricks in the top layer. Next choose, also independently, the states of the bricks in the penultimate layer, using again the respective probability vectors, except that selected children of *on* bricks are conditioned to themselves be *on*. Continue downward, finally choosing the states of the terminal bricks. The procedure selects a complete subgraph  $I \in \mathcal{I}$  according to the distribution  $P(I)$  and establishes the Markov property with respect to the directed acyclic graph represented by nodes and arrows in Figure 1.

**Perturbation – the Compositional Distribution.** One way to assess a Bayesian model, in this case the “Markov backbone” defined in Equation 1, is to examine samples. The upper panel in Figure 3 is a random sample from the set of instantiations of a “4-digit-string” brick of the fourth layer (counting from the bottom) of the composition machine for reading plates depicted in Figure 2. The black and white pixels represent the filters associated with the states of the selected terminal bricks in the instantiation. (The filters themselves are reusable parts of characters – see discussion of data models below.) The evident poor fit of the subtrees (numerals and parts of numerals) is a signature of the Markov property. Whereas the distribution accommodates the basic structures of interest, the coverage is too broad. This works against selectivity, and hence ROC performance, in a recognition system. One approach to the coverage problem is through expanded state spaces – the state of a brick can be elaborated to include detailed positional information about its instantiation. This solution, very much analogous to adopting attribute grammars in computational linguistics, can be short-sighted since the potential number of relevant and interacting attributes (position, size, stroke width, color, etc.) is potentially unmanageable in a Markov system.

We have chosen instead to treat each attribute with a non-Markov perturbation, starting with the Markov backbone. Briefly, the derivation is as follows: Associate with each brick  $\beta \in \mathbf{B}$  a (possibly vector-valued) attribute function  $a^\beta(I)$ , which measures the “fit” among the “parts” that instantiate  $\beta$ , as it appears in the particular interpretation  $I \in \mathcal{I}$ . If  $\beta$  is a “4-digit-string” brick, specifically, then

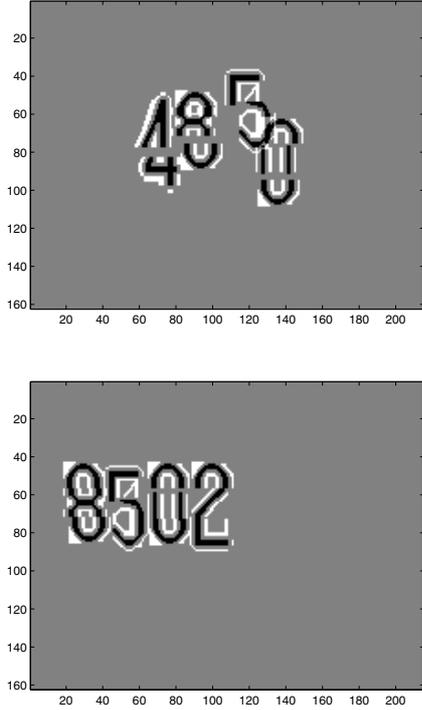


Figure 3. Samples from Markov backbone (upper panel, ‘4850’) and compositional distribution (lower panel, ‘8502’).

$a^\beta(I)$  returns the relative coordinates of the four numerals that instantiate  $\beta$  in the interpretation  $I$ . Similarly, each character brick, and each numeral in particular, has an associated attribute function that computes the relative coordinates of the particular parts that are composed into that character in a particular interpretation. A “compositional distribution” is built from a Markov backbone (Equation 1) and a pair of probability distributions,  $p_\beta^c$  (“composed”) and  $p_\beta^0$  (“null”), on each attribute  $a^\beta$ . The former, *composed* distribution, captures regularities of the arrangements (i.e. instantiations) of the children bricks, given that they are parts of the object represented by  $\beta$ ; the latter, *null* distribution, is the attribute distribution in the absence of the non-Markovian term. The set of relative coordinates of the three parts that make up the ‘0’ in the upper panel of Figure 3 is an example of an attribute, and the particular arrangement of the parts in the figure is a sample from the corresponding null distribution.

In a compositional distribution, the *null* attribute distributions are compared to their *composed* counterparts: given  $I \in \mathcal{I}$ ,

$$P(I) \propto \frac{\prod_{\beta \in \mathbf{B}} (\epsilon_{x^\beta}^\beta)}{\prod_{\beta \in \mathbf{B}(I)} (1 - \epsilon_0^\beta)} \prod_{\beta \in \mathbf{A}(I)} \frac{p_\beta^c(a^\beta(I))}{p_\beta^0(a^\beta(I))} \quad (2)$$

where  $\mathbf{A}(I)$ , the “above set”, is the set of non-terminal *on*

(active) bricks. The proportionality sign ( $\propto$ ) can be replaced with equality ( $=$ ) if, at the introduction of each attribute function,  $a^\beta$ , care is taken to ensure that  $p_\beta^0(a^\beta)$  is exactly the current (“unperturbed”) conditional distribution on  $a^\beta$  given  $x^\beta > 0$ . In general, it is not practical to compute an exact null distribution and  $P$  must be re-normalized.

The effect on coverage of the perturbation can be seen by comparing the upper and lower panels in Figure 3. For each non-terminal brick  $\beta$ , the denominator,  $p_\beta^0(a^\beta)$ , was approximated by assuming that in the absence of an explicit constraint, the prior distribution on  $a^\beta$  is the one consistent with independent instantiations of the children. The numerator,  $p_\beta^c(a^\beta)$ , was constructed to encourage regularity in the relative positions of character parts, and of characters, in composing characters and strings, respectively. The upper panel is a sample instantiation from the Markov backbone; the lower panel is a sample instantiation from the full compositional distribution. Samples from the full compositional distribution can be computed (at considerable computational cost) through a variant of importance sampling.

**Conditional Data Models.** The data model connects interpretations to the grey-level image, and completes the Bayesian framework. In the license-plate-reading demonstration system, we have assumed that the data distribution, conditioned on an interpretation, is a function only of the states of the terminal bricks:

$$P(\vec{y}|I) = P(\vec{y}|\{x^\beta : \beta \in \mathcal{T}\})$$

where  $\mathcal{T} \subseteq \mathbf{B}$  is the set of terminal, or bottom-row, bricks.

Good performance in most image analysis applications requires some degree of photometric invariance. In the context of a probability model, the notion of invariance is closely connected to the statistical notion of sufficiency. The following data model, employed in the demonstration system, is an example of the application of sufficiency to invariance. As remarked earlier, the terminal bricks in the demonstration system represent reusable parts of alphanumeric characters. The *states* of the terminal bricks code the local position of the represented part. Some of the parts can be more-or-less clearly discerned from the upper-hand (Markov) panel in Figure 3. The zero and the eight are each made of three parts whereas the four and the five are each made of two parts. The black portion of each “part filter” represents image locations that are expected to be dark, relative to the locations represented by the white portion of the filter. The *rank sum*  $R$  (cf. Lehmann [26]) of the intensities of the corresponding “black” pixels, among the union of intensities of black and white pixels, is a convenient statistic that is demonstrably invariant to all monotone transformations of the image histogram. We model pixel grey levels by assuming that their distribution depends only on  $R$  ( $R$  is *sufficient*), and we model  $R$  with an exponential probability distribution, thereby promoting small rank sums cor-

responding to dark-on-light characters. Pixels that are not referenced by any active terminal brick are modeled as uniformly and independently distributed. More details on this data model can be found in [20].

**Parsing.** The *a posteriori* distribution on interpretations, given a particular image as represented by  $\vec{y}$ , is

$$P(I|\vec{y}) = \frac{P(\vec{y}|I)P(I)}{P(\vec{y})} \propto P(\vec{y}|I)P(I) \quad (3)$$

The interpretation  $I$  corresponds to a full semantic analysis of the scene – an explicit labeling of every pixel, either as background or as participating in one or more particular hierarchies, each of which instantiates a brick of a specific type. From  $I$  one reads off the locations and identifications of license plates, strings of characters, characters, lines, parts of characters, etc., as they may be found throughout the image. In short,  $I$  represents a semantic and syntactic parsing of the scene with respect to the variables embodied in the composition machine.

Ideally we would make exact computations under  $P(I|\vec{y})$ . Perhaps we would compute the probability that a scene contains a license plate along with the most likely reading of the plate, or perhaps we would compute the most likely parse of the entire scene. Unfortunately, these and other functionals of the posterior distribution are intractable (indeed, NP-hard). We are forced to explore the computationally feasible alternatives.

Motivated by the observation that the states of lower-level bricks (e.g. terminal bricks) represent coarse hypotheses in the set  $\mathcal{I}$  (the set of all interpretations)<sup>3</sup>, and the good computational performance of coarse-to-fine (ctf) vision systems [1, 40, 44], not to mention the *optimality* of ctf search under some conditions [6], we have explored parsing methodologies that start with a bottom-up pass for indexing into a set of likely interpretations.

Consider again the license plate application and the composition machine depicted in Figure 2. In the bottom-up pass the computation is launched by evaluating, via a local likelihood ratio test, the evidence for every state of every terminal brick. Each state signals a part at a particular location. A threshold is adjusted so that very few, if any, actual parts are missed, resulting in a large number of “false positive” detections of parts of characters and plate boundaries.

This same sequence of likelihood ratio tests and conservative thresholds can then be used, in turn, to elicit possible activities among next-level bricks, this time based upon the already-computed collection of possible terminal-brick states. Recursive, bottom-to-top, application of the procedure generates a large list of possible parts and objects, each corresponding to a consistent subgraph within the compositional architecture, and each equipped with a measure of

<sup>3</sup>A given low-level brick participates (is active) in many more interpretations than a given high-level brick.

fitness based on a likelihood ratio. The list includes local interpretations that are largely redundant, differing only in the fine detail of positioning, as well as others that are mutually inconsistent. This is the index set, a set of candidate parts and objects that we next employ in a simple greedy algorithm to compute a full-blown parse.

The best candidate, as measured by likelihood ratio, is selected to seed the parse. Conditioned on the selected candidate (which itself is a sub-graph in the compositional architecture, and hence a parse), we choose next, from all consistent candidates in the remaining index set, the one that most increases the likelihood of the parse when combined with the already-selected candidate. The pair of consistent sub-graphs defines a new parse with higher likelihood. The *list optimization* procedure continues until there are no further additions from the index set that improve likelihood. The process can be repeated,  $n$  times, by seeding the  $k$ 'th parse with the  $k$ 'th best candidate from the index set, and finally choosing the best (most likely) parse among the  $n$ .

### 3. Demonstration: Reading License Plates

The approach is Bayesian: given an image, and given a composition machine with the semantic variables listed in Figure 2 and the architecture outlined in §2, we look for a high-likelihood interpretation,  $I \in \mathcal{I}$ . The presence and identity of license plates are then read off by visiting top-layer active bricks (see Figure 2) and their instantiations (subtrees) – which, in particular, include the license-plate numbers.

**Data.** A set of 458 images (each containing  $494 \times 652$  pixels) were collected and supplied by the Visics Corporation. Some images contained plates from other states with other fonts and syntaxes, and in some cases the entire plate was not imaged. The experiments were confined to the 385 images that contained human-readable standard-syntax Massachusetts license plates. A typical image is shown in Figure 5, and a collage of plates from multiple images is shown in Figure 4. There is only a small amount of rotation and variation in scale across the image set.

**Performance.** Interpretations commonly include character parts, characters, and even strings of characters, at multiple locations throughout the scene. The bottom panel of Figure 5 shows the top 25 “objects” (complete subtrees) that participate in a full-blown parse of the top-panel image. The full parse includes hundreds to thousands of additional annotations. Nevertheless, the system, starting with its first implementation and including the implementation reported on here, has never produced a false detection at the license-plate level. This is regardless of whether it is run on scenes with multiple plates (as in Figure 6) or no plates at all. The reading rate for characters contained in the license-plate ID's is about 99.5%, and above 98% for the ID's themselves

(an ID is misread if any of its characters are misread).

**Search Strategies.** Bottom-up seeding, as described in §2 is slow, even if candidates are heavily pruned during the bottom-up (indexing) pass. Although it is indeed a coarse-to-fine exploration of  $\mathcal{I}$ , the overwhelming majority of the calculations of likelihood are unnecessary in that they could be eliminated, before the fact, if the goal were to find instances of a particular object (e.g. find and read license plates).

These observations suggest a more efficient ctf strategy: traverse the bricks associated with the objects of interest (top-level bricks in the license-plate system). For each brick, perform a *depth-first* search for an instantiation. Lower-level bricks might be visited multiple times. Hence, for each brick, a list of instantiations is maintained and re-used every time that brick appears in the computation. Computation passes immediately to the terminal bricks, and the search remains coarse-to-fine in the sense discussed in §2. Yet many of the terminal bricks, indeed the vast majority, are never visited. Furthermore, the algorithm admits easily to multi-threading or implementation on a multi-processor system.

A simplified version of depth-first search was implemented. Top-level (license-plate) bricks are instantiated by a pair of bricks: a license-plate number (chosen from the penultimate layer) and a license-plate boundary (chosen from the third layer from the top). For each top-layer brick, the possible children among the license-plate boundaries were first explored. Although there were some false-positive boundaries (one is seen in Figure 6), only a small fraction of the image needed to be further explored for the corresponding license number. The result was a many-fold improvement in computation speed with no loss in performance. It is likely that a fully implemented depth-first search would further improve computational efficiency.

**Observations.** How important is the non-Markovian perturbation? It is straight-forward to run the composition machine with and without the perturbation term. What is more, the states of intermediate bricks signal detections of intermediate structures (such as characters, strings, and boundaries), and can therefore be assessed, in and of themselves, by their recognition performance.

We consistently find a substantial drop in performance, at all levels of recognition, from characters up to license plates, when running the Markov backbone in place of the full compositional (non-Markovian) system. For example, although we have not run the full data set under the Markov backbone, a random sample points to a substantial drop in detection performance, from the current 98+% of correctly read plates to something closer to 90%, as well as the appearance of some false detections at the license-plate level.

A different kind of experiment bears on the justification of hierarchical structure, *per se*. As formulated in §2, an in-

terpretation amounts to an annotation of a scene in terms of a multitude of parts and objects. (See Figure 5 for the top 25 parts and objects participating in a particular interpretation.) Consider now a highly simplified version of the license-plate composition machine, consisting of only the bottom two layers. The system can be used to detect characters in images. An alternative use of the character models embodied in the compositional structure would be to test at each location for the presence of a particular character, *against the alternative that neither the character nor any part of the character is present*. In other words, an all-or-none test instead of a test for character against the *compound* alternative of background *or* part(s). (We restrict ourselves to the character layer because the all-or-none test is nearly computationally prohibitive.) We find that recognition performance, as measured for example by the ROC curve, suffers substantially when we force an all-or-none decision. We will have more to say about this observation shortly.



Figure 4. Extracted plate region of sample images

## 4. Concluding Remarks

The machine was “built by hand,” but possibly some of it could be inferred directly from data. For example, it is not difficult to imagine parameter estimation schemes that would employ labeled or unlabeled data to statistically adjust the brick-based probabilities ( $\epsilon_0^\beta, \epsilon_1^\beta, \dots, \epsilon_{n\beta}^\beta$  – see §2), or the relational distributions ( $p_\beta^c$  and  $p_\beta^0$ ) that govern the attribute likelihood ratios. On the other hand, learning the architecture itself, including the selection of bricks, children sets, and attribute functions, is an enormously challenging problem. We have little to say on this matter except to speculate that such a system would probably have to be inferred bottom-up, one layer at a time, perhaps based upon the principle of “suspicious coincidences” articulated by Barlow in his theory of unsupervised learning [3].

We believe that there is an important connection between reusability and the persistent gap between human and machine performance in vision. As every practitioner knows, the computer vision problem would be far easier if “background” could be reasonably modeled as some kind of sim-



Figure 6. Test image and its parse with license and boundary objects

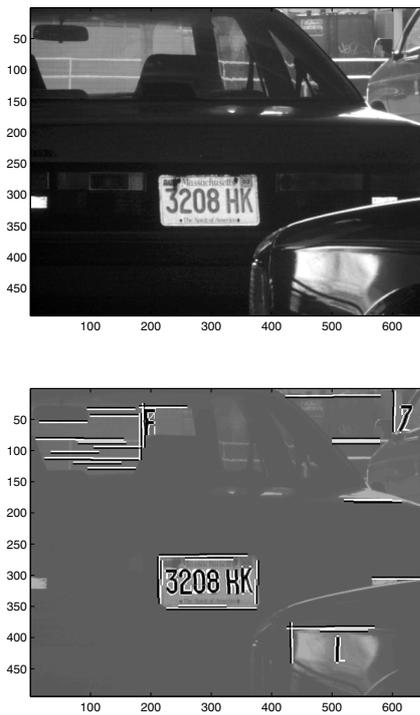


Figure 5. A typical image and a parse with its top 25 objects (Note: the top object is the license plate, followed by L-junctions, lines and false positive characters)

ple, stationary, stochastic process, such as white noise or a simple Markov random field. Instead, real scenes are typically filled with structure, and structured backgrounds have a way of conspiring to look surprisingly like the objects of interest, at least as seen by artificial vision systems. We would argue that this is a manifestation of the compositional nature of the visual world, and that it is the source of the poor performance of artificial vision systems, relative to biological vision systems, when operating near the zero-missed-detection end of the ROC curve. Backgrounds and

foregrounds are made of the same stuff – the same reusable parts. This would suggest that false detections occur predominantly at locations that share parts with the objects of interest, and it would argue strongly for compositional scene interpretation, whereby these locations can be labeled as parts without forcing an artificial distinction between object and background.

Consistent with these observations, we have been careful to define an interpretation as any complete subgraph (see §2), including multiple trees rooted at multiple levels. As mentioned earlier, experiments that artificially limit interpretations to include, say, either a full character, on the one hand, or no part of a character on the other, result in inferior ROC performance. In essence, by building compositional representations for the objects of interest, we equip these same objects with effective background models, namely their proper subtrees.

**Acknowledgement.** We wish to thank the Visics Corporation for supplying the license-plate data, and Eric Hopkins, President of Visics, for a great deal of information on the state of the art in license-plate reading.

## References

- [1] Y. Amit, D. Geman, and X. Fan. A coarse-to-fine strategy for multi-class shape detection. *IEEE Trans. PAMI*, 2004.
- [2] Y. Amit and A. Troune. Pop: Patchwork of parts models for object recognition. Technical report, University of Chicago, 2004.
- [3] H. Barlow. What is the computational goal of the neocortex? In C. Koch and J. L. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 1–22. MIT Press, Cambridge, 1994.
- [4] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [5] E. Bienenstock, S. Geman, and D. Potter. Compositionality, mdl priors, and object recognition. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 838–844. MIT Press, 1997.

- [6] G. Blanchard and D. Geman. Hierarchical testing designs for pattern recognition. *Annals of Statistics*, 33:1155–1202, 2005.
- [7] E. Borenstein and S. Ullman. Class specific top down-segmentation. In *Proc. ECCV*, pages 110–122, 2001.
- [8] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, 1965.
- [9] D. Crandall, P. F. Felzenszwalb, and D. P. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Proc. CVPR*, pages 10–17, 2005.
- [10] R. Duda and P. Hart. *Pattern classification and scene analysis*. Wiley, New York, 1973.
- [11] W. Fleming and W. McEneaney. Deterministic and stochastic approaches to nonlinear filtering. *Math. Control Signals Systems*, 14:109–142, 2001.
- [12] J. Fodor and Z. Pylyshyn. Connectionism and cognitive architecture: a critical analysis. *Cognition*, 28:3–71, 1988.
- [13] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. *Intl. Jour. of Comp. Vis.*, 40:25–47, 2000.
- [14] Y. Freund and R. E. Schapire. Discussion of three papers regarding the asymptotic consistency of boosting. *Annals of Statistics*, 32(1), 2004.
- [15] K. S. Fu. *Syntactic Methods in Pattern Recognition*. Academic Press, New York, 1974.
- [16] K. Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1:119–130, 1988.
- [17] S. Geman. On the formulation of a composition machine. Technical report, Division of Applied Mathematics, Brown University, 2005.
- [18] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1991.
- [19] S. Geman and M. Johnson. Probability and statistics in computational linguistics, a brief review. In M. Johnson, S. Khudanpur, M. Ostendorf, and R. Rosenfeld, editors, *Mathematical foundations of speech and language processing*, volume 138, pages 1–26. Springer-Verlag, 2003.
- [20] S. Geman, K. Manbeck, and E. McClure. Coarse-to-fine search and rank-sum statistics in object recognition. Technical report, Division of Applied Mathematics, Brown University, 1995.
- [21] S. Geman, D. F. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, LX:707–736, 2002.
- [22] U. Grenander. *Abstract Inference*. Wiley, New York, 1980.
- [23] G. E. Hinton, Z. Ghahramani, and Y. W. Teh. Learning to parse images. In *NIPS 12*, pages 463–469. MIT Press, 2000.
- [24] S. Krempp, D. Geman, and Y. Amit. Sequential learning of reusable parts for object detection. Technical report, Johns Hopkins University, 2003.
- [25] P. Laplace. *Essai philosophique sur les probabilités*. New York, 1965. Translation of Truscott and Emory.
- [26] E. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, Inc., San Francisco, 1975.
- [27] D. Lowe. Object recognition from local scale-invariant features. In *In Proc. ICCV*, pages 1150–1157. IEEE Press, 1999.
- [28] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.
- [29] M. Riesenhuber and T. Poggio. Models of object recognition. *Nature Neuroscience*, 3 supp.:1199–1204, 2000.
- [30] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14(3):1080–1100, 1986.
- [31] S. Roth and M. Black. Fields of experts: A framework for learning image priors. *IEEE Conf. on CVPR*, II:860–867, 2005.
- [32] D. Sheinberg and N. Logothetis. Noticing familiar objects in real world scenes: The role of temporal cortical neurons in natural vision. *Journal of Neuroscience*, 21:1340 V1350, 2001.
- [33] J. M. Siskind, J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. Spatial random tree grammars for modeling hierarchical structure in images. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 2004.
- [34] G. Socher, G. Sagerer, and P. Perona. Bayesian reasoning on qualitative descriptions from images and speech. *Image and Vision Computing*, 18:155–172, 2000.
- [35] M. Stone. Cross-validators choice and assessment of statistical predictors (with discussion). *J.R. Statist. Soc.*, B36:111–147, 1974.
- [36] A. J. Storkey and C. K. I. Williams. Image modeling with position-encoding dynamic trees. *IEEE PAMI*, 25(7):859–871, July 2003.
- [37] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. CVPR*, pages 762–769, 2004.
- [38] Z. W. Tu, X. R. Chen, Y. A. L., and S. C. Zhu. Image parsing: unifying segmentation, detection and recognition. *Intl J. of Computer Vision*, 2005.
- [39] S. Ullman. Sequence-seeking and counter streams: A computational model for bi-directional information flow in the visual cortex. *Cerebral Cortex*, 5:1–11, 1995.
- [40] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5:1–6, 2002.
- [41] D. Van Essen, C. Anderson, and D. Felleman. Information processing in the primate visual system: an integrated systems perspective. *Science*, 255:419–423, 1992.
- [42] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.
- [43] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, 1995.
- [44] P. Viola and M. J. Jones. Robust real time object detection. *Intl. Jour. Comp. Vis.*, 2002.
- [45] G. Wahba. A survey of some smoothing problems and the method of generalized cross-validation for solving them. In P. Krishnaiah, editor, *Applications of Statistics*, pages 507–523. North Holland, 1977.
- [46] S. Zhu and X. Liu. Learning in gibbsian fields: How accurate and how fast can it be? *IEEE PAMI*, 24:1001–1006, 2002.