A Visual Language Model for Estimating Object Pose and Structure in a Generative Visual Domain

Siddharth Narayanaswamy, Andrei Barbu, and Jeffrey Mark Siskind

Abstract— **We present a generative domain of visual objects by analogy to the generative nature of human language. Just as small inventories of phonemes and words combine in a grammatical fashion to yield myriad valid words and utterances, a small inventory of physical parts combine in a grammatical fashion to yield myriad valid assemblies. We apply the notion of a language model from speech recognition to this visual domain to similarly improve the performance of the recognition process over what would be possible by only applying recognizers to the components. Unlike the context-free models for human language, our visual language models are context sensitive and formulated as stochastic constraint-satisfaction problems. And unlike the situation for human language where all components are observable, our methods deal with occlusion, successfully recovering object structure despite unobservable components. We demonstrate our system with an integrated robotic system for disassembling structures that performs whole-scene reconstruction consistent with a language model in the presence of noisy feature detectors.**

I. Introduction

Human language is *generative*:[1] a small inventory of phonemes combine to yield a large set of words and then this inventory of words combine to yield a larger set of utterances. Systems that process language must deal with the combinatorial nature of generativity. The probability of correct word recognition becomes fleetingly small with even a slight probability for error in phoneme recognition and the probability of determining the correct parse of an utterance becomes fleeting small with even a slight probability for error in word recognition. This is remedied with a *language model*, a specification of which combinations of phonemes constitute valid words and which combinations of words constitute valid utterances. Such a language model often takes the form of a *grammar*.

The vast majority of computer-vision research in pose estimation and object recognition deals with nongenerative collections of objects. Such nongenerative collections require distinct models or exemplars for each object (class) that varies greatly in shape, structure, or appearance. We instead present an approach for doing pose estimation and structure recognition in generative visual domains, analogous to the approach for human language. We illustrate this approach with the domain of LINCOLN LOG assemblies. LINCOLN LOGS is a children's assembly toy with a small component inventory. We limit this inventory to three component types: 1-notch, 2-notch, and 3-notch logs. These combine in myriad ways to yield a large set of assemblies. We present low-level feature detectors that collect evidence for the components in a fashion analogous to low-level feature detectors in speech recognizers. But as in speech, the probability of correct recognition of an entire assembly becomes fleetingly small with even a slight probability for error in log recognition. We remedy this with a *visual language model* or a *grammar* of LINCOLN LOGS, a specification of which combinations of logs constitute valid assemblies.

The analogy breaks down in two ways requiring novel methods. First, most computer models of speech and language assume that the grammar is context free. This allows a top-down tree-structured generative process where the generation of siblings is independent. In contrast, the symbolic structure underlying LINCOLN LOG assemblies takes the form of graphs with cycles and thus the visual language model is context sensitive and is formulated as a stochastic constraint-satisfaction problem. Second, in language, all of the components are observable; at least in principle, one can obtain perceptual evidence of each phoneme in a word and each word in an utterance. In contrast, visual domains exhibit *occlusion*; it is almost always necessary to determine object structure without perceptual evidence for all of the components. Our methods address both of these issues. Our work builds upon the notion that scenes and objects are represented as descriptions involving parts and spatial relations [1]–[10], differing from prior work in the extreme degree of generativity of the LINCOLN LOG domain. None of this prior work focuses on domains that can generate as large a class of distinct structures from as small a class of components. Moreover, we focus on determining the precise pose and structure of an assembly, including the 3D pose of each component, with sufficient accuracy to support robotic manipulation and, in particular, the ability to robotically construct a symbolically precise replicate of a structure from a single image.

LINCOLN LOG structures are composed out of a small inventory of components, namely 1-notch, 2-notch, and 3-notch logs. As shown in Fig. 1, such logs are characterized by a small number of shape parameters: the inter-notch distance l_1, the log diameter l_2, and the distance l_3 from a log end to the closest notch center. Valid structures contain logs arranged so that their notches are aligned and their medial axes are parallel to the work surface. Thus valid structures

http://engineering.purdue.edu/~qobi/icra2011

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47907, USA {snarayan,abarbu,qobi}@purdue.edu

[1]We mean the Chomskyan sense of generative, not the sense in contrast to discriminative. Indeed, while our domain is generative in the Chomskyan sense, our recognizer uses a discriminative model.

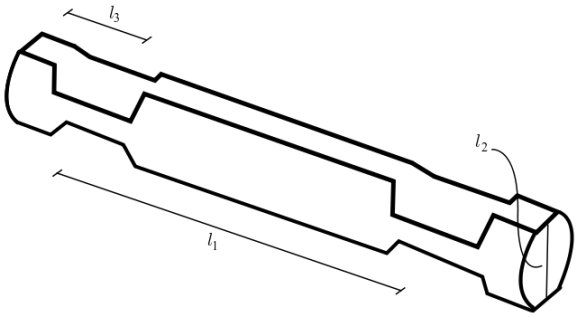


Fig. 1. The 3D geometric shape parameters of LINCOLN LOGS.

have logs on alternating layers j at height $l_2(j+0.5)$ oriented along one of two orthogonal sets of parallel lines spaced equally with horizontal distance l_1. The lines for even layers are mutually parallel, the lines for odd layers are mutually parallel, and the projections of a line from an even layer and an odd layer onto the work surface are perpendicular. We refer to this set of lines as the *grid* (see Fig. 2). This grid imposes a symbolic structure on the LINCOLN LOG assembly. Symbolic grid coordinates (i,j,k) map to metric camera-relative coordinates (x,y,z) by the parameters l_1, l_2, and l_3 together with the structure pose: the transformation from the grid coordinate system to the camera coordinate system. Estimating the structure of a LINCOLN LOG assembly thus reduces to two phases: estimating the structure pose (section II) and determining the log occupancy at each symbolic grid position (section III).

II. ESTIMATING THE STRUCTURE POSE

Before beginning these two phases, we first compute a mask that separates the LINCOLN LOG structure in the image foreground from the background. We manually collect 20–30 image segments of LINCOLN LOG components and compute the mean μ and covariance Σ of the pixel values in these segments in a five-dimensional color space UVHSI. We then derive a mask M from an input image I containing those pixels p with values whose Mahalanobis distance from μ is less than or equal to a threshold t:

$$M_p = \begin{cases} 1 & \|C(I_p) - \mu\|_\Sigma \leq t \\ 0 & \text{otherwise} \end{cases}$$

where C denotes the map from input pixel values to UVHSI.

Nominally, the structure pose contains six degrees of freedom corresponding to translation and rotation about each axis. To simplify, we assume that the structure rests on the horizontal work surface. Thus we fix vertical translation, roll around the camera axis, and pitch around the horizontal axis perpendicular to the camera axis to be zero, leaving only three free parameters: horizontal translation of the structure along the work surface and yaw around the vertical axis. To resolve the periodic translation ambiguity in the symbolic grid coordinate system, we assume that the minimum occupied i, j, and k values are zero. We further assume that we know the symbolic grid size: the maximum occupied i, j, and k values.

Images of LINCOLN LOG assemblies contain a predominance of straight edges that result from log edges. Given this, we estimate the structure pose in a two-step process. We first find the pose p that maximizes the coincidence between the set $L(p)$ of projected grid lines l_g and the set L_I of image-edge line segments l_i:

$$\operatorname*{argmin}_p \sum_{l_i \in L_I, l_g \in L(p)} \|l_i, l_g\|$$

where $\|l_i, l_g\|$ denotes the Euclidean distance between the midpoint of a line segment and its closest point on a line, weighted by the disparity in orientation between the line and the line-segment. We then refine this pose estimate by maximizing the coincidence between projected grid lines and the set P_I of image edge points p_i:

$$\operatorname*{argmin}_p \min_{p_i \in P_I, l_g \in L(p)} \|p_i, l_g\|$$

where $\|p_i, l_g\|$ denotes the Euclidean distance between a point and the closest point on the line. We use a soft min function [11]–[13] when computing the latter with gradient-based methods (reverse-mode automatic differentiation [14]).

To obtain L_I, we apply a Canny edge detector [15] together with the KHOROS line finder [16] to extract linear edge segments from the input image, discarding short segments and those that do not lie wholly within the mask region defined by M. We then select the edge segments corresponding to the two most prominent edge orientations, by placing the segments into bins according to their orientation and selecting the edge segments in the two largest bins. To obtain P_I, we apply Phase Congruency [17] to the input image I to compute the orientation image $O(I)$. Each pixel in $O(I)$ contains a quantized orientation. We chose P_I to be those pixels whose quantized orientation is closest to the mean edge-segment orientations of the above two largest bins.

This two-step process offers several advantages. The first step converges quickly but exhibits error in the recovered prominent edge orientations. The second step estimates pose more accurately (typically within 5mm translation and 2° rotation), but only with close initial estimates, such as those provided by the first step.

Fig. 2 illustrates successful pose estimation of several LINCOLN LOG structures. Note that we estimate the pose of a target object from a single image without any knowledge of the specific 3D shape or structure of that object, without any prior training images of that object in different poses, using only generic information from the domain, namely that the object is a valid LINCOLN LOG assembly.

III. DETERMINING THE LOG OCCUPANCY AT EACH SYMBOLIC GRID POSITION

The symbolic grid positions $q = (i,j,k)$ refer to points along log medial axes at notch centers. Each such grid position may be either unoccupied, denoted by $\emptyset$, or occupied with the n^{th} notch, counting from zero, of a log with m notches, denoted by (m,n). For each grid position we wish to

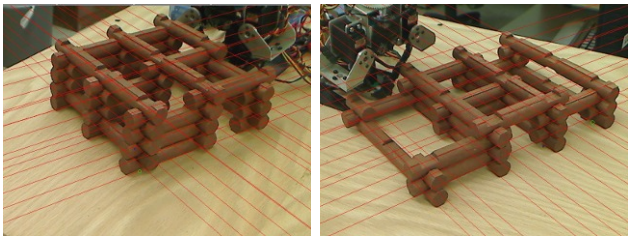


Fig. 2. Estimating the pose of an arbitrary LINCOLN LOG assembly and the symbolic grid thus imposed on the assembly.

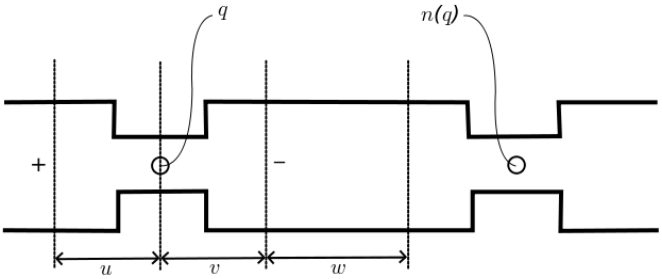


Fig. 3. The random variables Z_q^+ and Z_q^- that correspond to log ends for grid position q and the random variables Z_q^u, Z_q^v, and Z_q^w that correspond to log segments.

determine its occupancy, one of seven possibilities: $\emptyset$, $(1,0)$, $(2,0)$, $(2,1)$, $(3,0)$, $(3,1)$, and $(3,2)$. We construct a discrete random variable Z_q for each grid position q that ranges over these seven possibilities.

We determine several forms of image evidence for the log occupancy of a given grid position. LINCOLN LOGS, being cylindrical structures, generate two predominant image features: ellipses that result from the perspective projection of circular log ends and line segments that result from the perspective projection of cylindrical walls. We refer to the former as *log ends* and the latter as *log segments*. Log ends can potentially appear only at distance $\pm l_3$ from grid positions along the direction for the layer of that grid position. We construct boolean random variables Z_q^+ and Z_q^- to encode the presence or absence of a log end at such positions. There are two kinds of log segments: ones corresponding to l_1 and ones corresponding to l_3. Given this, we construct three boolean random variables Z_q^u, Z_q^v, and Z_q^w for each grid position q that encode the presence or absence of log segments for the bottoms of logs, i.e., log segments between a grid position and the adjacent grid position below. Z_q^u and Z_q^v encode the presence or absence of a log segment of length l_3 behind and ahead of q respectively, along the direction for the layer of q while Z_q^w encodes the presence or absence of a log segment of length $l_1 - 2l_3$ between grid positions along the same layer. Fig. 3 depicts the log ends and log segments that correspond to a given grid position as described above.

We formulate a stochastic constraint-satisfaction problem (CSP [18]) over these random variables. The constraints encode the validity of an assembly. We refer to these constraints as the *grammar* of LINCOLN LOGS (section III-C). We take image evidence to impose priors on the variables Z_q^+, Z_q^-, Z_q^u, Z_q^v, and Z_q^w (sections III-A and III-B) and solve this stochastic CSP to perform structure estimation (section III-D).

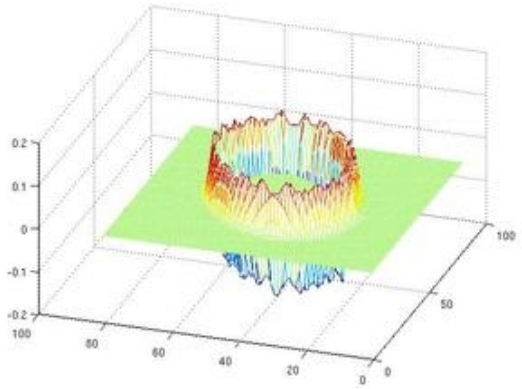


Fig. 4. Elliptical edge filter for detecting log ends

A. Evidence for the presence or absence of logs

Given the pose p, a log end present as the result of Z_q^+ or Z_q^- being true will manifest as an ellipse of known shape, size, and position in the image. We use $x^+(p,q)$, $y^+(p,q)$, $a^+(p,q)$, $b^+(p,q)$, and $\theta^+(p,q)$ to denote the parameters (center, lengths of major and minor axes, and orientation of major axis) of an ellipse that would manifest from Z_q^+ and similarly for Z_q^-. We find these parameters by a least-squares fit of 20 equally spaced 3D points on the log end projected to the image. The 3D points can be determined in closed form from the grid position q and the parameters l_1, l_2, and l_3. We then construct an indicator function $f(x,y)$ with the value 1 for points (x,y) inside the ellipse and the value 0 for points outside the ellipse and convolve this with a Laplacian of a Gaussian filter, $\text{LoG}(r,\sigma)$, to obtain an elliptical edge filter $E(x,y,a,b,\theta)$ (Fig. 4). Nominally, a high response to this filter applied to an image correlates with the presence of an elliptical feature with parameters x, y, a, b, and θ. To provide robustness in the face of inaccurate pose estimation, we compute the maximal filter response in a 5-dimensional region centered on x, y, a, b, and θ derived by perturbing each axis a small amount.

Similarly, given the pose p, a log segment present as the result of Z_q^u, Z_q^v, or Z_q^w being true will manifest as a line segment between known image points. We denote the points for Z_q^u as $(x_1^u(p,q), y_1^u(p,q))$ and $(x_2^u(p,q), y_2^u(p,q))$ and similarly for Z_q^v and Z_q^w. These image points can be determined in closed form by projecting the 3D points derived from the pose p, the grid position q, and the parameters l_1, l_2, and l_3.

In principle, we could use a similar filter method to determine evidence for log segments. However, log ends usually yield highly pronounced edges because logs are never stacked horizontally end to end. Log are often stacked vertically and the log segments between two such vertically stacked logs would yield less-pronounced edges. Thus we use a more sensitive method to determine evidence for log segments. Given the pose p of the structure, we recompute the prominent edge orientations o_1 and o_2 using the methods from section II (this time applied to the output of the

second step of pose estimation, not the first, to give a more accurate estimate of these orientations). For each prominent orientation o, we compute the disparity between o and $O(I)$ at each pixel, compute the prominence at each pixel by attenuating the disparity, and scale the energy image, $E(I)$, by this prominence: $W(I,o) = E(I) \circ \cos^2(O(I) - o)$. This constitutes a graded edge map for edges with orientation o. We search a rectangular region in $W(I,o)$, after thresholding, for the longest line segment. The search region corresponds to a dilation of the rectangle bounded by the endpoints of the target log segment. The length of the longest line segment found correlates with the presence of the target log segment.

B. Mapping evidence to priors

We train a mapping function from evidence to priors for the log-segment and log-end evidence functions respectively on a set of 30 images annotated with ground truth, i.e., true positives and true negatives, along with occlusion. For each evidence function, we bin their respective raw, real-valued responses into 20 bins and annotate each bin with the percentage of responses that are true positives and the central response value for that bin. The annotated bins correspond to a discrete sequence of impulses with impulse magnitude representing the percentage of true positives for the central response value. We then employ a weighted linear interpolation function between impulses to provide the mapping function. The weighting factor e typically takes the form of a real value $e \in (0,1)$.

C. The grammar of Lincoln Logs

We refer to the adjacent grid position below q as $b(q)$, the adjacent grid position further from the origin along the direction of the grid lines for the layer of q as $n(q)$, and the adjacent grid position closer to the origin along the direction of the grid lines for the layer of q as $p(q)$. Ignoring boundary conditions at the perimeter of the grid, the grammar of LINCOLN LOGS can be formulated as the following constraints:

a) 2-notch logs occupy two adjacent grid points

$$Z_q = (2,0) \leftrightarrow Z_{n(q)} = (2,1)$$

b) 3-notch logs occupy three adjacent grid points

$$\begin{aligned} Z_q = (3,0) &\leftrightarrow Z_{n(q)} = (3,1) \\ Z_q = (3,0) &\leftrightarrow Z_{n(n(q))} = (3,2) \\ Z_{n(q)} = (3,1) &\leftrightarrow Z_{n(n(q))} = (3,2) \end{aligned}$$

c) 1- and 2-notch logs must be supported at all notches

$$Z_q \in \{(1,0),(2,0),(2,1)\} \rightarrow Z_{b(q)} \neq \emptyset$$

d) 3-notch logs must be supported in at least 2 notches

$$Z_q = (3,0) \rightarrow \begin{pmatrix} (Z_{b(q)} \neq \emptyset \wedge Z_{b(n(q))} \neq \emptyset) \vee \\ (Z_{b(q)} \neq \emptyset \wedge Z_{b(n(n(q)))} \neq \emptyset) \vee \\ (Z_{b(n(q))} \neq \emptyset \wedge Z_{b(n(n(q)))} \neq \emptyset) \end{pmatrix}$$

e) log ends must be at the ends of logs

$$\begin{aligned} Z_q^- &\leftrightarrow Z_q \in \{(1,0),(2,0),(3,0)\} \\ Z_q^+ &\leftrightarrow Z_q \in \{(1,0),(2,1),(3,2)\} \end{aligned}$$

f) short log segments indicate occupancy above or below

$$\begin{aligned} Z_q^u &\leftrightarrow (Z_q \neq \emptyset \vee Z_{b(b(q))} \neq \emptyset) \\ Z_q^v &\leftrightarrow (Z_q \neq \emptyset \vee Z_{b(b(q))} \neq \emptyset) \end{aligned}$$

g) long log segments indicate presence of a multi-notch log above or below

$$Z_q^w \leftrightarrow \begin{pmatrix} \begin{pmatrix} Z_q \in \{(2,0),(3,0),(3,1)\} \wedge \\ Z_{n(q)} \in \{(2,1),(3,1),(3,2)\} \end{pmatrix} \vee \\ \begin{pmatrix} Z_{b(b(q))} \in \{(2,0),(3,0),(3,1)\} \wedge \\ Z_{b(b(n(q)))} \in \{(2,1),(3,1),(3,2)\} \end{pmatrix} \end{pmatrix}$$

To handle the boundary conditions, we stipulate that the grid positions beyond the perimeter are unoccupied, enforce the support requirement (constraints c–d) only at layers above the lowest layer, and enforce log-segment constraints (f–g) for the layer above the top of the structure.

D. Structure estimation

To perform structure estimation we first establish priors over the random variables Z_q^+ and Z_q^- that correspond to log ends and the random variables Z_q^u, Z_q^v, and Z_q^w that correspond to log segments using image evidence and establish a uniform prior over the random variables Z_q. This induces a probability distribution over the joint support of these random variables. We then marginalize the random variables that correspond to log ends and log segments and condition this marginal distribution on the language model Φ. Finally, we compute the assignment to the random variables Z_q that maximizes this conditional marginal probability.

$$\underset{\mathbf{Z}}{\operatorname{argmax}} \sum_{\substack{\mathbf{Z}^+,\mathbf{Z}^-,\mathbf{Z}^u,\mathbf{Z}^v,\mathbf{Z}^w \\ \Phi[\mathbf{Z},\mathbf{Z}^+,\mathbf{Z}^-,\mathbf{Z}^u,\mathbf{Z}^v,\mathbf{Z}^w]}} \Pr\left(\bigwedge_q Z_q, Z_q^+, Z_q^-, Z_q^u, Z_q^v, Z_q^w\right)$$

To speed up the conditional marginalization process, we prune assignments to the random variables that violate the grammar Φ using arc consistency [19]. To speed up the maximization process, we use a branch-and-bound algorithm [20] that maintains upper and lower bounds on the maximal conditional marginal probability. Without both of these, structure estimation would be intractable.

An alternate method to perform structure optimization is to establish the same priors over the random variables that correspond to log ends and log segments but parametrize the priors over the random variables Z_q. We then marginalize over all random variables, computing this marginal probability over the parameterized priors for the random variables Z_q. We then search over this parameter space for the distributions over the random variables Z_q that maximize this marginal probability. We do this using the reduced-gradient optimization algorithm [21], [22] where the gradients are calculated using reverse-mode AD. The linear constraints are used to constrain the parameters of the probability distribution to be nonnegative and sum to one. Ideally, we'd prefer to use the latter method exclusively, but the former method is faster to compute for the relatively larger assemblies when compared to the latter.

E. Occlusion

Nominally, with the above method, one derives evidence for the presence or absence of log ends and log segments of the various kinds at every possible grid position. In other words, one uses image evidence to impose a prior on all of the random variables Z_q^+, Z_q^-, Z_q^u, Z_q^v, and Z_q^w. However, some of these log ends and log segments may be occluded. If we *know* that a log end or log segment is occluded then we ignore all evidence for it from the image, giving it chance probability of being occupied. With this, the grammar can often fill in the correct values of occluded random variables for both log ends and log segments, and thus determine the correct value for an occluded Z_q. The question then arises: how does one determine whether a log end or log segment is occluded? We propose the following method. One first assumes that all of the log ends and log segments on the frontal faces of the grid are visible but all other log ends and log segments are occluded. One then performs structure estimation under this initial hypothesis. With the recovered structure estimate, one determines log-end and log-segment visibility by projective geometry given the known pose, and iterates this process until convergence. We have recently implemented this algorithm and expect to report on its performance in the future. All experiments reported in section IV were performed with manual annotation of occlusion information. Note that we only annotate for a given symbolic log-segment or log-end *position* whether or not it is *visible*, **not** whether or not that position is *occupied* with a log segment or log end. The latter is determined automatically.

IV. Experimental results

We took images of 32 distinct Lincoln Log structures, each from 5 distinct poses resulting in a total of 160 images. We performed foreground-background separation and pose estimation for all 160 images using the methods from section II. Pose was estimated within 5mm translation and 2° rotation of ground truth for 142 images. We discarded the 18 images with inaccurate pose estimation and performed structure estimation on the remainder. The results for 5 images, all of distinct structures, are shown in Fig. 7. Fig. 7(a) was derived by thresholding the priors on Z_q^+, Z_q^-, Z_q^u, Z_q^v, and Z_q^w at $t = 0.5$. Fig. 7(b–d) were derived by solving a stochastic CSP with various subsets of the constraints and rendering the values of Z_q^+, Z_q^-, Z_q^u, Z_q^v, and Z_q^w for the solution provided by the first method in section III-D. Fig. 7(e) was derived by solving the stochastic CSP with all constraints and rendering the values of Z_q for the solution provided by the first method in section III-D. Note that our method determines the correct component type (Z_q) of most occluded logs in the assemblies in the second row of Fig. 7(e). It gives an incorrect component type for only a single log in that row.

We conducted experiments to determine how much the grammar improves the accuracy of structure estimation. We performed variants of the runs in Fig. 7(a-d), varying the threshold t and the mapping from evidence to priors to produce the ROC curves depicted in Fig. 5. The mapping

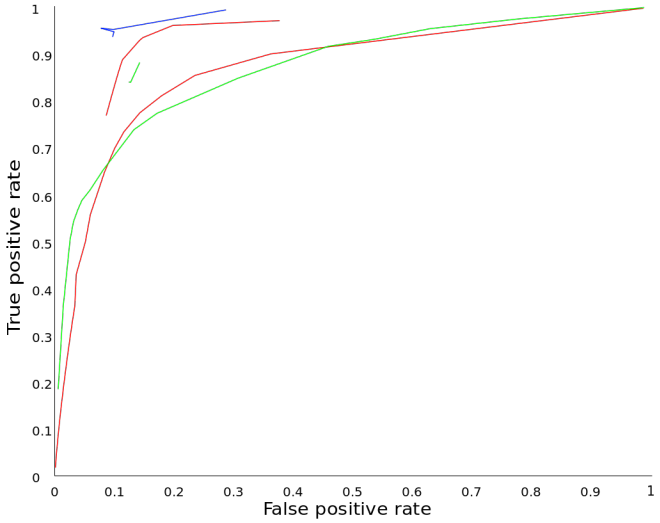


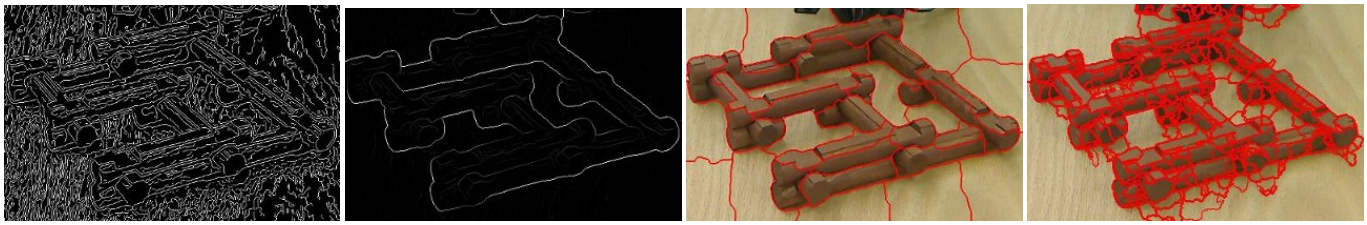
Fig. 5. ROC curves. The lower green and red curves constitute the ROC for the log-end and log-segment detectors respectively with varying thresholds t without the grammar. The upper green curve measures ROC for Z_q^+ and Z_q^- under constraints a–e varying the mapping from evidence to priors. The upper red curve measures ROC for Z_q^u, Z_q^v, and Z_q^w under constraints a–d and f–g varying the mapping from evidence to priors. The blue curve measures ROC for Z_q^+, Z_q^-, Z_q^u, Z_q^v, and Z_q^w under all constraints varying the mapping from evidence to priors.

function is varied through the weighting factor e for the linear interpolator discussed in section III-B.

Pose and structure estimation is sufficiently robust to support robotic manipulation. Supplementary material included on the website for this paper contains videos of fully autonomous robotic disassembly of six different Lincoln Log structures whose pose and structure have been determined from a single image as well as videos of semiautonomous robotic assembly of replicate Lincoln Log structures from the same estimated pose and structure.

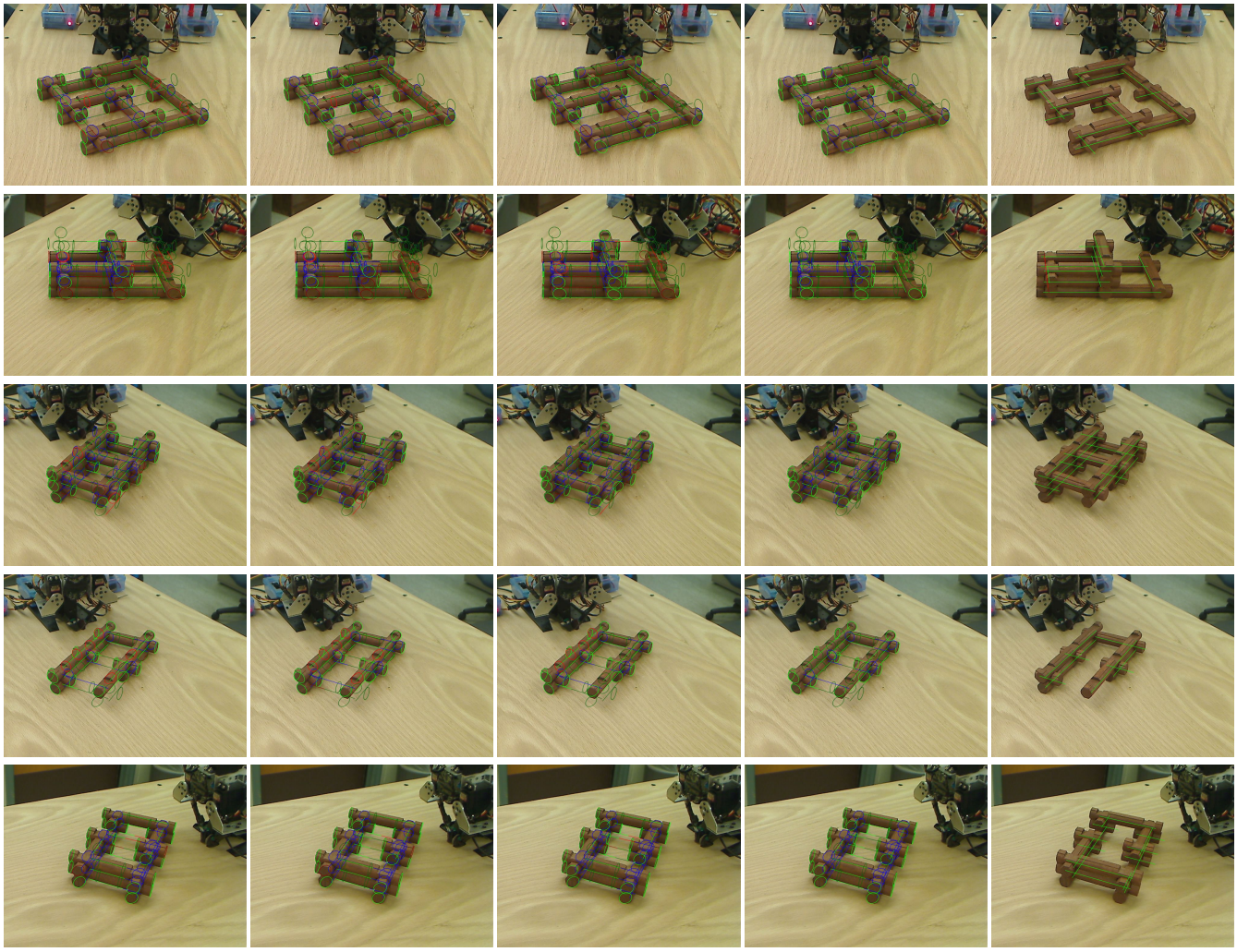
V. Conclusion

Lincoln Logs are children's toys yet the computational problem we present is *not* a toy. Pose and structure estimation of Lincoln Log assemblies is *far* more difficult than may appear on the surface. The space of objects to be recognized is combinatorially large. Much of every structure is in self occlusion. The low contrast due to shadows and color, intensity, and texture uniformity make it impossible to recognize even *visible* logs with existing techniques. No standard edge detector (e.g., Canny [15] or Pb [23]) can reliably find edges separating adjacent logs or circular log ends and no standard segmentation method (e.g., Normalized Cut [24] or Mean Shift [25]) can reliably find log parts *even when fully visible* as shown in Fig. 6. Even our filter-based feature detectors, which use pose information along with constraints from the language model to *tune to the expected feature at the expected image position*, produce correct binary decisions only about 65% of the time. Occlusion only makes matters worse. Performing non-stochastic constraint satisfaction (e.g., Waltz line labeling [26]) on the binary



(a) (b) (c) (d)

Fig. 6. A comparison with a number of standard edge detectors and segmentation methods. Neither (a) MATLAB's Canny edge detector nor (b) the PB edge detector reliably find edges separating adjacent logs or log ends. Neither (c) Normalized Cut nor (d) Mean Shift segment out the log parts.



(a) (b) (c) (d) (e)

Fig. 7. (a) Raw detector response. (b) Detector response with just constraints a–d and f–g. (c) Detector response with just constraints a–e. (d) Detector response with all constraints. (e) Estimated structure. In (a–d), bright red indicates true negative, dark red indicates false negative, bright green indicates true positive, dark green indicates false positive, and blue indicates occlusion. In (e), green indicates true positive and red indicates false negative. There are no false positives and true negatives are not indicated. We suggest that the reader view this figure at a high magnification level in a PDF viewer to appreciate the images.

output of these detectors leads to inconsistent CSPs on *all* images in our dataset.

We have demonstrated a visual domain that is generative in much the same way that human language is generative. We have presented a visual language model that improves recognition accuracy in this domain in much the same way that language models improve speech-recognition accuracy. Unlike context-free models of human language, our visual language models are context sensitive and formulated as stochastic CSPs. Much of our visual experience in the artifactual world is perceiving generative man-made structures like buildings, furniture, vehicles, etc. Our LINCOLN LOG domain is a first step towards building visual language models for such real-world domains.

Language models for vision are more complex than those for human language as they must deal with occlusion resulting from perspective projection and pose variation. However, visual domains exhibit a novel possibility: recovering structure despite occlusion by integrating the perceptual evidence from multiple images of the same object taken from different poses. In the LINCOLN LOG domain, one can carry this even further. When faced with ambiguity arising from occlusion, a robot can partially disassemble a structure to view occluded substructure and integrate perceptual evidence from multiple images taken at different disassembly stages to yield a complete unambiguous estimate of the structure of the original assembly prior to disassembly. Moreover, it is possible to integrate information about pose or structure from different modalities. One can integrate partial pose and structure information from one or more images with partial pose and structure information expressed in human language to yield a complete unambiguous estimate of pose and structure. We are, in fact, able to do this and expect to report on this in the future.

VI. ACKNOWLEDGMENTS

This work was supported, in part, by NSF grant CCF-0438806, by the Naval Research Laboratory under Contract Number N00173-10-1-G023, by the Army Research Laboratory accomplished under Cooperative Agreement Number W911NF-10-2-0060, and by computational resources provided by Information Technology at Purdue through its Rosen Center for Advanced Computing. Any views, opinions, findings, conclusions, or recommendations contained or expressed in this document or material are those of the author(s) and do not necessarily reflect or represent the views or official policies, either expressed or implied, of NSF, the Naval Research Laboratory, the Office of Naval Research, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

REFERENCES

[1] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 200, no. 1140, pp. 269–94, 1978.

[2] I. Biederman, "Recognition-by-components: A theory of human image understanding." *Psychological review*, vol. 94, no. 2, pp. 115–47, 1987.

[3] W. Wang, I. Pollak, T.-S. Wong, C. A. Bouman, M. P. Harper, and J. M.Siskind, "Hierarchical stochastic image grammars for classification and segmentation," *IEEE Trans. on Image Processing*, vol. 15, pp. 3033–52, 2006.

[4] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2006.

[5] J. M. Siskind, J. S. Jr., I. Pollak, M. P. Harper, and C. A. Bouman, "Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1504–19, 2007.

[6] L. L. Zhu, Y. Chen, and A. Yuille, "Unsupervised learning of a probabilistic grammar for object detection and parsing," in *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

[7] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *Proceedings of the 10th European Conference on Computer Vision*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 30–43.

[8] M. Aycinena Lippow, L. P. Kaelbling, and T. Lozano-Perez, "Learning grammatical models for object recognition," in *Logic and Probability for Scene Interpretation*, ser. Dagstuhl Seminar Proceedings, no. 08091, Dagstuhl, Germany, 2008.

[9] V. Savova and J. Tenenbaum, "A grammar-based approach to visual category learning," in *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, 2008.

[10] V. Savova, F. Jäkel, and J. Tenenbaum, "Grammar-based object representations in a scene parsing task," in *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*, 2009.

[11] S. Smale, "Algorithms for solving equations," in *Proceedings of the International Congress of Mathematicians*, 1986, pp. 172–95.

[12] B. Chen and P. T. Harker, "A non-interior-point continuation method for linear complementarity problems," *SIAM Journal of Matrix Analysis Applications*, vol. 14, no. 4, pp. 1168–90, 1993.

[13] C. Kanzow, "Some noninterior continuation methods for linear complementarity problems," *SIAM Journal of Matrix Analysis Applications*, vol. 17, no. 4, pp. 851–68, 1996.

[14] B. Speelpenning, "Compiling fast partial derivatives of functions given by algorithms," Ph.D. dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, Jan. 1980.

[15] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–98, 1986.

[16] K. Konstantinides and J. R. Rasure, "The Khoros software development environment for image and signal processing," *IEEE Trans. on Image Processing*, vol. 3, pp. 243–52, 1994.

[17] P. Kovesi, "Image features from phase congruency," *Videre: A Journal of Computer Vision Research*, vol. 1, no. 3, 1999.

[18] J.-L. Lauriere, "A language and a program for stating and solving combinatorial problems," *Artificial Intelligence*, vol. 10, no. 1, pp. 29–127, 1978.

[19] A. K. Mackworth, "Consistency in networks of relations," *Artificial Intelligence*, vol. 8, no. 1, pp. 99–118, 1977.

[20] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960.

[21] P. Wolfe, "The reduced gradient method," Jun. 1962, unpublished.

[22] ——, "Methods of nonlinear programming," in *Nonlinear Programming*, J. Abadie, Ed. Interscience, John Wiley, 1967, ch. 6, pp. 97–131.

[23] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, "Using contours to detect and localize junctions in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2008, pp. 1–8.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[25] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–19, 2002.

[26] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision*, P. Winston, Ed. McGraw-Hill, 1975, pp. 19–91.