Composition Systems

Stuart Geman, Daniel F. Potter, Zhiyi Chi
Division of Applied Mathematics
Brown University
Providence, Rhode Island 02912

[1]Supported by Army Research Office contract DAAH04-96-1-0445, National Science Foundation grant DMS-9217655, and Office of Naval Research contract N00014-97-0249.

1 Introduction

Compositionality refers to the evident ability of humans to represent entities as hierarchies of parts, with these parts themselves being meaningful entities, and being reusable in a near-infinite assortment of meaningful combinations. Compositionality is generally considered to be fundamental to language (Chomsky [7], [8]), but many believe, as do we, that it is fundamental to all of cognition. Objects and scenes, for example, decompose naturally into a hierarchy of meaningful and generic parts. Furthermore, compositions help us to identify parts unambiguously: It is often the case that components can not be correctly interpreted in the absence of the contextual constraints imposed by their incorporation into a larger whole, i.e. a composition. Indeed, such compositions are sometimes called "higher-level constraints."

It has been argued that artificial neural networks, by virtue of their ability to *learn by example*, reasonably approximate the workings of natural neural networks. But as pointed out by Foddor and Pylyshyn ([15]), these artificial networks are not compositional, and therefore they fail to mimic a basic attribute of human cognition. (See, however, von der Malsburg [39], Smolensky [38], Prince and Smolensky [32], Bienenstock [2], Hummel and Biederman [24], and Mjolsness [28] for efforts to address compositionality within a neural network framework.)

As early as 1812 Laplace discussed the compositional nature of perception: In his Essay on Probability ([26]), he remarks on one's overwhelming preference to interpret the string CONSTANTINOPLE as a single word, rather than a collection of fourteen letters. In some sense, it is "more probable" that the letters came together in the context of a known word than that they found their placements by coincidence. Of course the Gestalt psychologists were getting at very much the same thing (cf. [11]), as are today's cognitive scientists studying modern compositionality (see, especially, the work by Feldman [14], which connects closely with the development here).

The purpose of this report is to propose a mathematical formulation of compositionality. Inspired by Rissanen's Minimum Description Length Principle ([33]), a probability will be devised that promotes a recursive grouping, or composing, of constituents.

A primary goal is to make a contribution to machine vision: We believe that this formulation can be a basis for building vision systems that system-

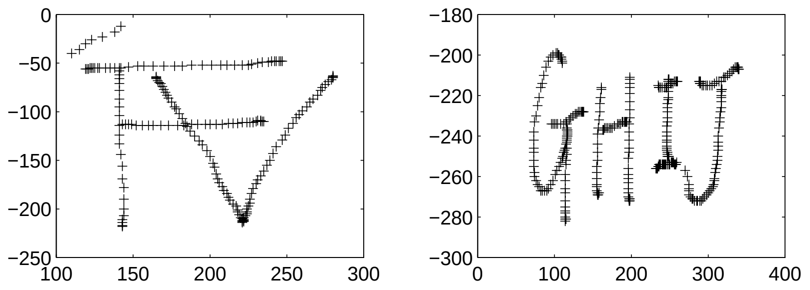
Figure 1: On-line images. Stylus position is sampled at regular intervals. Sampled locations are indicated with "+" symbol.

atically exploit contextual constraints, and thereby address the many levels of ambiguity that arise in image interpretation. Many others have taken a similar approach for similar reasons—see, for example, Narasimhan ([29]), Shaw ([36]), Pavlidis ([30]), Fu ([17]), Biederman ([1]), Grenander ([18]), and Casadei & Mitter ([5]).

2 Example: Experiments in On-Line Character Recognition

In way of introduction, we shall first examine the basic ideas informally through a relatively simple (but nonetheless largely unsolved) application: on-line upper-case character recognition.

Figure 1 shows some simple images of the type that we wish to interpret. Strokes and characters are drawn on a pad with a stylus whose position is sampled at a constant rate. The markings in Figure 1 represent the locations of sampled points. Of course there is order information, and this can be quite useful, but for the purposes of this illustration the order information will be ignored: The data is simply the collection of sampled locations.

As a first step, we will need to develop hierarchical representations for objects in the object library. The library will certainly include the upper-case letters, but in addition there are numerous other object types that will emerge from the intermediate-level representations, including, for example,

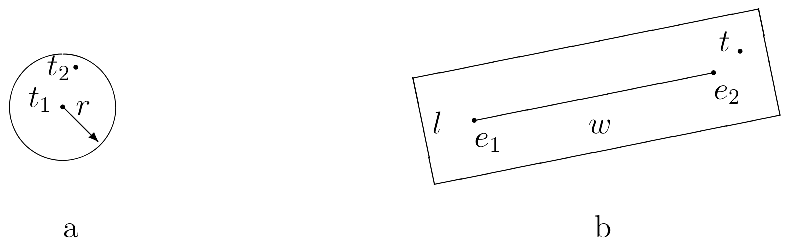

Figure 2: Syntactic constraints for two points forming a linelet (panel a) and a point joining a line to make a larger line (panel b).

"lines," "arcs," "T-junctions," and "L-junctions."

It might be expected that compositional hierarchies would be most conveniently defined via production rules within a formal grammar. But to the contrary, it turns out to be more convenient and more natural to come at this from the other direction, which is to say via *composition rules* rather than productions. Composition rules are syntactic rules under which entities are composed to form composite entities, very much like the process of *unification* in Unification Grammars ([37], [25]).

Recursive application of the composition rules defines the set of recognizable objects. The process is initiated with a "primitive" class of objects, which in this case is the set of individual points at which the stylus could be sampled. Let us suppose that the set of possible sampled locations consists of M^2 points arranged on an $M \times M$ grid. Let T be the subset of objects representing these M^2 primitives (so that each $t \in T$ is a particular location on the $M \times M$ grid).

A simple composition rule would allow two primitives to be composed into a kind of mini-stroke, which we might term a linelet: Given a radius r, two points, t_1 and t_2, can join if their distance does not exceed r. See Figure 2a.

What sort of compositions give rise to a straight line? A straight line could be grown by adjoining a single point (primitive) to either a linelet or to an already-existing straight line. Let λ be the linelet or the straight line which is to be bound to the primitive. The object λ itself comprises a set of primitives (just two, in the case of a linelet). Define e_1 and e_2 to be two points that achieve the maximum distance among pairs of points in this set, and let this distance be d. Fix two positive numbers w and l, and situate a rectangle of length $d+2l$ and width $2w$ symmetrically around the line segment joining e_1 and e_2 (refer to Figure 2b). Allow λ to bind to a primitive t provided that t is contained in this rectangle.

Composition rules can be added that allow two colinear straight lines to bind to form a larger straight line, or two straight lines to bind to form an L or a T junction. Linelets can be combined with primitives to form arcs, and arcs together with primitives, or arcs together with arcs, can form larger arcs. Xiaohua Xing, while a student in the Division of Applied Mathematics at Brown University, and Dan Potter ([31]) and Shih-Hsiu Huang ([23]), as described in their dissertations, have run on-line character recognition experiments. Compositional hierarchies involving dozens of rules were constructed, giving rise to the twenty-six upper-case characters as well as numerous intermediate object types, including primitives, straight-lines, various junction types, arcs, and so-on.

Any collection of composition rules together with the set T of primitives defines a set, or library, of objects, Ω. To make this precise it is necessary to interpret objects as trees in which the leaves are primitives, and in which each non-leaf node is labeled with an object type (linelet, line, etc.). The label of the tree itself (i.e. the object type) is the label of its root node. If, for example, the object arose from the rule *straight line binds to straight line to form straight line*, then the root node and each of its daughters would be labeled "straight line," and the remaining interior nodes would either be labeled "straight line" or "linelet." The library Ω is the set of trees such that for each non-terminal node n with label l there exists a composition rule under which the daughters of n can bind to form an object of type l. The set T of primitives is viewed as a set of single-node objects: $T \subseteq \Omega$.

The set of objects is unimaginably large, even if we were to restrict ourselves to composition rules for just linelets and straight lines. Furthermore, given any collection of primitives that can be interpreted as a particular object with label "l" (in other words, the primitives constitute the terminal nodes of an object with label l), there will typically be a large number of distinct objects of the same type (same label) containing the same primitives. Because of this, in formal language theory, systems such as ours are termed "ambiguous." This may turn out to be a virtue: All of the many explanations which share a common root-node label are essentially equivalent, and therefore there are many computational paths to what amounts to a “correct” solution. This kind of redundancy may open the door to pruning, or coarse-to-fine, or other heuristic search methods. (But K.S. Fu, who pioneered syntactic pattern recognition, would probably disagree: in a book on the subject ([16], page 27) he writes: “In pattern description languages, it is clear that ambiguity should be avoided; therefore, to find a family of unambiguous grammars is a problem of interest in this area.”)

Within this framework, an “interpretation” is the assignment of each element of an image (in the present example, each primitive) to an object. One easy-to-compute interpretation simply labels each sampled point as a primitive; no aggregations, or compositions, are offered. This of course is not what we are after. In the left-hand panel of Figure 1, we would prefer to join the seven nearly-colinear points in the upper left region and label them, collectively, as a straight line segment. The evident tendency of humans to manufacture such compositions is of course the cornerstone of compositionality. (See Feldman, [12] and [13], for recent work making use of psychophysical and analytic tools to explore the aggregation process in human subjects.)

Aggregation is an instance of Occam’s Razor, and it can be formulated rather conveniently using Rissanen’s Minimum Description Length (MDL) Principle ([33]). The idea is to encode, for example in a binary code, each object hierarchy, as if it were to be transmitted over a channel or stored on a disk. A “sensible” encoding would assign shorter codes to intuitively-succinct descriptions, such as the description of the seven points in terms of a straight line segment versus their description as individual and independent locations. There is a more-or-less natural encoding induced by the hierarchical structure, and in this regard the use of composition rules instead of productions is a central feature of the approach. In particular, each rule can be appended with a formula for encoding the composition *in terms of* the already-encoded components; the encoding scheme is recursive. Let us put aside the general scheme and examine, instead, some specific instances based upon the composition rules defined earlier.

We suppose that there are L object types (primitives, linelets, straight lines, etc.) in our object library. For simplicity, we will assign a uniform encoding to the different object types, meaning that we will use $\log_2(L)$ bits to indicate an object label. (Bit counts will usually be fractions. These should be rounded, generally upward, but it is easier and more clear to just work with real numbers.) A specific instance of a primitive would be most naturally encoded with $2\log_2(M)$ bits, indicating the values of each of the two coordinates. (Recall that we are working on an $M \times M$ grid.) Thus a primitive encoding involves $\log_2(L) + 2\log_2(M)$ bits. Consider now a linelet. The label, "linelet," requires $\log_2(L)$ bits. Referring to Figure 2a, the "seed" point, t_1, requires $2\log_2(M)$ bits to specify (the label, "primitive," is now superfluous—linelets always consist of two primitives), and t_2, by virtue of its restriction relative to t_1, can be encoded with $\log_2(\pi r^2)$ bits (corresponding to—approximately—πr^2 allowed lattice locations). Thus a linelet is encoded with $\log_2(L)+2\log_2(M)+\log_2(\pi r^2)$ bits. There is a savings: coded separately, t_1 and t_2 would require a total of $2\log_2(L) + 4\log_2(M)$ bits, and πr^2 is of course substantially smaller than M^2.

The encoding of straight lines proceeds similarly, but in this case the labels of the constituents need to be specified. The first constituent could be a linelet or a line, and this specification will require one bit (still a saving over the $\log_2(L)$ bits associated with the unbound item). Similarly, if the first constituent is a line, then an additional bit is required to specify whether the second constituent is a primitive or itself a line. In either case, the position of the second constituent is constrained by the location of the first constituent. Hence there is a further savings over an independent encoding of the constituents.

In principle, the encoding of lines is recursive: When two straight lines are joined to form a straight line, the code of the composite embeds the codes of the constituents. Actually, however, a recursive form is difficult to construct. This will be discussed further in §4, both from the point of view of coding as well as a more traditional probabilistic viewpoint. (Of course, the two viewpoints are essentially equivalent if we adopt a Shannon code when given a probability distribution—see [10], or take code lengths as log-probabilities when given a code.) In any case, there are many details concerning the existence and scope of codes (and/or probability measures) satisfying such recursive relationships, extensions to nonuniform encodings of labelings, and so on. For now, we wish only to point out that compositional codes promote aggregation by assigning more succinct codes to compositions than to constituents, and that these codes give an explicit formula for evaluating competing interpretations as may be associated with either inconsistent aggregations or inconsistent labelings of a common region.

Recall that an "interpretation" is the assignment of each element of an image to an object. An *optimal* interpretation is an assignment that achieves the minimum total description length. We have experimented with a simple algorithm for computing an approximately optimal interpretation. Briefly, the algorithm proceeds in two steps: In the first step, the observed primitives are recursively aggregated under the composition rules. This creates a large collection of labels, with many contradictory and multiple coverings of the original image. Usually some sort of pruning, based upon description length, is used in order to maintain a manageable list size. In the second step, a greedy algorithm chooses a subset from this collection by choosing successively the next best labeling (shortest description length) among those not chosen, until the original image is entirely labeled. The greedy algorithm is fast, and can be restarted dozens or even hundreds of times, from different choices of the first label.

The algorithm is simple and easy to implement. There can be no doubt that more sophisticated search strategies will be needed for more complex applications. Nonetheless, systems based on this approach have been able to read overlapping and highly irregular characters, as demonstrated in experiments by Xiaohua Xing (see Figures 1 and 3), Dan Potter (see [31]), and Shih-Hsiu Huang (see [23]).

More levels of composition can be included in the hierarchy (again, see Potter, [31]). For example, under more-or-less straightforward composition rules, characters can be grouped to form strings. At this point, an on-line dictionary can be used to create thousands of virtual composition rules: strings can be viewed as specific words, with a saving of label bits accrued for each character. These high-level compositions can resolve ambiguities. In fact, many single-character confusions are impossible to resolve in isolation, but easily resolved in the context of words.

The MDL procedure is exactly Bayesian MAP: use code lengths as "energies" and use the associated Gibbs distribution as the prior. Among other advantages (see sections 4 & 5), the Bayesian viewpoint suggests the possibility of estimating (learning) composition costs. Consider, for example, the joining of two lines to form an L-junction. In principle, the distribution on the relations between end points of the two component lines could be estimated. The uniform encoding used in the examples discussed here could then be replaced by a Shannon code associated with the estimated distribution—atypical joinings would then be appropriately penalized with long code words. As we shall see, distributions governing relationships among constituents will emerge naturally from a probabilistic formulation.

The idea of using description lengths to define an energy functional, and

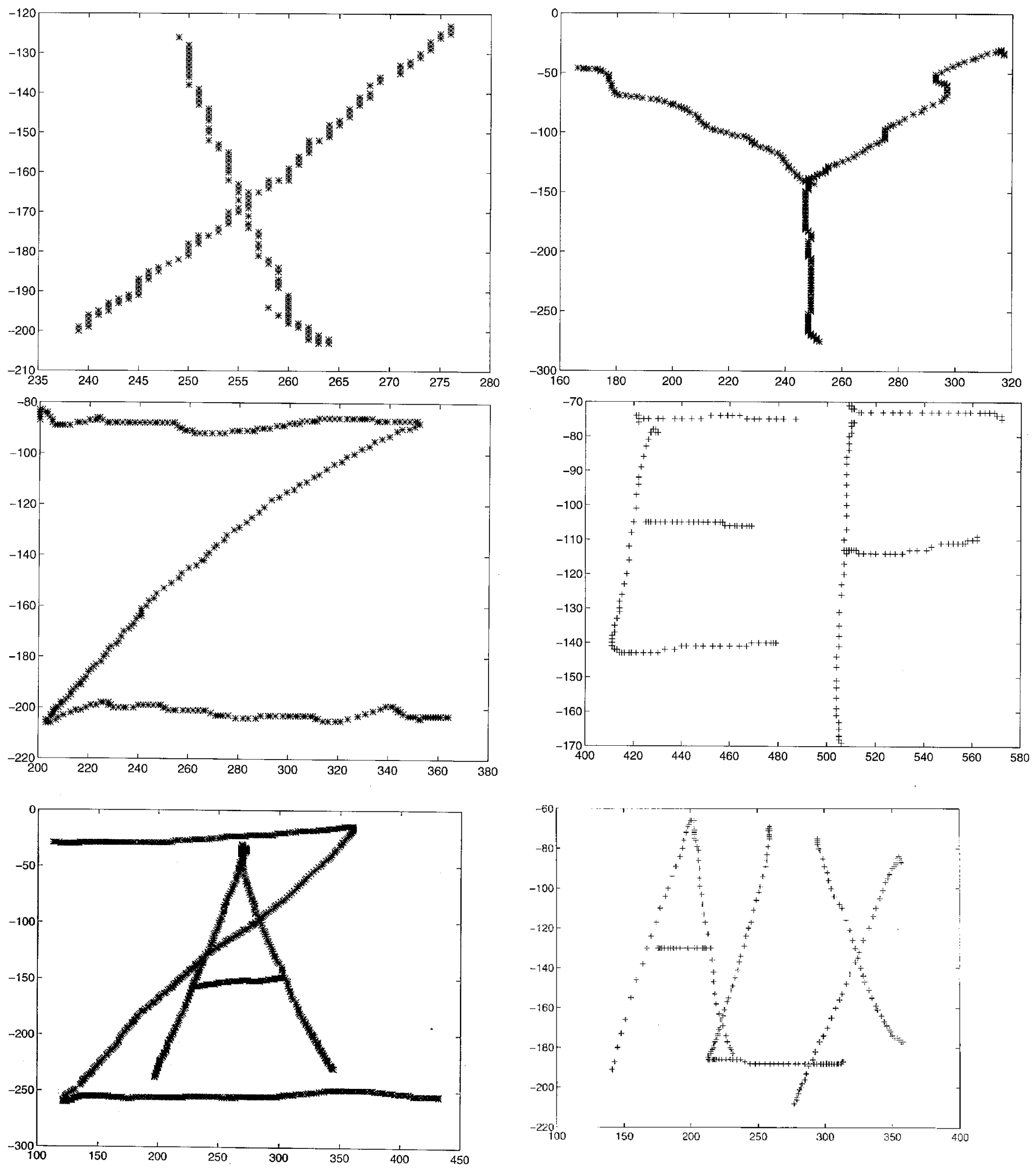

Figure 3: Examples of images interpreted by on-line character recognition algorithm.

thereby a prior for Bayesian inference, is not new to machine vision. In one form or another the "MDL Principle" has been applied to image segmentation (cf. Leclerc [27], Zhu and Yuille [41]), image restoration (cf. Saito [34]), motion analysis (cf. Schweitzer [35], Gu et al. [19]), and image interpretation (cf. Canning [4], Hinton et al. [21]). Our approach is in the same spirit as these, although the emphasis here is on compositionality, very much along the lines proposed by Cooper (see [9]): We will use description lengths to guide the development of distributions that promote hierarchical aggregations of parts.

3 Objects and the Rules of Composition

Following the example of the previous section, the set of objects is defined as the closure of a given set of primitive objects under a set of composition rules. The composition rules are defined in terms of attributes of the constituents. The purpose of this section is to make this construction precise and to discuss some of its properties. In analogy to formal language theory, the object class defines a language on strings of primitives—the scope of languages attainable from such constructions will also be discussed.

3.1 Labeled Trees

Objects that arise from composition rules are most naturally described as trees. The terminal nodes are primitives, corresponding to the most elementary constituents that participate in an object hierarchy. Let us denote by T the set of terminal or primitive objects. A 256×256 grid of possible pen locations corresponds to $T = \{0, 1, ...255\} \times \{0, 1, ...255\}$. Different applications will dictate different terminal sets. A set of line segments may be convenient when modeling polygonal regions in the plane or for approximating regression lines; a lexicon, or set of words would be most appropriate when fitting a grammar to a corpus of phrases and sentences. Depending upon the application, T may be discrete or continuous.

In addition to the primitives, there are other, "compound," objects with "labels" or "types" such as 'linelet,' 'line,' 'letter_A,' and so-on. Let N represent, abstractly, the set of labels. N is discrete but not necessarily finite. Formally, an object is a *labeled tree*, where

Definition. The set of labeled trees, Θ, is the set of directed tree graphs

(a) that are planar[1], finite, and connected;

(b) that have a single root with edges directed away from the root and towards the leaves;

(c) and in which every non-leaf node is labeled with an element of N (nonterminal) and every leaf node is labeled with an element of T (terminal).

Remarks.

1. Singletons are labeled from elements of T, so in this sense $T \subseteq \Theta$.

2. By the *label of the tree* $\omega \in \Theta$ we will mean the label of its root node. We use $L(\omega)$ $(L : \Theta \to T \bigcup N)$ to represent the label of ω.

3. If $\omega \in \Theta$, $\omega \notin T$, we will write $\omega = l(\alpha_1, \alpha_2, \ldots \alpha_n)$ when $L(\omega) = l$, and $\alpha_1, \alpha_2, \ldots \alpha_n \in \Theta$ are, respectively, the left-to-right daughter subtrees of ω $(1 \leq n < \infty)$.

4. The ordering of daughter nodes is distinguished. So, for example, $l(\alpha, \beta) \neq l(\beta, \alpha)$ unless $\alpha = \beta$.

3.2 Attributes and Composition Rules

Not all elements of Θ are objects. Objects are distinguished by being consistent with a set of *composition rules*, which is to say that the daughter subtrees of each non-terminal node represent an *allowed* composition. Let Θ^* denote the set of finite *nonempty* strings of labeled trees.[2] We will use starred letters, α^*, to designate generic elements of Θ^*: $\alpha^* \in \Theta^* \Rightarrow \alpha^* = \alpha_1, \ldots \alpha_n$ for some $n \geq 1$ and $\alpha_1, \ldots \alpha_n \in \Theta$.

[1] *Planar* so that we can distinguish the left-to-right order of daughter nodes.

[2] Given a set S, we will use S^* to represent the set of finite *nonempty* strings of elements of S. This is nonstandard—usually S^* includes the empty string.

Definition. A **composition rule** for the label $l \in N$ is a pair $(B_l, \mathcal{S}_l)$ where B_l, the *binding function*, maps Θ^* into an arbitrary *range space*, $\mathcal{R}_l$:

$$B_l : \Theta^* \to \mathcal{R}_l,$$

and $\mathcal{S}_l$, the *binding support*, is a distinguished subset of $\mathcal{R}_l$, $\mathcal{S}_l \subseteq \mathcal{R}_l$.

Composition rules dictate legitimate bindings through $\mathcal{S}_l$, the "allowed" values of B_l: $\alpha_1, \ldots \alpha_n$ can bind to form $l(\alpha_1, \ldots \alpha_n)$ if $B_l(\alpha_1, \ldots \alpha_n) \in \mathcal{S}_l$.

Some compositions are monadic: $\omega = l(\alpha)$. As an example, α may represent a string of letters, such as '**CAT**', whereas ω may represent the word *'cat'*. Thinking again for a moment in terms of description lengths, the 'composition' $\alpha \to \omega$ saves bits: The label *'cat'* removes the need to encode the labels of the particular letters constituting the string α.

We can always take $\mathcal{R}_l = \{0, 1\}$, $\mathcal{S}_l = \{1\}$, and $B_l(\alpha_1, \ldots \alpha_n) \in \{0, 1\}$ with $B_l(\alpha_1, \ldots \alpha_n) = 1$ indicating that $\alpha_1, \ldots \alpha_n$ can be bound to form $l(\alpha_1, \ldots \alpha_n)$. On the other hand, our primary goal is to put probabilities on Ω (see §4). This will be done via certain empirical distributions, such as the distribution on the angle between two lines that are bound and labeled "right angle," or on the distance between two points that are bound and labeled "linelet." In these cases B_l is chosen to explicitly restrict the attribute or attributes of interest (e.g. the angle between lines or the distance between points), $\mathcal{S}_l$ defines allowable values under the composition, *and a distribution imposed on $\mathcal{S}_l$ reflects the likelihood of particular configurations of the composition.*

Often there is more than one attribute involved in a binding rule (labels, positions, *and* sizes of letters for a rule of the form *'letter'* + *'letter'* $\to$ *'string of letters'*), in which case it is most natural to represent B_l as a vector, possibly having both continuous and discrete components. This is the reason for declaring $\mathcal{R}_l$ to be arbitrary, in the general set up.

As a specific example, a composition rule for linelets (labeled '1' say) might be based upon the binding function

$$B_1(\alpha_1, \ldots \alpha_n) = \begin{cases} \text{distance}(\alpha_1, \alpha_2) & \text{if } n = 2 \text{ and } \alpha_1, \alpha_2 \in T \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

with range $\mathcal{R}_1 = R$ and support $\mathcal{S}_1 = (0, r]$, for some maximum radius r. Of course we could as well take

$$B_1(\alpha_1, \ldots \alpha_n) = \begin{cases} +1 & \text{if } n = 2,\ \alpha_1, \alpha_2 \in T, \text{ and distance}(\alpha_1, \alpha_2) \leq r \\ -1 & \text{otherwise} \end{cases}$$

by defining $\mathcal{S}_1 = \{1\}$; the representation is certainly not unique. On the other hand, if we wish to develop distributions that promote small distances between the constituents of a linelet, while not necessarily excluding large distances, then the first representation (1) is clearly preferable (see §4).

As another example, consider a rule that allows a line or linelet to combine with a line or linelet to form a line. Let $(\vec{e}_1, \vec{e}_2)$ and $(\vec{e}_3, \vec{e}_4)$ be the endpoints, respectively, of the first and second constituents. Define

$$d = min\{distance(\vec{e}_1, \vec{e}_3), distance(\vec{e}_1, \vec{e}_4), distance(\vec{e}_2, \vec{e}_3), distance(\vec{e}_2, \vec{e}_4)\}$$

and define θ to be the angle (in $[0, 2\pi]$) between the two line segments. (If for example the minimum is achieved at $(\vec{e}_1, \vec{e}_3)$—breaking ties systematically when the minimum is not unique—then θ is the angle between $\vec{e}_1 - \vec{e}_2$ and $\vec{e}_4 - \vec{e}_3$.) Let '2' be the line label. If α is a line or linelet, and β is a line or linelet then define $B_2(\alpha, \beta) = (d, \theta)$. Otherwise, define $B_2(\alpha_1, \ldots \alpha_n) = -1$. In this case, B_2 maps Θ^* into $\mathcal{R}_2 = R^2 \bigcup \{-1\}$; now take, for example, $\mathcal{S}_2 = (0, c] \times [\pi - \Delta, \pi + \Delta]$, representing the tolerances on end-point proximity and colinearity. (A better, *scale invariant*, way to do this is to introduce the coordinates of $(\vec{e}_3, \vec{e}_4)$ relative to $(\vec{e}_1, \vec{e}_2)$, normalized to make $distance(\vec{e}_1, \vec{e}_2)$ a unit length. In this system, $\mathcal{R}_2 \subseteq R^4$, and $\mathcal{S}_2$ is a union of four regions which are related by translation. See Potter [31] and Huang [23] for more on using relative coordinates and building scale-invariant binding rules.)

Of course the range of B_2 can be extended to allow other compositions that might define lines, such as a line or linelet paired with a single point, or multiple (three or more) lines composed into a larger line.

3.3 Objects and the Scope of Composition Systems

Objects result from the recursive application of composition rules, starting with the terminal set:

> **Definition.** Given a terminal set T, a set of nonterminals, or labels, N, and a set of composition rules $(B_l, \mathcal{S}_l)$, one for each

label $l \in N$, the set of **objects** Ω is the closure of T under $\{(B_l, \mathcal{S}_l)\}_{l \in N}$ in Θ.

Remarks.

1. Let Ω^* denote the set of finite nonempty strings of elements from Ω. Then, equivalently, $\omega \in \Theta$ is an object ($\omega \in \Omega$) if and only if either $\omega \in T$ or $\omega = l(\alpha^*)$ where $\alpha^* \in \Omega^*$ and $B_l(\alpha^*) \in \mathcal{S}_l$.

2. Composition rules may or may not allow overlap (sharing) of primitives. So, for example, it can happen that $B_l(\alpha, \beta) \in \mathcal{S}_l$, and some of the terminal nodes of α are assigned the same primitive (label in T) as some of the terminal nodes of β. This would certainly be the case if, for example, $T = \{0, 1\}$ and we were interested in sentences of finite binary strings, $\{0, 1\}^*$.

What subsets of Θ can constitute a set of objects? It is certainly necessary that $T \subseteq \Omega$ and that $l(\alpha^*) \in \Omega \Rightarrow \alpha^* \in \Omega^*$. These are also sufficient conditions: Given such an Ω, define $\mathcal{R}_l = \{0, 1\}$, $\mathcal{S}_l = \{1\}$, and define $B_l(\alpha^*) = 1$ iff $l(\alpha^*) \in \Omega$. Ω is then the closure of T under these composition rules.

Obviously, these structures are very general. In line with this generality we will call such a system (consisting of T, N, & $\{(B_l, \mathcal{S}_l)\}_{l \in N}$) a *composition system* or *composition grammar*. Any language is attainable from a composition system, in the following sense: Let $\mathcal{L}$ be an arbitrary (nonempty) subset of T^* (*nonempty* finite strings of terminals). Then there are composition systems such that the set of ordered terminal strings, read left to right, corresponding to objects with some label, say l, is exactly $\mathcal{L}$. This is trivial: Include the composition rule $\mathcal{R}_l = \{0, 1\}$, $\mathcal{S}_l = \{1\}$, and $B_l(\alpha^*) = 1$ iff $\alpha^* \in \mathcal{L}$; B_l is an "acceptor" for the language $\mathcal{L}$. (This is not a particularly *useful* grammar—the identification of an object of type l is an entirely global affair, and hence does not lend itself to efficient computation. What we are really after is composition systems in which objects are *built up*, in more or less *small steps*.)

This generality may appear to be a weakness—what can be *usefully* said of such a general class? In the next section we will argue that the simplicity (generality) of the composition rules suggests a natural (and we hope compelling) mechanism for fitting composition systems with probability distributions in terms of certain "observable" frequencies.

Our object set Ω resembles what Grenander terms the "Configuration Space" in his General Pattern Theory (cf. [18]). In Pattern Theory discrete entities called generators can be composed provided that a "bond relation" is satisfied. The bond relation is a Boolean function of certain generator attributes known as "bond values." A configuration is a collection of generators together with a set of "true" bond relations among pairs of the generators. Composition rules, on the other hand, are more global: Trees are composed, and the composition rules can reference arbitrary details within the constituent subtrees. Whereas this allows us to express more general constituent relationships, it also complicates the development of a probability measure. In the Pattern Theory, the local binding structure leads to a Markov property which is heavily exploited in parameter estimation and in all computational aspects of the theory. In contrast, the probability developed here (§4) has no non-trivial Markov property.

Among the simplest syntactic systems are the context-free grammars. When does a composition system look like a context-free grammar? In general, the rules of composition can reference any aspects of the constituent trees—the binding functions are arbitrary. On the other hand, if binding depends only on the *labels* (roots) of the constituents, then a composition system (with finite T and N) is basically a context-free grammar. This is easy to see: If $B_l(\alpha_1, \ldots \alpha_n) = (L(\alpha_1), \ldots L(\alpha_n))$, then define a set of production rules on N according to the recipe

$$l \rightarrow l_1, \ldots l_n \in \{T \cup N\}^* \Longleftrightarrow \exists \omega \in \Omega \ni w = l(\alpha_1, \ldots \alpha_n)$$

where for each $k = 1, 2, \ldots n$, $l_k = L(\alpha_k)$. Evidently, these rules are context-free, and they produce exactly Ω.

Another way to get a context-free grammar, without restricting binding to depend only on labels, is to introduce a sufficiently rich set of "attributes"—functions of objects that contain the essential information for binding. The key condition is that the attributes of a composition must be computable from the attributes of its constituents: Call $\mathcal{A} : \Omega \rightarrow R$ an attribute function if $|\mathcal{A}(\Omega)| < \infty$ (finite number of attribute values), and if $\forall l \in N \; \exists A_l : R^* \rightarrow R$ such that

$$\omega = l(\alpha_1, \ldots \alpha_n) \Rightarrow \mathcal{A}(\omega) = A_l(\mathcal{A}(\alpha_1), \ldots \mathcal{A}(\alpha_n))$$

where $R^* = \cup_{k=1}^{\infty} R^k$.

Proposition 3.1 *Let T, N, $\{B_l, \mathcal{S}_l\}_{l \in N}$ be a composition system, $\mathcal{C}$, with $|T| < \infty$, $|N| < \infty$, and attribute function $\mathcal{A}$. If the binding rules depend only on attribute values,*

$$B_l(\alpha_1, \ldots \alpha_n) = B_l(\mathcal{A}(\alpha_1), \ldots \mathcal{A}(\alpha_n)),$$

then there is a context-free grammar that produces the same yield (set of left-to-right ordered terminal strings) as $\mathcal{C}$.

Proof. Define a context-free grammar $G = (V, T, \mathcal{P}, S)$ with arbitrary "start symbol" $S \notin \mathcal{A}(\Omega)$, nonterminal symbols $V = S \cup \mathcal{A}(\Omega)$, terminal symbols T, and production (rewrite) rules $\mathcal{P}$:

$$\begin{array}{ll} S \to k & \forall k \in \mathcal{A}(\Omega), \\ & \\ k \to k_1, \ldots k_n & \text{if } \exists \omega \in \Omega, \omega = \mathrm{l}(\alpha_1, \ldots \alpha_\mathrm{n}) \ni \\ & k = \mathcal{A}(\omega) \ \& \ k_i = \mathcal{A}(\alpha_i), 1 \leq i \leq n, \\ & \\ k \to t & \text{if t} \in \text{T and k} = \mathcal{A}(\text{t}) \end{array}$$

Suppose g is a derivation tree in G. We will exhibit an $\omega \in \Omega$ whose terminal nodes coincide with the terminals in g. First we will associate every subtree of g that is rooted at some $k \in V$, $k \neq S$, with an object $\omega \in \Omega$ is such a way that $k = \mathcal{A}(\omega)$, and ω and the subtree rooted at k have the same yield (string of terminals). This is done recursively in a "bottom-up" sequence as follows:

If $k \to t \in T$ then the subtree rooted at k is associated with the object t; note that $k = \mathcal{A}(t)$. Now suppose $k \to k_1, \ldots k_n$, and the subtrees rooted at $k_1, \ldots k_n$ have been associated with $\alpha_1, \ldots \alpha_n \in \Omega$, respectively. Since $k \to k_1 \ldots k_n$, there exists $\omega' = l(\alpha'_1, \ldots \alpha'_n) \in \Omega$ such that $k = \mathcal{A}(\omega')$ and $k'_i = \mathcal{A}(\omega'_i)$, $1 \leq i \leq n$. Let $\omega = l(\alpha_1, \ldots \alpha_n)$. Note that $\mathcal{A}(\alpha_i) = k_i = \mathcal{A}(\alpha'_i)$, $1 \leq i \leq n$. Since $\omega' \in \Omega$, $B_l(k_1, \ldots k_n) \in \mathcal{S}_l$. Hence $\omega \in \Omega$, and $k = \mathcal{A}(\omega') = A_l(k_1, \ldots k_n) = \mathcal{A}(\omega)$. Thus we associate k with ω.

Eventually, the subtree rooted at k, where k is the output of the first production ($S \to k$), is associated with some $\omega \in \Omega$ in such a way that $k = \mathcal{A}(\omega)$ and the yield of ω matches the yield of k—and hence the yield of ω also matches that of g.

Now fix an arbitrary $\omega \in \Omega$. Associate every node of ω with a production from $\mathcal{P}$ as follows: If the node is a terminal, $t \in T$, associate it with $\mathcal{A}(t) \to t$. Otherwise, let $\alpha = l(\alpha_1, \ldots \alpha_n)$ be the subtree defined by the (nonterminal) node. Assign to the node the production $\mathcal{A}(\alpha) \to \mathcal{A}(\alpha_1), \ldots \mathcal{A}(\alpha_n)$, which exists since $\alpha \in \Omega$.

Finally, starting with $S \to \mathcal{A}(\omega)$ work down the tree ω using the above-defined productions at each node. The result is a parse tree in G with the same yield as ω. $\square$

Of course, even if a context-free representation exists, it may not be desirable. Sometimes, very large state spaces will be needed—to capture information about positioning and scale, or color, or style, to name just a few attributes that may be called upon in composition rules. It is unnatural to "load" these attributes into an extended list of nonterminal variables, and it pretty much negates one of the chief virtues of context-free systems: the existence of efficient dynamic programming algorithms for parsing. Furthermore, the resulting *probabilistic system*, which then necessarily involves distributions on very large production systems, may become unmanageable.

4 Recursive Encoding and its Probabilistic Interpretation

4.1 The MDL Viewpoint

We seek a formulation in which the interpretation of a collection of objects as a single composite object, when possible, is generally favored over the interpretation of these same objects as independent entities. This is a tenant of compositionality ("Occam's Razor"), and one way to formalize it is through Rissanen's "Minimum Description Length Principle" (as in §2).

By looking a little deeper at the implications of MDL for our vision problem, we will be led to a probabilistic formulation that accommodates a "natural" distribution on objects (Ω), and a resulting distribution on scenes (collections of objects) that indeed favors compositions.

Return to the example of on-line character recognition discussed in §2, and imagine that we have two lines, α and β, each consisting of a set of more-or-less colinear points sampled from the stylus trajectory. Then T, the primitives, is the set of (discrete) locations on the writing pad. Suppose that we have devised composition rules that allow us to combine points into linelets, and linelets and points into lines. Each rule comes equipped with a formula for encoding the resulting composition, along the lines of the examples worked out in §2. Thus there are binary codes, $c(\alpha)$ & $c(\beta)$, describing exactly the lines α & β.

Now consider a new composition rule, under which two appropriately-situated lines (α & β) can be bound to form a single, larger, line ($\alpha+\beta \rightarrow \omega = l(\alpha, \beta)$, $l \leftrightarrow$ "line")—see §3.2. The binding function B_l would be constructed to require that the constituents are nearly colinear, and "close"—an endpoint of α is in a neighborhood of an endpoint of β. Reasoning as in §2, encoding ω saves bits: the label of ω, "line", having been specified, restricts the possible labels of the constituents (α & β) to point, linelet, or line. Thus the labels of the constituents require fewer bits (supposing that there is a large repertoire of possible labels). Furthermore, the endpoints of β, when β is viewed as an individual object, are essentially unconstrained; whereas, when β is viewed as a constituent of ω one of its endpoints is highly restricted relative to α and the other essentially loses one degree of freedom on account of the colinearity constraint. More bits are saved.

How many bits are saved? If $c(\omega)$ is the code assigned to ω, and if $|c|$ represents the length (number of bits) of a code c, then the bits saved in describing α & β as constituents of ω, instead of as separate entities, is $|c(\omega)| - |c(\alpha)| - |c(\beta)|$. What exactly is $c(\omega)$? Of course there are many ways to construct a code for ω. One might, for example, attempt to use the existing codes for α and β as a starting point, and then add and delete bits according to information that needs to be included (the label of ω, for example) or deleted (concerning the positioning of β relative to α, for example). Consider, specifically, a recoding of the positioning of β relative to α. *If* the existing code for β is organized so as to include an encoding of the endpoints of β and an encoding of the remainder of β in terms of the endpoints, then a recoding of β is straightforward: Recode the endpoints as positions *relative to* α, and leave in tact the remaining portions of the β code. More generally, the idea is to identify the attributes that are restricted by composition, and then to devise codes that factor into a code for these attributes followed by a "residual" code for the remainder of the object given the attribute values.

But this approach may not be workable. In general, it will be difficult to identify a sufficient set of attributes that anticipate every possible composition (cf. our earlier discussion of context-free composition systems). For example, what attributes would we identify for the letter "A"? Depending on the composition, we may appeal to size, shape, style, color, and/or stroke widths. We are, then, faced with our original problem, but two-fold: For a given composition, we will need to recode the remaining attributes based upon a restriction on a subset of attributes (say, given a restriction on stroke width), and we will need to understand how to develop a code for the remaining details of an "A" (i.e. the residuals) given the attribute values. There is, furthermore, the complication that we need to design the code of a composition (say the word "AT") in such a way that it too factors into attribute and residual subcodes, so as to anticipate *its* role as a constituent in still later compositions.

Alternatively, we could "start from scratch" with each composition: Recode the constituents in a manner that is natural and particular for the given composition rule. More specifically, we could build a compositional code around the binding functions B_l introduced in §3 by concatenating codes for: The label of the composite; the value of the binding function (evaluated at the constituents); and *a residual code for the constituents* given the binding function value. In essence, this is what we propose to do. Our approach, however, will be through *probabilities*—it is our claim that the notion of a "residual code" is most naturally formulated in terms of *conditional probabilities*, rather than more directly in terms of the actual code itself. Of course we can always connect probabilities and codes: Generate a Shannon code when given any assignment of probabilities (see [10], for example), or go from code lengths to "energies" (log probabilities) to probabilities when given an assignment of codes. In the latter case, the original code is a Shannon code for the derived probabilities. In the end, the approaches are equivalent.

4.2 Probabilities

4.2.1 From Context-Free Grammars, by Extension

Consider first a special case: Let $\mathcal{C}$ be a composition system with finite T (finite number of terminals) and finite N (finite number of labels), and with $B_l(\alpha_1, \ldots \alpha_n) = (L(\alpha_1), \ldots L(\alpha_n))$. In this case (see §3.3), $\mathcal{C}$ is context free and the business of equipping $\mathcal{C}$ with a probability is more-or-less standard fare: For each $l \in N$ introduce a *production probability distribution* Q_l:

$$Q_l : \{N \cup T\}^* \to [0,1], \qquad \sum_{l^* \in \{N \cup T\}^*} Q_l(l^*) = 1$$

Probabilities $P(\cdot|l)$ on each $\Omega_l \overset{\triangle}{=} \{\omega : L(\omega) = l\}$ are then defined recursively with the formula

$$P(\omega|l) = Q_l(L(\alpha_1), \ldots L(\alpha_n)) P(\alpha_1|L(\alpha_1)) \cdots P(\alpha_n|L(\alpha_n)), \tag{2}$$

when $\omega = l(\alpha_1, \ldots \alpha_n)$. (When $\alpha \in T$, $L(\alpha) = \alpha$, so that $P(\alpha|L(\alpha)) = 1$.)

To get to a single probability on all of Ω we will need to introduce, additionally, marginal probabilities on N and T (recall that $T \subseteq \Omega$). Then, when $\omega \in T$, $P(\omega) = Q(\omega)$, and when $\omega = l(\alpha^*)$, $P(\omega) = P(\omega|l)Q(l)$:

$$P(\omega) = \begin{cases} Q(\omega) \\ \qquad \omega \in T \\ Q(l)Q_l(L(\alpha_1), \ldots L(\alpha_n))P(\alpha_1|L(\alpha_1)) \cdots P(\alpha_n|L(\alpha_n)) \\ \qquad \omega = l(\alpha_1, \ldots \alpha_n) \end{cases} \tag{3}$$

So far we have the standard construction: Through a Markov property (the probability on each constituent given its label is independent of the remaining constituents), Q and $\{Q_l\}_{l \in N}$ induce a probability on Ω. This is the usual way to get probabilities on context-free grammars ([40]), or equivalently, on branching processes ([20]). What is the "right" construction when, more generally, $B_l(\alpha_1, \ldots \alpha_n) \neq (L(\alpha_1), \ldots L(\alpha_n))$? To get at this, rewrite (3) in terms of the *product measures* P^n on Ω^n: Given a measure P on Ω, define

$$P^n(\alpha_1, \ldots \alpha_n) = \prod_{k=1}^{n} P(\alpha_k),$$

and then define P^* on Ω^* by

$$P^*(A) = \sum_{n=1}^{\infty} P^n(A_n),$$

whenever $A = \bigcup_{n=1}^{\infty} A_n$, $A_n \subseteq \Omega^n$. If $\omega = l(\alpha_1, \ldots \alpha_n)$, then from (3),

$$P(\omega) = Q(l)Q_l(L(\alpha_1), \ldots L(\alpha_n))P(\alpha_1|L(\alpha_1)) \cdots P(\alpha_n|L(\alpha_n))$$

$$= Q(l)Q_l(L(\alpha_1), \dots L(\alpha_n))P^n(\alpha_1, \dots \alpha_n | L(\alpha_1), \dots L(\alpha_n))$$
$$= Q(l)Q_l(B_l(\alpha_1, \dots \alpha_n))P^n(\alpha_1, \dots \alpha_n | B_l(\alpha_1, \dots \alpha_n))$$

In summary:

$$P(\omega) = \begin{cases} Q(\omega) & \text{when } \omega \in T \\ \\ Q(l)Q_l(B_l(\alpha^*))P^*(\alpha^* | B_l(\alpha^*)) & \text{when } \omega = l(\alpha^*) \end{cases} \tag{4}$$

This points the way towards generalization: make sense of (4) when $B_l(\alpha_1, \dots \alpha_n) \neq (L(\alpha_1), \dots L(\alpha_n))$, and when T and N are not necessarily finite.

We will take up the task shortly, but it might be useful here to pause and return to the earlier question about bits gained. Suppose $\omega \in \Omega$ is a binary composition of constituents $\alpha, \beta \in \Omega : \ \omega = l(\alpha, \beta)$. How much more efficient is the interpretation of α and β as components of the composition ω then as independent (separate) entities? In a Shannon code, the length of a code word $c(\xi)$ is essentially $-\log_2 P(\xi)$. Hence, the composition *saves*

$$\log_2 P(\omega) - \log_2 P(\alpha) - \log_2 P(\beta) = \log_2 \frac{P(\omega)}{P(\alpha)P(\beta)}$$
$$= \ \text{(use 4)} \ \log_2 \left(Q(l) \frac{Q_l(B_l(\alpha, \beta))}{P \times P(B_l(\alpha, \beta))} \right)$$
$$= \log_2 Q(l) + \log_2 \left(\frac{Q_l(B_l(\alpha, \beta))}{P \times P(B_l(\alpha, \beta))} \right)$$

bits. The first term, $\log_2 Q(l)$, is negative and represents the (unavoidable) cost of coding the label, l, of ω. As for the second term, we expect that the observed value of $B_l(\alpha, \beta)$ will be far more likely in the context of the composition $\omega = l(\alpha, \beta)$ then if α and β were to be chosen independently under P:

$$\frac{Q_l(B_l)}{P \times P(B_l)} \gg 1$$

Imagine, for instance, that α and β are straight lines, that ω is an extended straight line with constituents α and β, and that B_l restricts α and β to be either linelets or lines and restricts their placements so as to form a composite line. Then it is quite clear that the likelihood of observing $B_l(\alpha, \beta)$ when in fact α and β are constituents of ω is far higher than "by chance"—i.e. then when α and β are placed independently according to the measure P.

4.2.2 Technical Foundation

The idea is to make sense of (4), in some generality, and to thereby define a measure on Ω. Ω is not necessarily discrete since T (the set of terminals) may be a continuous space. Furthermore, if $\mathcal{S}_l$ is also continuous, then we need to be careful about interpreting quantities like $P^*(\alpha^*|B_l(\alpha^*))$ and $Q_l(B_l(\alpha^*))$.

Sigma Algebra for Θ. N is always discrete. If T is discrete as well, then so are Θ and Ω, and there is no issue about the domain for measures on Ω. More generally, we will go from a σ-algebra on T to one on Θ, and finally to Ω. Along the way, we will need to define measurability of B_l and show that the resulting Ω is a measurable subset of Θ.

We start, then, with a given σ-algebra, σ_T, on T. σ_T extends to Θ in a natural way through a "skeleton" partitioning: Define $\tilde{\Theta}$ to be the set of finite trees with *unlabeled* terminal nodes, and nonterminal nodes labeled by elements of N. Define a skeleton function S on Θ which maps $\omega \in \Theta$ to the element $s \in \tilde{\Theta}$ that has the same topology and the same nonterminal labels. Let $\Theta_s = \{\omega \in \Theta : S(\omega) = s\}$ and note that the skeletons partition Θ, so that $\Theta = \bigcup_{s\in\tilde{\Theta}} \Theta_s$.

Let n_s be the number of leaves in $s \in \tilde{\Theta}$ (and, therefore, the number of terminals in each $\omega \in \Theta_s$). Let $\sigma_T^{n_s}$ be the product σ-algebra on T^{n_s} generated by σ_T. There is a natural one-to-one and onto mapping from T^{n_s} to Θ_s, that corresponds to the left-to-right sequence of terminals $t_1, t_2, \ldots t_{n_s}$ in a tree $\omega \in \Theta_s$. Call this mapping M_s. M_s induces a σ-algebra on Θ_s that we will denote σ_s:

$$\sigma_s = \{M_s(A) : A \in \sigma_T^{n_s}\}$$

Finally, we define a σ-algebra on Θ by

$$\mathcal{F} = \{\cup_{s\in\tilde{\Theta}} A_s : A_s \in \sigma_s,\ \forall s \in \tilde{\Theta}\}$$

(Note that $A \in \mathcal{F},\ A = \bigcup_{s\in\tilde{\Theta}} A_s \Rightarrow A^c = \bigcup_{s\in\tilde{\Theta}} A_s^c \in \mathcal{F}$.)

Measurability. We need to establish that Ω is a measurable subset of Θ. Since Ω is defined through the binding functions $B_l,\ l \in N$, the measurability of Ω is tied to the measurability of B_l.

Recall that the binding functions are defined on Θ^*, which is a union of product spaces, $\Theta^* = \bigcup_{n=1}^{\infty} \Theta^n$. Given the σ-algebra $\mathcal{F}$ on Θ, let $\mathcal{F}^n$ be the product σ-algebra on Θ^n and let $\mathcal{F}^*$ be the corresponding σ-algebra on Θ^*:

$$\mathcal{F}^* = \{\cup_{n=1}^{\infty} A_n : A_n \in \mathcal{F}^n,\ n = 1, 2, \ldots\}$$

We shall assume that $\mathcal{R}_l$ is equipped with a σ-algebra $\mathcal{B}_l$, that $\mathcal{S}_l \subseteq \mathcal{R}_l$ is $\mathcal{B}_l$-measurable, and that $B_l : \Theta^* \to \mathcal{R}_l$ is a measurable function relative to the corresponding σ-algebras: $B_l^{-1}(A) \in \mathcal{F}^*,\ \forall A \in \mathcal{B}_l$. Then Ω is measurable:

Proposition 4.1 $\Omega \in \mathcal{F}$

Proof. Since $\Omega = \bigcup_{s \in \tilde{\Theta}} (\Omega \cap \Theta_s)$, and since $\tilde{\Theta}$ is countable (recall that N is discrete), it is enough to show that $\Omega_s \triangleq \Omega \cap \Theta_s$ is measurable for each $s \in \tilde{\Theta}$.

Let $i = 1, 2, \ldots m_s$ index the nodes of the skeleton s, left-to-right, breadth first, beginning with the root node. Let $I_s \subseteq \{1, 2, \ldots m_s\}$ be the indices of the nonterminal nodes, and given $i \in I_s$, let $l_i \in N$ be the label of node i. For each $i \in I_s$, let $\{L_i, L_i + 1, \ldots R_i\}$ be the indices, left-to-right, of the daughter nodes of i (so that $L_i = R_i$ if there is only a single daughter node).

Introduce terminal variables, $t_1, t_2, \ldots t_{n_s} \in T$, and associate these, left-to-right, with the terminal nodes of s, where n_s is the number of terminal nodes in s. (Note that the index of the node associated with t_i will not be i.) Let $\vec{t_i} = (t_{b_i}, t_{b_i+1}, \ldots t_{e_i})$ be the terminal variables belonging to the subtree rooted at $i \in \{1, 2, \ldots m_s\}$ ($e_i = b_i$ whenever i is a terminal node). Finally, for each $i \in \{1, 2, \ldots m_s\}$, let M_s^i be the mapping from $T^{b_i - e_i + 1}$ into Θ that assigns to $(t_{b_i}, \ldots t_{e_i})$ the subtree rooted at i and having left-to-right terminals $t_{b_i}, \ldots t_{e_i}$.

Then

$$\Omega_s = M_s \left\{ \bigcap_{i \in I_s} \{(t_1, \ldots t_{n_s}) : B_{l_i}(M_s^{L_i}(\vec{t}_{L_i}), \ldots M_s^{R_i}(\vec{t}_{R_i})) \in \mathcal{S}_{l_i}\} \right\}$$

Hence, it is enough to show that

$$\{(t_1, \ldots t_{n_s}) : B_{l_i}(M_s^{L_i}(\vec{t}_{L_i}), \ldots M_s^{R_i}(\vec{t}_{R_i})) \in \mathcal{S}_{l_i}\} \in \sigma_T^{n_s}$$

for each $i \in I_s$. But this follows immediately from: the measurability of $\mathcal{S}_{l_i}$; the measurability of the function B_{l_i}; and the (*by definition*) measurability of the mappings M_s^i. $\square$

Compositional Measures. To put a measure on Ω, we start with the auxiliary (or "observable") measures Q and Q_l:

Q is a probability measure on $T \cup N$

Q_l is a probability measure on $\mathcal{R}_l$ concentrating on $\mathcal{S}_l$, for each $l \in N$.

Q determines the relative frequencies among object types, and Q_l determines the distribution on binding values, B_l, for each object type $l \in N$.

Given a probability measure P on Ω, define P^* on Ω^* by

$$P^*(A) = \sum_{n=1}^{\infty} P^n(A_n)$$

where $A = \bigcup_{n=1}^{\infty} A_n$, $A_n \in \Omega^n$, and P^n is the n-fold product measure $P \times \cdots \times P$. Then, define P_l^* to be the measure induced by P^* on $\mathcal{R}_l$ through B_l:

$$P_l^*(S) = P^*(\alpha^* : B_l(\alpha^*) \in S)$$

for measurable $S \subseteq \mathcal{R}_l$. If $Q_l \ll P_l^* \ \forall l \in N$, then

$$\frac{dQ_l}{dP_l^*}(B_l(\alpha^*))P^*(\alpha^*)$$

defines a measure (say μ^*) on Ω^*:

$$\mu^*(A) = \int_{\alpha^* \in A} \frac{dQ_l}{dP_l^*}(B_l(\alpha^*))dP^*(\alpha^*)$$

Hence, if $Q_l \ll P_l^* \ \forall l \in N$, then

$$Q(l)\frac{dQ_l}{dP_l^*}(B_l(\alpha^*))dP^*(\alpha^*)$$

defines a measure on $N \times \Omega^*$, for which we will use the more intuitive form[3]

$$Q(l)Q_l(B_l)P^*(\alpha^*|B_l) \tag{5}$$

[3]Alternatively, we could make sense of (5) more directly by first defining a conditional probability $P^*(\alpha^*|B_l)$, and arrive at the same construction. But this is perhaps more complicated since we need to choose a "version" of $P^*(\alpha^*|B_l)$ and, what's more, P^* is not a probability measure.

(Code the label, l; code the binding value, B_l; and then code the constituents, α^*, as though they were independent, but conditioned on the binding value.)

Note that (5) defines a measure on $\Omega \backslash T$ through the mapping $(l, \alpha^*) \leftrightarrow l(\alpha^*)$:

> **Definition.** A probability measure P on Ω is a *compositional measure* if
>
> **a.** P agrees with Q on T
>
> **b.** $Q_l \ll P_l^* \ \forall l \in N$
>
> **c.** P agrees with
> $$Q(l)Q_l(B_l)P^*(\alpha^*|B_l)$$
> on $\Omega \backslash T$

Remarks.

1. More succinctly, $P(\omega)$ is $Q(\omega)$ if $\omega \in T$ and $Q(l)Q_l(B_l)P^*(\alpha^*|B_l)$ if $\omega = l(\alpha^*)$, which is (4).

2. Neither the existence nor the uniqueness of a compositional measure is guaranteed. Chi ([6]) has an existence result that applies when Ω is finite, and applies as well to various extensions of finite systems. An example of nonuniqueness is constructed in Potter ([31]). The following section (§4.3) contains several examples of compositional measures, and some examples in which no compositional measure exists. One way to guarantee existence (and uniqueness) is to build the measure "bottom up:" Assume $T \bigcap \mathcal{S}_l = \emptyset \ \forall l \in N$, define a "type" function, τ_ω, by

$$\tau_\omega = \begin{cases} \omega & \omega \in T \\ B_l(\alpha^*) & \omega = l(\alpha^*) \in \Omega, \end{cases}$$

and assume that there exists $\emptyset = A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots$ such that

(a) $\bigcup_{i=1}^{\infty} A_i = \bigcup_{l \in N} \mathcal{S}_l$, and

(b) If $\tau_\omega \in A_{k+1}$ $(k \geq 0)$ for some $\omega = l(\alpha_1, \ldots \alpha_n) \in \Omega$, then $\tau_{\alpha_1}, \ldots \tau_{\alpha_n} \in T \cup A_k$

If $\omega \in T$ then $P(\omega) = Q(\omega)$. This defines P_l^* on $A_1 \cap \mathcal{S}_l$, $\forall l \in N$, which in turn defines P on $\{\omega : \tau_\omega \in A_1\}$. Now take the next step: use P on $\{\omega : \tau_\omega \in A_1\}$ to define P_l^* on $A_2 \cap \mathcal{S}_l$, $\forall l \in N$, and from this get P on $\{\omega : \tau_\omega \in A_2\}$. Continuing inductively we get P on $T \bigcup \{\bigcup_{i=1}^{\infty} \{\omega : \tau_\omega \in A_i\}\} = \Omega$.

Of course for the construction to make sense one needs to check that $Q_l \ll P_l^*$ $\forall l \in N$, but this is generally easy to arrange. On the other hand, this bottom-up solution is not entirely satisfactory, since it precludes any real recursion. In order to allow for a construction like *'line'* + *'line'* $\rightarrow$ *'line'*, for example, one would need either to assign separate labels to lines of different "depths," or (perhaps more attractive) to construct $B_{\text{'line'}}$ in such a way that it reflects, in its value, the depths of its arguments.

3. If P satisfies (4), then $P(\Omega) = 1$. To see this, note first that if $l(\alpha^*) \notin \Omega$ then $B_l(\alpha^*) \notin \mathcal{S}_l$, and therefore $B_l(\alpha^*) \notin \text{Support}(Q_l)$. Hence $Q_l(B_l)P^*(\alpha^*|B_l)$ concentrates on $\{\alpha^* : l(\alpha^*) \in \Omega\}$. Thus

$$P(\Omega) = Q(T) + \sum_{l \in N} Q(l) \int_{\{\alpha^* \in \Omega^* : l(\alpha^*) \in \Omega\}} \frac{dQ_l}{dP_l^*}(B_l(\alpha^*))dP^*(\alpha^*)$$

$$= Q(T) + \sum_{l \in N} Q(l) \int_{\alpha^* \in \Omega^*} \frac{dQ_l}{dP_l^*}(B_l(\alpha^*))dP^*(\alpha^*)$$

$$\text{(change variables: } \alpha^* \rightarrow B_l(\alpha^*),\ P^* \rightarrow P_l^*\text{)}$$

$$= Q(T) + \sum_{l \in N} Q(l) \int_{b \in \{B_l(\alpha^*) : \alpha^* \in \Omega^*\}} \frac{dQ_l}{dP_l^*}(b)dP_l^*(b)$$

$$= Q(T) + \sum_{l \in N} Q(l) \int_{b \in \{B_l(\alpha^*) : \alpha^* \in \Omega^*\}} dQ_l(b)$$

$$= Q(T) + \sum_{l \in N} Q(l) Q_l\{B_l(\alpha^*) : \alpha^* \in \Omega^*\}$$

$$= \text{(since } Q_l \ll P_l^*\text{)}\ Q(T) + \sum_{l \in N} Q(l) Q_l(\mathcal{R}_l) = Q(T) + Q(N) = 1$$

4. As we have already seen, if $|T| < \infty$ and $|N| < \infty$, and if $B_l(\alpha_1, \dots \alpha_n) = (L(\alpha_1), \dots L(\alpha_n))$, then the composition system is equivalent to a context-free grammar. The probabilities $Q_l(\cdot)$ then correspond to the production probabilities of a probabilistic context-free grammar (PCFG). But in a PCFG, there is a special "start" symbol $S \in N$, and all of the mass concentrates on the set $\omega \in \Omega \ \ni L(\omega) = S$. The PCFG comes out of a compositional measure by conditioning on $L(\omega) = S$: If Q is any probability measure with support $N \bigcup T$, and if P is the associated compositional measure, then $P(\omega | L(\omega) = S)$ coincides with the standard PCFG measure. The result, of course, is independent of Q, but we can not simply take $Q(S) = 1$ without violating $Q_l \ll P_l^*$.

5. Recall the earlier discussion about bits saved (§4.2.1): In a discrete world, the composition $\alpha + \beta \rightarrow \omega = l(\alpha, \beta)$ saves

$$\log_2 \frac{P(\omega)}{P(\alpha)P(\beta)} = \log_2 \frac{Q(l)Q_l(B_l)}{P \times P(B_l)}$$

bits. This can be viewed as the logarithm of a likelihood ratio of two measures on $\Omega \times \Omega$: A compositional measure $Q(l)Q_l(B_l)P \times P(\alpha, \beta | B_l)$, and a product measure $P \times P(\alpha, \beta)$. More generally (but still in the discrete case): $\alpha^* \rightarrow \omega = l(\alpha^*)$; the likelihood ratio is between $Q(l)Q_l(B_l)P \times P(\alpha^* | B_l)$ and P^*, on Ω^*; and there is a savings of

$$\log_2 \frac{Q(l)Q_l(B_l)}{P_l^*(B_l)}$$

bits.

Of course the constituents, coded under P, already themselves represent a savings over their respective (sub) constituents, and so on from the root to the leaves of the tree. To pursue this, fix $\omega \in \Omega$ and consider the skeleton $s = S(\omega)$ (as in §4.2.2). Following the proof of proposition 4.1, let $t_1, \dots t_{n_s}$ be the terminal variables (keeping in mind that these are not necessarily distinct), let I_s index the nonterminal nodes, let $l_i \in N$ be the label of node i, and, finally, let b_i be the value taken by the binding function (B_{l_i}) at node i. (So, for example, if $\alpha_1, \dots \alpha_n \in \Omega$ are the trees rooted at the daughter nodes of i, then

$b_i = B_{l_i}(\alpha_1, \dots \alpha_n)$.) Then, through s, P induces a measure on T^{n_s}:

$$(\prod_{i \in I_s} R_i)(\prod_{i=1}^{n_s} Q(t_i))$$

where $R_i = Q(l_i)Q_{l_i}(b_i)/P^*_{l_i}(b_i)$ is the likelihood ratio associated with the i'th node, $i \in I_s$. Of course this is just a rewriting of $P(\omega)$ by successively expanding the constituent probabilities, but it points out the overall win achieved by encoding $t_1, \dots t_{n_s}$ as a composition, ω, over a coding as independent entities under Q: the likelihood ratio is $\prod_{i \in I_s} R_i$, which corresponds to saving $\sum_{i \in I_s} \log_2 R_i$ bits.

More or less the same analysis goes through in the general case: By fixing ω and the corresponding skeleton s, we observe that P defines a measure on the product space T^{n_s} which is absolutely continuous with respect to the product measure Q^{n_s}. The derivative is $\prod_{i \in I_s} R_i$ where, in general,

$$R_i = Q(l_i)\frac{dQ_{l_i}(b_i)}{dP^*_{l_i}(b_i)}$$

($\leftrightarrow Q(l_i)Q_{l_i}(b_i)/P^*_{l_i}(b_i)$ in the discrete case).

Actually, there is a whole hierarchy of measures on T^{n_s}, starting with $\prod_{i=1}^{n_s} Q(t_i)$, and working up to $P(\omega)$. Along the way are product measures with more than one but fewer than n_s terms. From this viewpoint, $\prod_{i \in I_s} R_i$ is just the result of applying the chain rule, starting at the top with $dP(\omega)$, going through $dP(\alpha_1) \cdots dP(\alpha_n)$ (if $\omega = l(\alpha_1, \dots \alpha_n)$), and working down to $dQ(t_1) \cdots dQ(t_{n_s})$.

There is also an interpretation in terms of bits. If there is enough structure to permit a discrete $\rightarrow$ continuous limit, then coding $t_1 \in A^\epsilon_1, t_2 \in A^\epsilon_2, \dots t_{n_s} \in A^\epsilon_{n_s}$, under the product measure Q^n, costs

$$\sum_{i=1}^{n_s} - \log_2 Q(A^\epsilon_i)$$

bits. With the skeleton, s, fixed, there is a corresponding set $A^\epsilon = A^\epsilon(A^\epsilon_1, \dots A^\epsilon_{n_s}) \subseteq \Omega$, and the corresponding event $\omega \in A^\epsilon$ can be coded with $-\log_2 P(A^\epsilon)$ bits. If, for each i, $Q(A^\epsilon_i) \downarrow 0$ as $\epsilon \downarrow 0$, then of course $\sum_{i=1}^{n_s} - \log_2 Q(A^\epsilon_i)$ and $-\log_2 P(A^\epsilon)$ will, in general, both become infinite. Nevertheless, since $P \ll Q^{n_s}$ (P viewed as a measure on

T^{n_s}), if the continuum limit exists then $P(A^\epsilon)/\prod_{i=1}^{n_s} Q(A_i^\epsilon)$ converges to a derivative, and the *gain*, $\log_2 P(A^\epsilon) - \sum_{i=1}^{n_s} \log_2 Q(A_i^\epsilon)$, has limit $\sum_{i=1}^{n_s} R_i$.

6. The model suggests an approach to *recognition*: use P to build a measure on *image interpretations* (see §5) and then choose the best interpretation among competing candidates. This is essentially what's behind the experiments discussed in §2, and, evidently, it requires an explicit form for the "win" when interpreting constituents as parts in a composition, over their interpretation as independent components. Unfortunately, this calculation appears to be very difficult in general and calls for some sort of approximation in actual implementations. An example is worked through in Huang ([23]). See also Potter ([31]) for a set up in which exact calculations are possible.

4.3 Examples

1. A Context-Free System. Probabilistic context-free grammars are special cases of compositional measures. Probabilities on context-free grammars demonstrate a kind of criticality (cf. [3], [40]): depending upon the production probabilities, there may be a nonzero probability of producing trees of *infinite* depth. This "lack of tightness" (better known as "inconsistency" in the computational linguistics literature) leaves the total mass on Ω smaller than one. Since compositional measures always have mass one (as demonstrated earlier), the corresponding system (4) must have no solution.

For a specific example, take $T = \{t\}$, $N = \{S\}$, and

$$B_S(\alpha^*) = \begin{cases} 1 & \text{if } \alpha^* = t \\ 2 & \text{if } \alpha^* = (S,S) \\ 0 & \text{otherwise} \end{cases}$$

Then $\mathcal{S}_S = \{1,2\}$ corresponds to the context-free system

$$\begin{array}{lcl} S & \rightarrow & SS \\ S & \rightarrow & t \end{array}$$

Q is basically irrelevant, but to be concrete take $Q(t) = Q(S) = 1/2$. The context-free system is critical at $Prob(S \rightarrow SS) = Prob(S \rightarrow t) = 1/2$:

$Prob(S \to SS) > 1/2 \Rightarrow Prob(\Omega) < 1$ and $Prob(S \to SS) \leq 1/2 \Rightarrow Prob(\Omega) = 1$. These two cases correspond to

$$Q_S(b) = \begin{cases} p & \text{if } b = 2 \\ 1-p & \text{if } b = 1 \end{cases}$$

with $p > 1/2$ or $p \leq 1/2$, respectively. When $p \leq 1/2$, a unique compositional measure P exists, and coincides with the corresponding probability for the context-free system. On the other hand, if $p > 1/2$, then (4) has no solution.

2. Another Kind of Nonexistence. Existence can fail in a different way—by including binding rules that are not sufficiently restrictive. Consider the simple system $T = \{t\}$, $N = \{S\}$, and $B_S(\alpha^*) = 1$, $\forall \alpha^*$, with $\mathcal{S}_S = \{1\}$. Anything goes: $\Omega = \Theta$. Since Q_S concentrates on $\mathcal{S}_S$, $Q_S(\{1\}) = 1$. Regardless of Q, if P exists then

$$P_S^*(\{1\}) = P^*(\alpha^* : B_S(\alpha^*) = 1) = \sum_{n=1}^{\infty} P^n((\alpha_1, \dots \alpha_n) : B_S(\alpha_1, \dots \alpha_n) = 1)$$

$$= \sum_{n=1}^{\infty} P^n(\Omega^n) = \sum_{n=1}^{\infty} 1 = \infty,$$

which makes $P(\omega) = 0$ for any $\omega \neq t$. So P exists only when $Q(t) = 1$, in which case $P(t) = 1$ as well.

The example is extreme, since there is no restriction on the number of constituents. Still, things can be made to work by coding the number of constituents into the binding-function value, e.g. with $B_S(\alpha_1, \dots \alpha_n) = n$ and $\mathcal{S}_S = \{1, 2, \dots\}$, in which case we have a simple branching process. Then, for suitable (i.e. subcritical—see [20]) Q_l, we get a probability measure P on Ω.

3. A Context-Sensitive System. Take, again, $T = \{t\}$, and $N = \{S\}$. Given $\alpha \in \Theta$, define $|\alpha|$ to be the number of terminals in α. If

$$B_S(\alpha^*) = \begin{cases} 1 & \text{when } \alpha^* = (\alpha, \beta),\ |\alpha| = |\beta| \\ 0 & \text{otherwise} \end{cases}$$

and $\mathcal{S}_S = \{1\}$, then Ω is the set of balanced binary trees. The associated *language* (i.e. the set of left-to-right sequences of terminal) is the set of strings of t of length 2^n, $n \geq 0$. This language is not context free, as is established by an elementary application of the Pumping Lemma (cf. Hopcroft and Ullman [22]).

Let $Q(S) = p$ and $Q(t) = q \overset{\triangle}{=} 1 - p$, with $p \in (0, 1)$. If there is a compositional measure, and if P_n is the probability of the (only) $\omega \in \Omega$ with $|\omega| = 2^n$, then

$$P_0 = q,\ P_1 = cpq^2,\ P_2 = c^3p^3q^4, \ldots P_n = (cpq)^{2^n}/cp$$

where

$$\frac{1}{c} = P \times P\{(\alpha, \beta) : |\alpha| = |\beta|\} = \sum_{n=0}^{\infty} P_n^2 = (\frac{1}{cp})^2 \sum_{n=0}^{\infty} (cpq)^{2^{n+1}} \tag{6}$$

Therefore, if there is a compositional measure, then there exists $c > 1$ (since $c = 1/P \times P\{(\alpha, \beta) : |\alpha| = |\beta|\}$) satisfying (6). On the other hand, if $c > 1$ satisfies (6), then $P_0, P_1, \ldots$ define a compositional measure. Does (6) have such a solution?

$$\frac{1}{c} = (\frac{1}{cp})^2 \sum_{n=0}^{\infty} (cpq)^{2^{n+1}} = \frac{1}{cp}\{\frac{1}{cp} \sum_{n=1}^{\infty} (cpq)^{2^n}\} = \frac{1}{cp}\{\frac{1}{cp} \sum_{n=0}^{\infty} (cpq)^{2^n} - q\}$$

$$\Longleftrightarrow$$

$$cp = \sum_{n=0}^{\infty} (cpq)^{2^n}$$

The right hand side is strictly convex in c. Draw the respective graphs of the right and left hand sides: a solution with $c > 1$ exits if and only if

$$p > \sum_{n=0}^{\infty} (pq)^{2^n},$$

and if a solution does exit then it is unique. But

$$\sum_{n=0}^{\infty} (pq)^{2^n} < \sum_{n=1}^{\infty} (pq)^n = \frac{pq}{1 - pq}$$

$$= p(\frac{1 - p}{1 - p + p^2}) = p(\frac{1}{1 + (\frac{p^2}{1-p})}) < p$$

So a unique compositional measure always exists.

4. Points, Linelets, and Lines.

4. Points, Linelets, and Lines. Return again to the example discussed in §2, involving "on-line" characters. Here T is the $M \times M$ grid of sampling locations, $T = \{1, 2, \ldots M\} \times \{1, 2, \ldots M\}$. Linelets are just pairs of points sufficiently close together:

$$\begin{aligned} 1 \in N \quad &\leftrightarrow \quad \text{'linelet'} \\ B_1(\alpha^*) \quad &= \quad \begin{cases} \|\alpha - \beta\| & \text{if } \alpha^* = (\alpha, \beta),\ \alpha, \beta \in T \\ -1 & \text{otherwise} \end{cases} \\ \mathcal{S}_1 \quad &= \quad (0, r] \end{aligned}$$

where $\|\cdot\|$ is Euclidean distance and r is a distance threshold, defining "sufficiently close."

(We proceed, here and later, as though we were in the continuum: $\mathcal{S}_1$ is continuous and we will choose Q_1 with support equal to $\mathcal{S}_1$. Since, in fact, the range of B_1 is discrete, it is clear that the condition $Q_1 \ll P_1^*$ will be violated. But this is really just an expedient way of constructing a composition—it is easier to work in the continuum. For a completely proper construction, partition $\mathcal{S}_1$ into *attainable* intervals, to which Q_1 assigns probabilities. B_1 can then be thought of as interval-valued, in which case $Q_1 \ll P_1^*$.)

Lines come from lines or linelets by adding single points, or from two suitably-situated colinear lines. We construct B_2 ($2 \in N \leftrightarrow$ "line") by treating these cases separately. First, introduce an "End Point" function $\mathcal{EP}(\omega)$, $\omega \in \Omega$, whose value is the endpoints of ω when ω is a line segment. Specifically, $\mathcal{EP}(\omega)$ is the set of two terminals in ω that achieves the greatest distance among all pairs of terminals in ω (with some convention for systematically breaking ties).

Given the endpoints $e_1,\ e_2$, let $RT(e_1, e_2)$ be the rectangle depicted in Figure 2, where w and l are additional "threshold" parameters. When $L(\alpha) \in \{1, 2\}$ (line or linelet) and $\beta \in T$, define

$$B_2(\alpha, \beta) = \begin{cases} 1 & \text{if } \beta \in RT(\mathcal{EP}(\alpha)) \\ 0 & \text{otherwise} \end{cases}$$

which covers growth by single points.

The composition *'line' + 'line' → 'line'* requires, first, that an endpoint of one line be close to an endpoint of the other. With this constraint in mind, let $\|\alpha - \beta\|_2$ be the minimum distance between the points in $\mathcal{EP}(\alpha)$ and the points in $\mathcal{EP}(\beta)$. As for the colinearity condition, this will be enforced in terms of the angle between the constituent lines: Let $\angle(\alpha, \beta)$ be the angle (between 0 and 2π) from α to β through the pair of endpoints achieving the minimum $\|\alpha - \beta\|_2$ (again, with a systematic mechanism for resolving ties). When $L(\alpha) = L(\beta) = 2$, define $B_2(\alpha, \beta) = (\|\alpha - \beta\|_2, \angle(\alpha, \beta))$. Putting together the pieces:

$$B_2(\alpha, \beta) = \begin{cases} 1_{RT(\mathcal{EP}(\alpha))}(\beta) & \text{if } L(\alpha) \in \{1, 2\},\ \beta \in T \\ \\ (\|\alpha - \beta\|_2, \angle(\alpha, \beta)) & \text{if } L(\alpha) = L(\beta) = 2 \\ \\ -1 & \text{otherwise} \end{cases}$$

and $\mathcal{S}_2 = ((0, c] \times [\pi - \triangle, \pi + \triangle]) \bigcup \{1\}$. The parameters c and $\triangle$ determine when two lines are "close" and "colinear," respectively.

It is more-or-less straightforward to design further compositions, for 'L-junctions' and 'T-junctions', letters, strings, words, and so-on. In general, it is a good idea to build some scale invariance into the rules, and for this it is decidedly easier to work in the continuum—see [31] and [23]. But the system here is simple and illustrative.

A compositional probability is generated by specifying the "Q measures"—Q and Q_l, $l \in N$. The label probabilities, $Q(t)$, $t \in T$, $Q(1)$, and $Q(2)$ are pretty much arbitrary, but would presumably reflect some knowledge or expectation about the relative likelihoods of observing points, linelets, and lines. (See §5, where we develop the associated distribution on scenes of objects). For illustration, we can take, simply, $Q(1) = Q(2) = 1/3$, and $Q(t) = 1/(3M^2)$, making points, linelets, and lines equally likely and adopting a uniform distribution on the set of points. The analogs of production probabilities are the distributions Q_1 and Q_2 on $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. These will govern the relative likelihoods of the various linelet and line configurations. It would be natural to adopt, for example, a parametric class for Q_1 that favors small values in $(0, r]$, perhaps just a triangular density $q_1(x) = a - bx$, $b > 0$. There is one free parameter (the slope, b, say) which could in principle be estimated from labeled images. The distribution on $\mathcal{S}_2$ is mixed, with an atom on $\{1\}$, say 1/2, that fixes the relative proportion of the *'line' + 'point'* $\rightarrow$ *'line'* composition, and a distribution on $(0, c] \times [\pi - \triangle, \pi + \triangle]$ that fixes the relative proportions of the various configurations of two line segments joined to make a longer line segment.

What about the existence of a probability P satisfying (4)? Given (discrete versions of—see earlier remark) Q, Q_1, and Q_2, this is guaranteed by a result by Chi ([6]), which more generally guarantees existence anytime $|\Omega_{l,b}| < \infty \ \forall l, b$, where

$$\Omega_{l,b} = \{l(\alpha^*) \in \Omega : B_l(\alpha^*) = b\}$$

What is the win when two terminals, α and β, with $\|\alpha - \beta\| \leq r$, are joined to make a linelet ($\alpha + \beta \to l(\alpha, \beta)$, $l = 1 \leftrightarrow$ 'linelet')? Consider an image that includes, possibly among other objects, these two terminals. Consider this image as interpreted with the two terminals as isolated points, versus this same image with this same interpretation, excepting that the two points are viewed as composing a linelet. As we shall see shortly (§5), the ratio of the probabilities of these two image interpretations is

$$\frac{P(\alpha)P(\beta)}{P(l(\alpha,\beta))} \tag{7}$$

This ratio is independent of the other structures in the scene and independent of the interpretations of these other structures. Since $P(l(\alpha,\beta)) = Q(1)Q_1(\|\alpha-\beta\|)P \times P(\alpha,\beta|B_1 = \|\alpha-\beta\|)$ (everything is discrete, so $Q_1(\|\alpha-\beta\|)$ is an actual probability, and not a density evaluated at $\|\alpha - \beta\|$), the ratio (7) is just

$$\begin{aligned}
&\frac{P \times P(\alpha', \beta' : \alpha', \beta' \in T, \|\alpha' - \beta'\| = \|\alpha - \beta\|)}{Q(1)Q_1(\|\alpha - \beta\|)} \\
&= \frac{P(T)^2 P \times P(\|\alpha' - \beta'\| = \|\alpha - \beta\| | \alpha', \beta' \in T)}{Q(1)Q_1(\|\alpha - \beta\|)} \\
&= \frac{Q(T)^2 P \times P(\|\alpha' - \beta'\| = \|\alpha - \beta\| | \alpha', \beta' \in T)}{Q(1)Q_1(\|\alpha - \beta\|)} \\
&= \frac{1}{3}\frac{P \times P(\|\alpha' - \beta'\| = \|\alpha - \beta\| | \alpha', \beta' \in T)}{Q_1(\|\alpha - \beta\|)}
\end{aligned}$$

If, for example, $\|\alpha - \beta\| = 1$, then there are four locations for β' for any (interior) α' such that $\|\alpha' - \beta'\| = \|\alpha - \beta\|$. Up to edge effects, then,

$$P \times P(\|\alpha' - \beta'\| = \|\alpha - \beta\| | \alpha', \beta' \in T) \approx \frac{4}{M^2}$$

and the ratio

$$\frac{P(\alpha)P(\beta)}{P(l(\alpha,\beta))} \approx \frac{4}{3M^2}\frac{1}{Q_1(\|\alpha-\beta\|)}$$

On the other hand, $Q_1(1)$ will be much larger than $1/M^2$, so that

$$\frac{P(\alpha)P(\beta)}{P(l(\alpha,\beta))} \ll 1$$

The interpretation "linelet" is substantially better than the alternative "two independent points."

What happens if, say, two line segments (α and β) are joined to form a longer line segment ($l(\alpha,\beta),\ l=2$)? The ratio $P(\alpha)P(\beta)/P(l(\alpha,\beta))$ would be

$$\frac{P(L(\alpha')=2)^2}{Q(2)Q_2((\|\alpha-\beta\|_2,\angle(\alpha,\beta))}\times$$

$$P\times P((\|\alpha'-\beta'\|_2,\angle(\alpha',\beta'))=(\|\alpha-\beta\|_2,\angle(\alpha,\beta))|L(\alpha')=L(\beta')=2)$$

$$=\frac{1}{3Q_2((\|\alpha-\beta\|_2,\angle(\alpha,\beta))}\times$$

$$P\times P((\|\alpha'-\beta'\|_2,\angle(\alpha',\beta'))=(\|\alpha-\beta\|_2,\angle(\alpha,\beta))|L(\alpha')=L(\beta')=2)$$

If, for example, $\|\alpha-\beta\|_2=1$, then under $P\times P$, the event $\|\alpha'-\beta'\|_2=1$ has, approximately, probability $16/M^2$ and $\angle(\alpha,\beta)$ is essentially uniform (again, up to edge effects). On the other hand, Q_2 severely restricts $\|\alpha-\beta\|_2$ as well as the angle $\angle(\alpha,\beta)$, and therefore if $B_2\in\mathcal{S}_2$ then $Q_2((\|\alpha-\beta\|_2,\angle(\alpha,\beta))\gg P\times P((\|\alpha'-\beta'\|_2,\angle(\alpha',\beta'))=(\|\alpha-\beta\|_2,\angle(\alpha,\beta))|L(\alpha')=L(\beta')=2)$. Again, composition is strongly favored.

By the same kind of reasoning, for the case *'line'* + *'point'* $\rightarrow$ *'line'* ($\alpha+\beta\rightarrow l(\alpha,\beta),\ l=2$),

$$\frac{P(\alpha)P(\beta)}{P(l(\alpha,\beta))} \approx \frac{1}{3}\frac{A(\alpha)/M^2}{Q_2(\{1\})}$$

where $A(\alpha)$ is the area of the rectangle in Figure 2. If for instance we assign half of the "line mass" to *'line'* + *'point'* (and the other half to *'line'* + *'line'*) then $Q_2(\{1\})=1/2$ and

$$\frac{P(\alpha)P(\beta)}{P(l(\alpha,\beta))} \approx \frac{2A(\alpha)}{3M^2}$$

which will, again, be much smaller than one.

Xiaohua Xing developed a hierarchy of composition rules that included, at "the top," the twenty-six upper case characters. Each rule was appended with a formula for computing the gain enjoyed by encoding a composition instead of encoding separately the constituents. In the examples above, this would roughly correspond to the negative logarithm of the probability ratios. (Although, the actual gains used were more or less ad hoc—we had not yet developed a probabilistic framework.) A simple brute force search algorithm (described briefly in §2) was used to find a best labeling of the entire image. Figures 1 and 3, taken from the experiments by Xing, show examples of images with handwritten characters correctly identified.

Potter's experiments (see [31]) go further, including compositions for strings of letters and entire words, and using gains computed, as in the example here, directly from a proper probability measure on Ω.

5 Scenes and Images

We will use the term "scene" in a formal sense to mean a finite collection of objects, i.e. a finite subset of Ω. All of the details about subcomponents, their relationships, and their placements are coded in $\omega \in \Omega$, so that a "scene" amounts to a very specific description. The term "image" will refer to what is actually observed, which may be, for example, just the terminal nodes of the objects in a scene, or, more generally, a "corrupted" or "noisy" version of the objects in a scene. The recognition problem is to find the scene given the image.

We propose to formulate this as an inference problem by using a compositional measure to construct a ("prior") probability distribution on scenes. Recognition becomes a problem in Bayesian inference: Use the *posterior* distribution, which is the conditional distribution on scenes given an image, to choose a good interpretation. Of course, the "best" interpretation (say, the maximum *a posteriori* scene) would be desirable, but is usually intractable. Instead, we foresee using the posterior distribution as a guide in identifying and choosing among sensible interpretations, as was the case in our experiments with handwritten character recognition (§2).

The goal, then, is to extend a compositional measure, P on Ω, to a distribution on the set of *scenes*, Ψ:

$$\Psi = \bigcup_{k=0}^{\infty} \Omega_k$$

where Ω_0 is the empty scene (no objects—call it $\{\epsilon\}$), and Ω_k is the collection of subsets of Ω of size k, $\Omega_k = \{\{\omega_1, \ldots \omega_k\} : \omega_i \in \Omega, i = 1, \ldots k\}$. It is perhaps best to follow our development of measures on Ω, by starting with the discrete case (Ω countable), where we can rely on the MDL point of view for motivation and intuition, and then extending to a more general framework.

5.1 Discrete Objects

Let us assume, for the time being, that Ω is countable, so that a compositional measure P is just an assignment of probabilities to the individual objects $\omega \in \Omega$. We will work from a prefix code c, which assigns a bit string to each $\omega \in \Omega$ and corresponds to the measure P—i.e. c is a Shannon code for P.

A scene is a finite collection of objects. One natural way to a code scene is to code the objects in the scene, one at a time. Order doesn't matter, so let us say, arbitrarily, that given $\sigma = \{\omega_1, \ldots \omega_k\} \in \Psi$, we use the code $c(\omega_{i_1})c(\omega_{i_2}) \ldots c(\omega_{i_k})$, where $c(\alpha)c(\beta)$ is the concatenation of $c(\alpha)$ and $c(\beta)$, and where $(i_1, i_2, \ldots i_k)$ is the permutation of $(1, 2, \ldots k)$ that yields $c(\omega_{i_1}) <_{\mathcal{L}} c(\omega_{i_2}) <_{\mathcal{L}} \ldots <_{\mathcal{L}} c(\omega_{i_k})$, "$<_{\mathcal{L}}$" being "lexicographic" ordering.[4]

But this is not yet a prefix code for Ψ—how would a receiver know when all of σ had been transmitted? In general, the code will be too short to correspond to a probability on Ψ. This is easily remedied: Take any two (preferably *short*) code words $c(\omega_1)$ and $c(\omega_2)$ with $c(\omega_1) <_{\mathcal{L}} c(\omega_2)$, and apply the suffix $c(\omega_2)c(\omega_1)$ to each of the scene codes just developed. Now we have a prefix code, and the corresponding probability distribution, D, on Ψ is simply

$$D(\sigma) = \frac{1}{z} \prod_{\omega \in \sigma} P(\omega) \tag{8}$$

The normalization $1/z$ corresponds, roughly, to the cost of the suffix, and is also the probability assigned to the empty scene, $\sigma = \{\epsilon\}$. Notice that

[4]Shorter words come first; words of the same length are ordered according to their interpretations as integers, represented in binary.

if we were to compare two scenes that differ by only one composition, such as $\sigma = \{\omega_1, \omega_2, \omega_3, \dots \omega_k\}$ and $\sigma' = \{\omega_1, \alpha, \beta, \omega_3, \dots \omega_k\}$ where $\omega_2 = l(\alpha, \beta)$, then the likelihood ratio $D(\sigma)/D(\sigma')$ is again just

$$\frac{Q(l)Q_l(B_l)}{P \times P(B_l)}$$

independent of the other elements, $\omega_1, \omega_3, \omega_4, \omega_5, \dots \omega_k$, of the scene. In fact, were we to *start* with these ratios, or "wins," then we would arrive at the same distribution, (8), over scenes, Ψ.

If a pair of object types, like tables and chairs, are systematically related in typical images, then the appropriateness of a product form (8) is called into question. But in a way, this is the whole point: such systematic relations *define* compositions. There is a binding function B_l and an empirical distribution Q_l under which $Q_l(B_l)$ is often much larger than $P_l^*(B_l)$.

Of course we need to make sure that (8) can actually be normalized, which is to say that $z < \infty$. In fact this follows from the coding argument, but more explicitly

$$z = \sum_{\sigma \in \Psi} \prod_{\omega \in \sigma} P(\omega) = \prod_{\omega \in \Omega} (1 + P(\omega)) < \infty$$

If, instead, we were to view scenes as *ordered* finite collections of objects, $\Psi = \{(\omega_1, \dots \omega_n) : 0 \leq n < \infty, \omega_k \in \Omega\}$, then

$$z = 1 + \sum_{n=1}^{\infty} P^n(\Omega^n) = \infty$$

No such measure exists. Since compositions respect order, viewing scenes as *ordered* collections of objects is like introducing a "start" or "scene" symbol $s \in N$, as in a context-free grammar, with the composition $\alpha^* \to s(\alpha^*)$ for any $\alpha^* \in \Omega^*$, which brings us back to the failure encountered in example 2 (§4.3). One way to unify the treatment of scenes and objects would be to treat trees with the same daughter structure as identical—i.e. ignore the order of branches. This works mathematically, but it is more awkward when it comes to constructing probabilities and calculating wins in specific examples.

The product form of (8) suggests a maximum entropy interpretation. Notice that for any $\omega \in \Omega$

$$D(\omega \in \sigma \,||\sigma| = 1) = \frac{\frac{1}{z}P(\omega)}{\sum_{\omega' \in \Omega} \frac{1}{z}P(\omega')} = P(\omega) \qquad \text{(a)}$$

and

$$D(\omega \in \sigma) = \frac{1}{z} P(\omega) \sum_{\substack{\sigma \in \Psi \\ \omega \notin \sigma}} \prod_{\omega' \in \sigma} P(\omega')$$

$$= \frac{1}{z} P(\omega) \prod_{\omega' \in \Omega/\omega} (1 + P(\omega')) = \frac{P(\omega)}{1 + P(\omega)} \qquad \text{(b)}$$

As it turns out, the maximum entropy distribution on Ψ, given (b), is in fact D: To see this, introduce Lagrange multipliers λ, $\{\lambda_\omega\}_{\omega \in \Omega}$, and maximize

$$- \sum_{\sigma \in \Psi} D(\sigma) \log D(\sigma) + \lambda \sum_{\sigma \in \Psi} D(\sigma) + \sum_{\omega \in \Omega} \lambda_\omega \Big(\sum_{\substack{\sigma \in \Psi \\ \omega \in \sigma}} D(\sigma) - \frac{P(\omega)}{1 + P(\omega)} \Big)$$

Differentiating with respect to $D(\sigma')$, and setting the derivative to zero, gives

$$- \log D(\sigma') - 1 + \lambda + \sum_{\omega \in \sigma'} \lambda_\omega = 0$$

$$\Longrightarrow D(\sigma') = k \prod_{\omega \in \sigma'} e^{\lambda_\omega} \Longrightarrow D(\omega \in \sigma') = \frac{e^{\lambda_\omega}}{1 + e^{\lambda_\omega}}$$

which, because of (b), must also equal $P(\omega)/(1 + P(\omega))$. Hence $D(\sigma') = \frac{1}{z} \prod_{\omega \in \sigma'} P(\omega)$.

5.2 Generalization

Generalization to the continuum case is more or less straightforward. The goal is again to preserve the "wins" already built into the compositional measure P. Of course in the continuum case the coding interpretation is lost (except as "bits saved" in a limit from discrete models—see §4.2.2), but the idea of using a product-type measure for scenes still makes sense. Formally, we want to define D on Ψ by (8), in which case we will get the "right" ratios when comparing scenes that differ only in that in one case a set of constituents is composed, and in the other it is not.

We will need to introduce a σ-algebra $\mathcal{F}_\Psi$ on Ψ, to serve as domain for D. Recall that $\Omega_k = \{\{\omega_1, \dots \omega_k\} : \omega_i \in \Omega, i = 1, \dots k\}$ is the set of subsets of Ω of size k. The σ-algebra $\mathcal{F}_\Psi$ will be defined through a collection of σ-algebras $\mathcal{F}_k$, one for each Ω_k. Start with $\mathcal{F}$, the σ-algebra developed for Ω (see §4.2.2), and let $\mathcal{F}^k$ be the corresponding product σ-algebra on Ω^k. Define Φ to be the mapping that carries each element of Ω_k into its associated symmetric set in Ω^k:

$$\Phi(\{\omega_1, \ldots \omega_k\}) = \{(\omega_{\sigma_1}, \ldots \omega_{\sigma_k}) : \sigma \text{ a permutation on } \{1, 2, \ldots k\}\},$$

and then define $\mathcal{F}_k$ by $A \in \mathcal{F}_k$ if $\Phi(A) \in \mathcal{F}^k$. Let $\mathcal{F}_0 = \{\{\epsilon\}, \emptyset\}$ (the σ-algebra for Ω_0, where $\emptyset$ is the empty set—*no scene*—which is different from $\{\epsilon\}$—the scene with *no objects*). And then, finally, define

$$\mathcal{F}_\Psi = \{\bigcup_{k=0}^{\infty} A_k : A_k \in \mathcal{F}_k\}$$

The formal expression (8), which is fine in the discrete case, doesn't really make sense in the general case. We can use, instead,

$$D(A) = \frac{1}{z} \frac{P^k(\Phi(A))}{k!}$$

for any $A \in \mathcal{F}_k$, since this extends immediately to $\mathcal{F}_\Psi$, and it reduces to (8) when Ω is discrete.

For the empty scene, $D(\{\epsilon\}) = 1/z$. The normalization is again finite, since

$$z = 1 + \sum_{k=1}^{\infty} \frac{P^k(\Phi(\Omega_k))}{k!} \leq 1 + \sum_{k=1}^{\infty} \frac{1}{k!} = e,$$

meaning that there exists a probability measure on scenes (formally, 8) which preserves, exactly, the likelihood ratios designed into P.

Acknowledgements. With remarkable efficiency, Xiaohua Xing developed an on-line upper-case character recognition system based upon a compositional representation of the twenty-six characters. The good performance of that system has done much to motivate our taking a deeper look at the principle of compositionality.

Elie Bienenstock has long argued for the importance of compositionality, not only as a basic principle of cognition, but also, even more fundamentally, as a basic principle of neural representation. Our work here draws from numerous and ongoing discussions with Bienenstock.

Finally, we have been heavily influenced by Ulf Grenander's way of thinking in the development of his General Pattern Theory. We are fortunate to have worked so closely with him.

References

[1] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[2] E. Bienenstock. Notes on the growth of a composition machine. In D. Andler, E. Bienenstock, and B. Laks, editors, *Proceedings of the Royaumont Interdisciplinary Workshop on Compositionality in Cognition and Neural Networks*, 1991.

[3] T.L. Booth and R.A. Thompson. Applying probability measures to abstract languages. *IEEE Trans. on Computers*, C-22:442–450, 1973.

[4] J. Canning. A minimum description length model for recognizing objects with variable appearances (the VAPOR model). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1032–1036, 1994.

[5] S. Casadei and S.K. Mitter. A hierarchical approach to high resolution edge contour reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996.

[6] Z. Chi. *Probability Models for Complex Systems.* PhD thesis, Division of Applied Mathematics, Brown University, 1998.

[7] N. Chomsky. *Syntactic Structures.* Mouton, 1976.

[8] N. Chomsky. *Knowledge of Language: Its Nature, Origin, and Use.* Praeger, 1986.

[9] D. B. Cooper. Feature selection and super data compression for pictures in remote conference and classroom communications. In *Proceedings of the Second International Joint Conference on Pattern Recognition*, pages 111–115, 1974.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, 1991.

[11] W. Ellis, editor. *A Source Book of Gestalt Psychology.* Humanities Press, 1938.

[12] J. Feldman. Formal constraints on cognitive interpretations of causal structure. In *Proceedings of the IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis*, 1995.

[13] J. Feldman. Perceptual models of small dot clusters. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 19:331–357, 1995.

[14] J. Feldman. Regularity-based perceptual grouping. *Computational Intelligence*, 13:582–621, 1997.

[15] J. Fodor and Z. Pylyshyn. Connectionism and cognitive architecture: a critical analysis. *Cognition*, 28:3–71, 1988.

[16] K. S. Fu. *Syntactic Methods in Pattern Recognition.* Academic Press, 1974.

[17] K. S. Fu. *Syntactic Pattern Recognition and Applications.* Prentice-Hall, 1982.

[18] U. Grenander. *General Pattern Theory: A Study of Regular Structures.* Oxford University Press, 1993.

[19] H. Gu, Y. Shirai, and M. Asada. MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:58–64, 1996.

[20] T.E. Harris. *The Theory of Branching Processes.* Springer-Verlag, Berlin, 1963.

[21] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

[22] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 1979.

[23] S.-H. Huang. *Compositional Approach to Recognition Using Multi-Scale Computations.* PhD thesis, Division of Applied Mathematics, Brown University, 2001.

[24] J. E. Hummel and I. Biederman. Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99:480–517, 1992.

[25] K. Knight. Unification: a multidisciplinary survey. *ACM Computing Surveys*, 21:93–124, 1989.

[26] P. S. Laplace. *Esssai philosophique sur les probabilités.* 1812. Translation of Truscott and Emory, New York, 1902.

[27] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, 1989.

[28] E. Mjolsness. Connectionist grammars for high-level vision. In V. Honavar and L. Uhr, editors, *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration.* Academic Press, 1994.

[29] R. Narasimhan. Labeling schemata and syntactic description of pictures. *Information and Control*, 7:151–179, 1964.

[30] T. Pavlidis. *Structural Pattern Recognition.* Springer-Verlag, 1977.

[31] D. F. Potter. *Compositional Pattern Recognition.* PhD thesis, Division of Applied Mathematics, Brown University, 1998.

[32] A. Prince and P. Smolensky. Optimality: from neural networks to universal grammar. *Science*, 275:1604–1610, 1997.

[33] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Press, 1989.

[34] N. Saito. Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. In E. Foufoula-Georgiou and P. Kumar, editors, *Wavelets in Geophysics*, pages 299–324. Academic Press, 1994.

[35] H. Schweitzer. Occam algorithms for computing visual motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:1033–1042, 1995.

[36] A. C. Shaw. A formal picture description scheme as a basis for picture processing systems. *Information and Control*, 14:9–52, 1969.

[37] S. Shieber. *Constraint-Based Grammar Formalisms.* MIT Press, 1992.

[38] P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence*, 46:159–216, 1990.

[39] C. von der Malsburg. Synaptic plasticity as a basis of brain organization. In J.P. Changeux and M. Konishi, editors, *The Neural and Molecular Bases of Learning*, pages 411–432. John Wiley and Sons, 1987.

[40] C.S. Wetherell. Probabilistic languages: a review and some open questions. *Computing Surveys*, 12:361–379, 1980.

[41] S. C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996.