Hierarchy in Machine and Natural Vision

S. Geman
Division of Applied Mathematics
Brown University
Providence, Rhode Island, USA

Abstract

I will argue that hierarchical representation and hierarchical computation are fundamental principles in biological vision systems. This fits with the theory known as compositionality, whereby human cognition is modeled as a process of assembling constituents into restricted compositions. The compositions themselves can be used and reused as constituents in a variety of still higher-level constructions, creating a hierarchy of part/object relationships. The formulation of compositionality begins with Chomsky's formal grammars. I will propose some extensions, and I will show how to fit these with probability distributions. I will explore an application to machine vision.

1 Compositionality

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 [5], [6]), 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 ([13]), these artificial networks are not compositional, and therefore they fail to mimic a basic attribute of human cognition. (See, however, von der Malsburg [35], Smolensky [34], Prince and Smolensky [28], Bienenstock [2], Hummel and Biederman [20], and Mjolsness [24] 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 ([22]), 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. [9]), as are today's cognitive scientists studying modern compositionality (see, especially, the work by Feldman [12], which connects closely with the development here).

I will outline here, through a discussion of a particular application—on-line character recognition, a possible formulation of the principle of compositionality. This is taken from a more complete and rigorous account proposed previously in [16] in collaboration with Zhiyi Chi and Daniel Potter.

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 systematically 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 ([25]), Shaw ([32]), Pavlidis ([26]), Fu ([15]), Biederman ([1]), Grenander ([17]), and Casadei & Mitter ([4]).

2 Application to On-Line Character Recognition

The best introduction is perhaps by example. I will present here a more-or-less informal introduction through a more-or-less 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

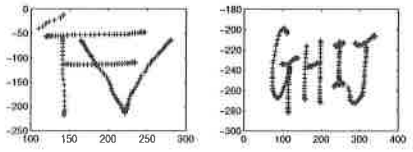
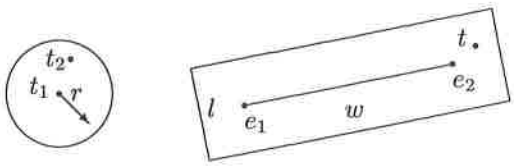


Figure 1: On-line images. Stylus position is sampled at regular intervals. Sampled locations are indicated with "+" symbol.



a b

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).

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, "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 ([33], [21]).

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, as reported in his thesis on *Compositional Pattern Recognition* ([27]), have each 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 partic-

ular 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 ([14], 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, [10] and [11], 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 ([29]). 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 is discussed further in [16], 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 [8], 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. See [16]. Here 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

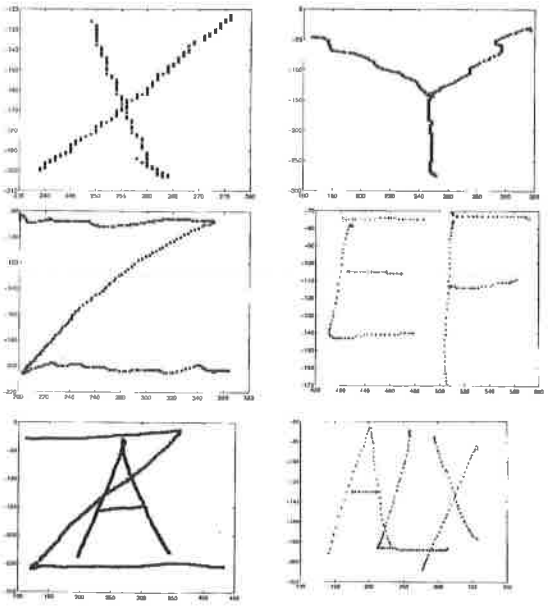


Figure 3: Examples of images interpreted by on-line character recognition algorithm.

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) and by Dan Potter (see [27]).

More levels of composition can be included in the hierarchy. 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 idea of using description lengths is not new to machine vision. In one form or another the "MDL Principle" has been applied to image segmentation (cf. Leclerc [23], Zhu and Yuille [36]), image restoration (cf. Saito [30]), motion analysis (cf. Schweitzer [31], Gu et al. [18]), and image interpretation (cf. Canning [3], Hinton et al. [19]). Our approach is in the same spirit as these, although the emphasis is on compositionality, very much along the lines proposed by Cooper (see [7]): We use description lengths to promote hierarchical aggregations of parts.

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 [16]), 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.

This opens the door to building parametric, but more-or-less generic, composition rules, and the possibility of building systems capable of learning, from example, hierarchical object and scene representations.

Acknowledgments

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.

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] 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.

[4] 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.

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

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

[7] 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.

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

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

[10] 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.

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

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

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

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

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

[16] S. Geman, D. F. Potter, and Z. Chi. Composition systems. Technical report, Division of Applied Mathematics, Brown University, 1998.

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

[18] 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.

[19] 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.

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

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

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

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

[24] 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.

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

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

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

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

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

[30] 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.

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

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

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

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

[35] 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.

[36] 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.