A Stochastic Grammar of Images

Paper by SC Zhu and David Mumford

-Barun Das

-Atul Kumar

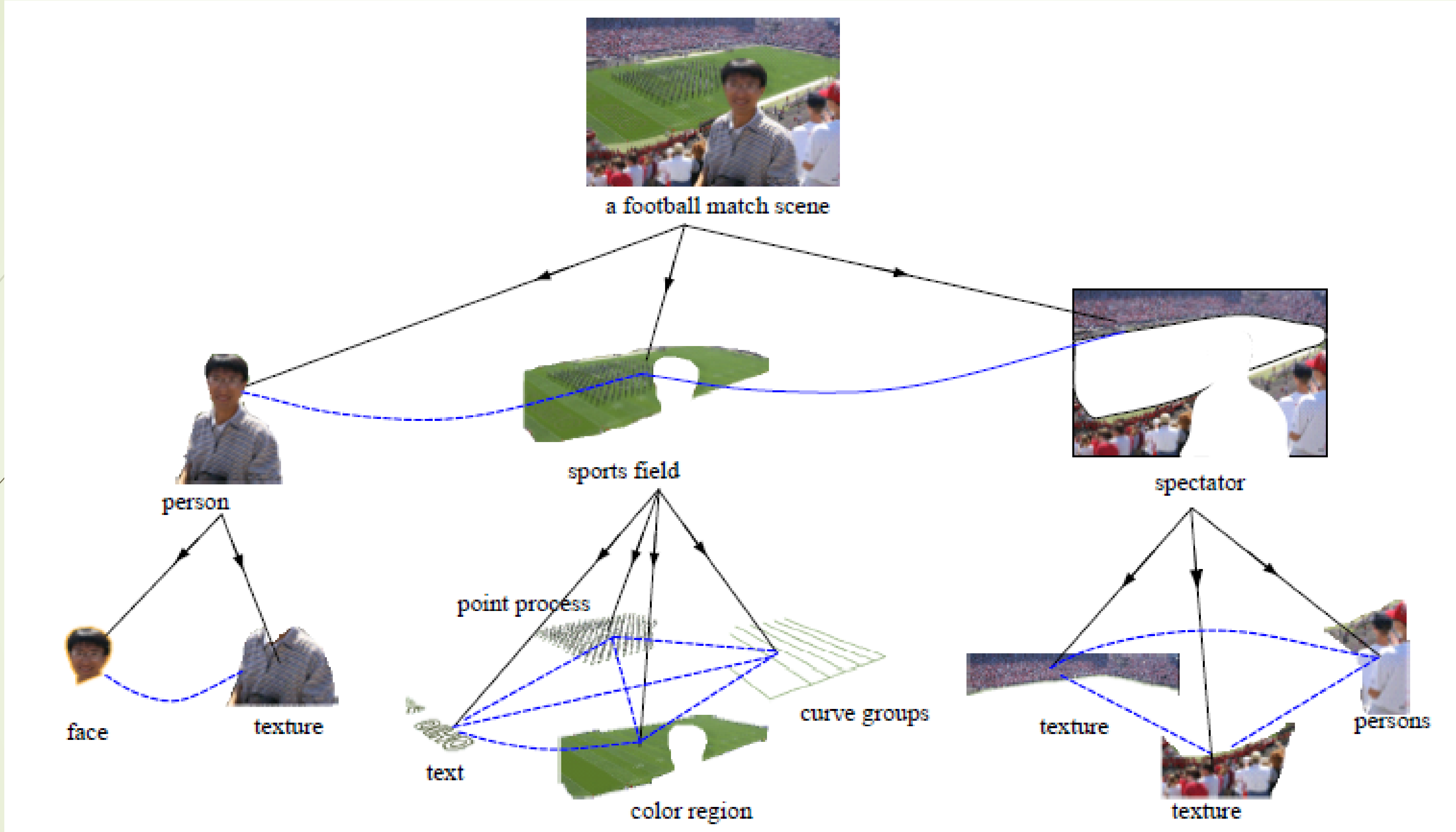


Figure 1: Illustrating the task of image parsing. The parse graph includes a tree structured decomposition in vertical arrows and a number of spatial and functional relations in horizontal arrows.

Objectives

- A common framework for visual knowledge representation and object categorization.
- Scalable and recursive top-down/bottom-up computation.
- Small sample learning and generalization.
- Mapping the visual vocabulary to fill the semantic gap – the grammar includes a series of visual dictionaries, which must have the following properties:
  - Elements organized through graph composition
  - The instances of each node can occur at any size.

Initial Problems

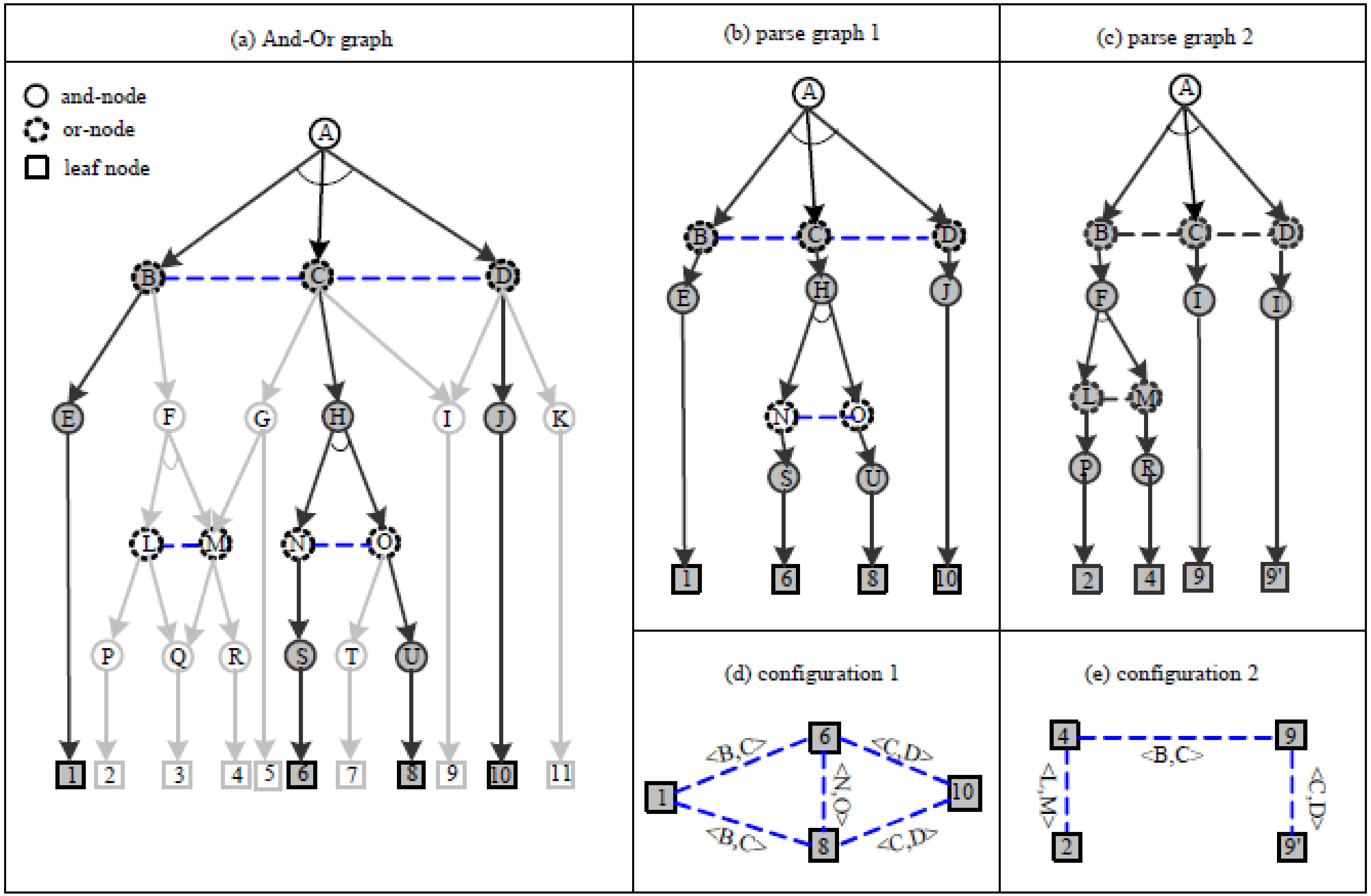
Image grammars were a dormant field until recently

- Huge amount of visual information to be represented.
- Huge computational complexity.
- Semantic gap between raw pixels and the symbolic token representation in early syntactic and structured methods.

Overview of Image Grammar – Representational Concepts and Data Structures

- And-Or Graph → And nodes, Or nodes and terminal nodes.
  - And nodes = decomposition of an entity into its parts → grammar rules.
  - Or nodes = "switches" for alternative sub-structures → labels of classification at various levels.
- Parse Graph: *a hierarchic generative interpretation of a specific image* → derived from the And-Or graph.
- Configuration: *a planar attribute graph formed by linking the open bonds of the primitives in the image plane.*
  - Inherits the relations from its ancestors (Markov networks with reconfigurable neighbourhoods).
  - Mixed random field model.

- Visual vocabulary: Each terminal node takes information from a certain set called dictionary – contains image patches of various complexities.
- Language of a grammar: *set of all possible valid configurations produced by the grammar.*
  - The sub-language for any node A is the set of all valid configurations produced by the And-Or graph rooted at A.
- Configuration(language element) → made up of several atomic structures (dictionary elements).
  - Zoomed out configuration = dictionary element.



Illustrating the And-Or graph representation. (a) An And-Or graph embodies the grammar productions rules and contexts. It contains many parse graphs, one of which is shown in bold arrows. (b) and (c) are two distinct parse graphs by selecting the switches at related Or-nodes. (d) and (e) are two graphical configurations produced by the two parse graphs respectively. The links of these configurations are inherited from the And-or graph relations.

Overview of Image Grammar – Data Set and Learning

- Image grammar is learned in a semi-automatic way: supervised learning + weakly supervised learning. Purely unsupervised learning is impractical:
  - Visual learning should be guided by objectives of vision, not purely statistical information.
  - In almost all unsupervised learning methods, human trainers have to select data to ensure proper contrasts.
- Learning steps guided by a minmax entropy learning scheme and maximum likelihood estimation. :
  - Learning the probabilities at the Or-node
  - Learning and pursuing the Markov models on the horizontal links and relations.
  - Learning the And-Or graph structures and dictionaries.

Advantages of stochastic context sensitive grammar (SCSG):

- Compositional power for representing large intra-class structural variations.
- Recursive structures for scalable computing.
- Small sample for effective learning.

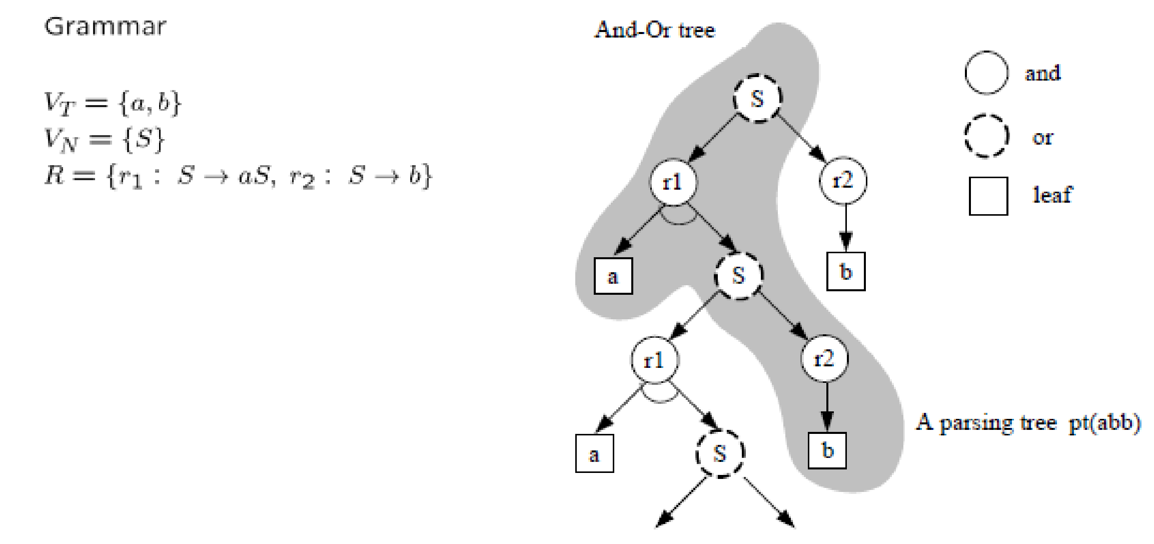
Origin of Grammars

- Vocabulary of reusable parts → certain parts of a signal occur together more frequently.
- Two parts of a signal are bound if the probability of their co-occurrence is significantly greater than the probability if their occurrence was independent.
- Parse graph: The set of such parts which one encounters in analysing statistically a specific signal.
- Grammar is powerful because it can generate a large number of valid sentences from a small set of vocabulary.
- The set of all possible strings of terminals ω derived from a grammar G is called its language.

Traditional Formulation of Grammar

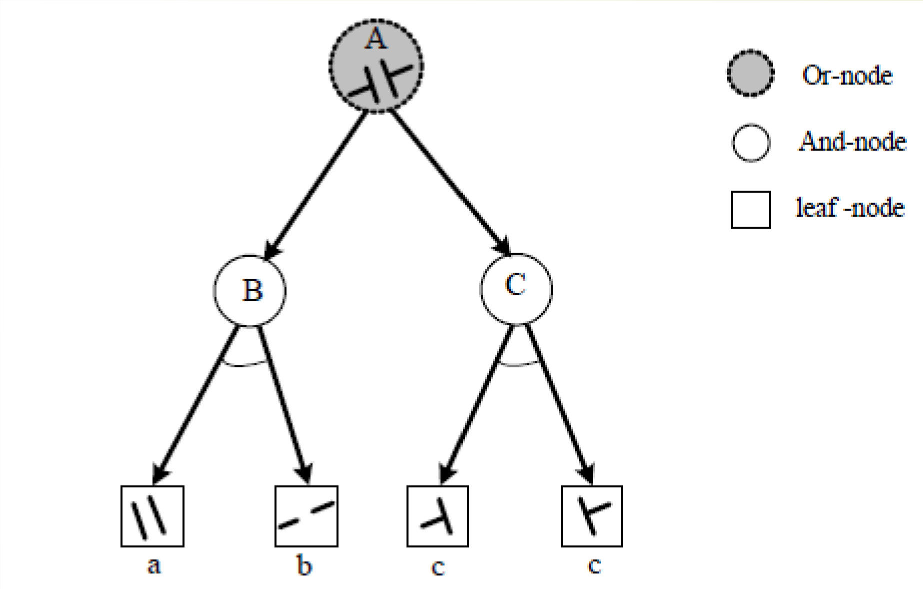
- Grammar is a 4-tuple G = $(V_N; V_T; \mathrm{R}; S)$.
- Grammar rules represent both structural regularity and flexibility.
- Entire grammar described using an And-Or tree which contains all parsings as subtrees. The Or-nodes are labelled by $V_N \, U \, V_T$ and the And-nodes are labelled by production rules R.
- Tree is generated recursively, starting from S (Or node).
- Wherever we have an Or-node with non-terminal label A, we consider all rules which have A on the left and create children which are And-nodes labelled by the corresponding rules. These in turn expand to a set of Or-nodes labelled by the symbols on the right of the rule.

A very simple grammar, its universal And-Or tree and a specific parse tree in shadow



Grammar

$V_T = \{a, b\}$
$V_N = \{S\}$
$R = \{r_1 : S \to aS,\ r_2 : S \to b\}$

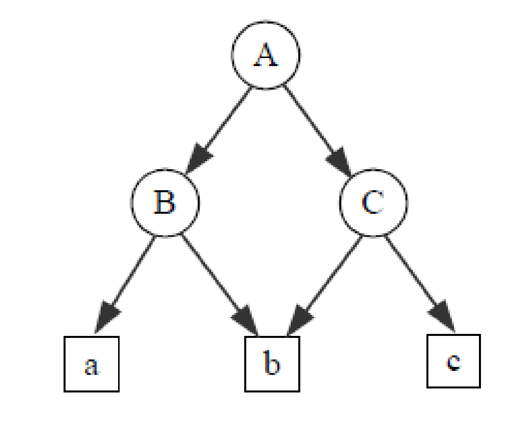


Overlapping Reusable Parts

If two reusable parts overlap, typically this leads to parse structures with a diamond in them.

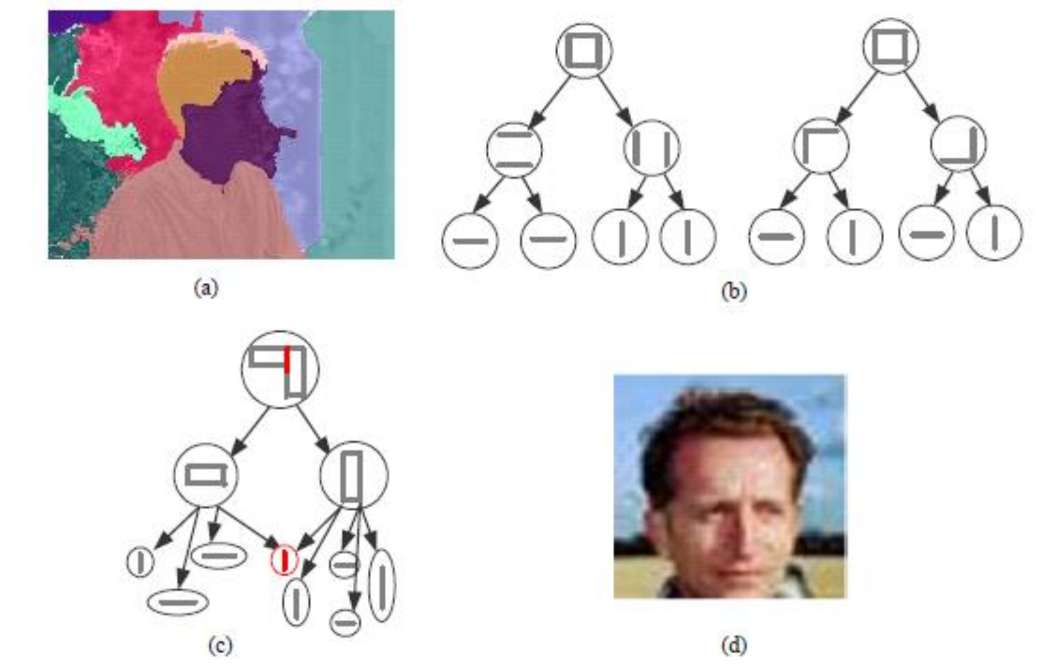
If there exists a string ω ϵ L(G) that has more than one parse tree, then G is said to be an ambiguous grammar.

- Ambiguous scenes where distinct parses suggest themselves.
- High level patterns which incorporate multiple partial patterns.
- ‘joints’ between two high level parts where some sharing of pixels or edges occurs.
- Occlusion.



Parts sharing and the diamond structure in And-Or graphs.

Ambiguities in vision

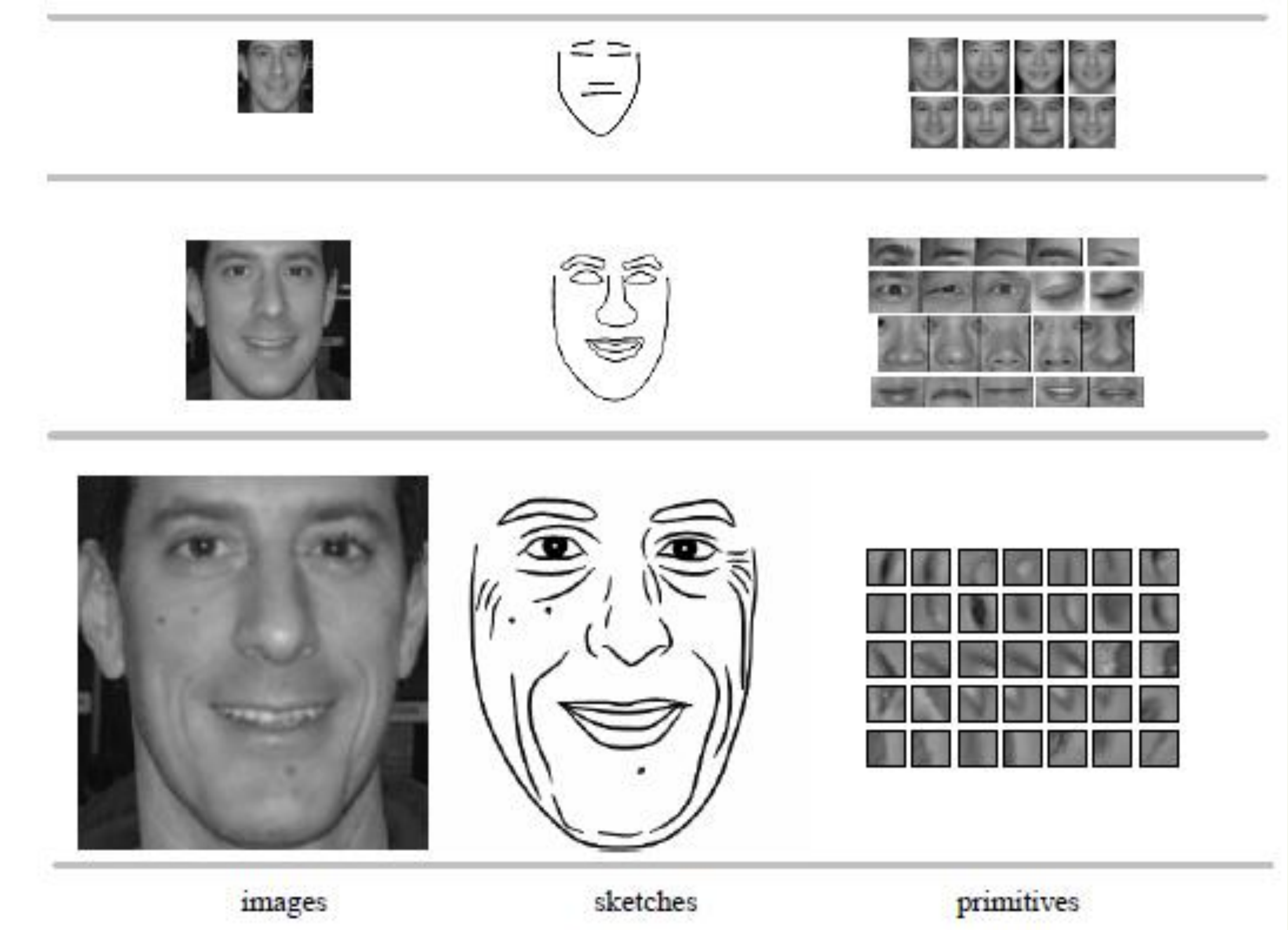


Stochastic Grammar with Context

- To connect with real world signals, we augment grammars with a set of probabilities $\mathcal{P}$ as a fifth component. Each production rule is associated with a probability. Therefore a stochastic grammar $\mathcal{G} = (V_N; V_T ;R; S;\mathcal{P})$ produces a probability distribution on its language.
- And-Or Tree $\xrightarrow{\text{Relations and context (horizontal links)}}$ And-Or graph.
- Example in language: bigram statistics + parse tree model (probabilistic model).
- In language, left to right sequence of words may not express the strongest contextual meaning.
  - Hidden relations, cannot be deterministically identified.
  - Represented by 'address variables' associated with each node.
  - The value of an address variable in a node ω_i is an index towards another node ω_j , and the node pair (ω_i, ω_j) observes a certain relation.
- In vision, these non-local relations occur much more frequently.
  - Represent the spatial context at all levels of vision from pixels and primitives to parts, objects and scenes, and lead to various graphical models, such as Markov random fields.
  - Example: occlusion, background object supporting foreground object.

Issues in Image Grammar in contrast to Language

- Going from 1-D language to 2-D grammar involves technological complexities, although the principles are the same.
  - **Loss of left-to-right ordering:** In language, every production rule A -> β_i is assumed to generate a linearly ordered sequence of nodes β. Following this down to the leaves, we get a linearly ordered sequence of terminal words. In vision, we have to replace these implicit links by the edges of a more complex `region adjacency graph‘.
  - **Image scaling:** objects appear at arbitrary scales in an image when the 3D object lies nearer or farther from the camera. Sentences in language do not occur at multiple scales.
  - **Wider spectrum of irregular local patterns:** Images contain highly structured objects which can be decomposed by production rules as well as clutter and texture (stochastic patterns) which are better represented by Markov random field models.



A human face at different levels of details and their decompositions

Previous Work in Image Grammars

- Syntactic pattern recognition (late 1970s – early 1980s)
- Medial axis techniques (for analysing 2D shapes)
- Pattern theory (defined a regular pattern on a set of graphs which are made from some primitives which he called “generators". Each generator is like a terminal element and has a number of attributes and “bonds" to connect with other generators).
- Sparse image coding model (can be viewed as an attribute stochastic context free grammar).

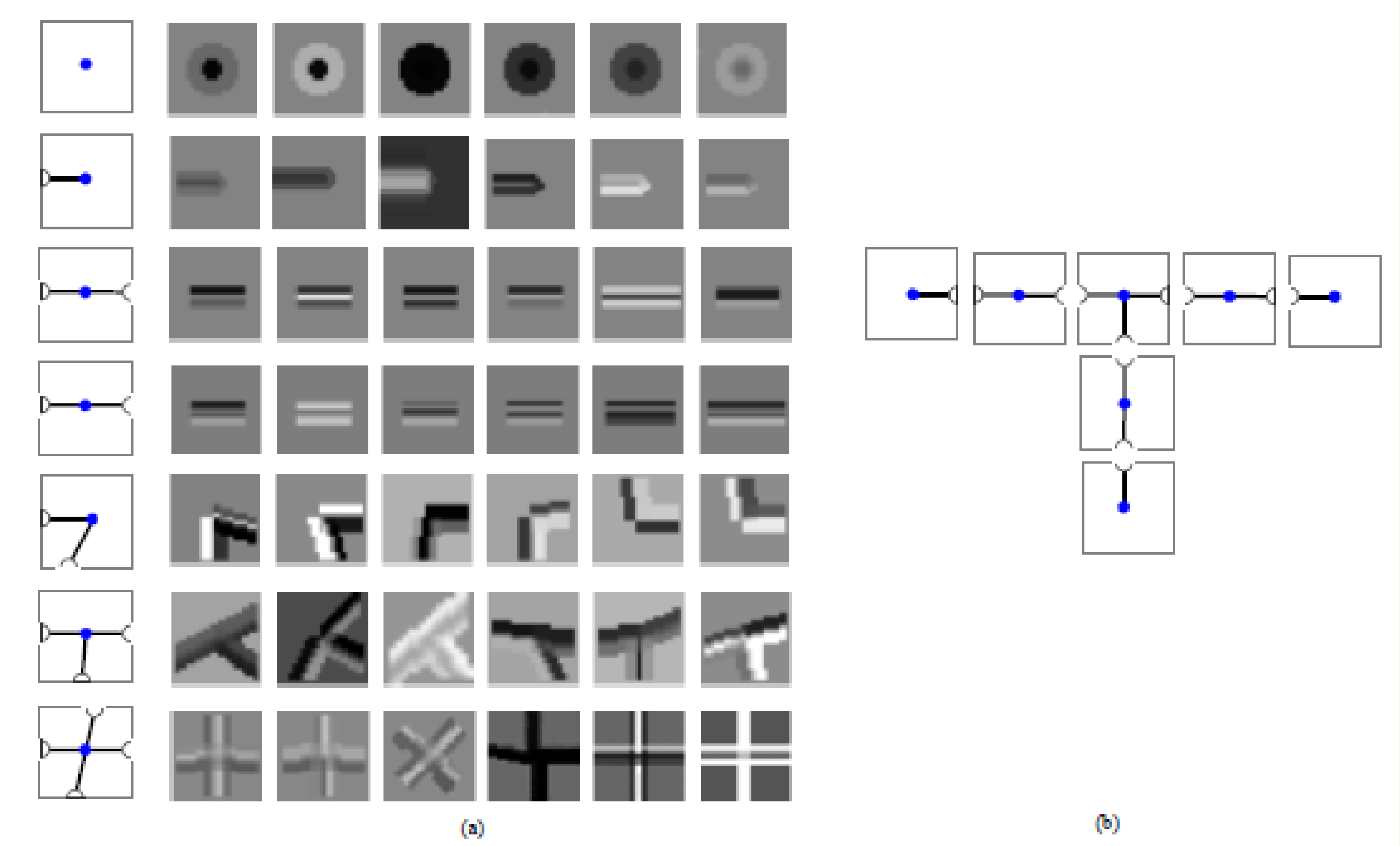
Visual Vocabulary

- As an image grammar must adopt a multi-resolution representation, the elements in its vocabulary represent visual concepts at all levels of abstraction and complexity.
- Visual vocabulary: *a set of pairs, each consisting of an image function* $\varphi_i(x, y; \alpha_i)$ *and a set of d(i) bonds (i.e. its degree), to be eventually connected with other elements, which are denoted by a vector* $\beta_i = (\beta_{i,1}, \ldots, \beta_{i,d(i)})$.
- We think of $\beta_{i,k}$ as an address variable or pointer.
- α_i is a vector of attributes for
  - (a) a geometric transformation, e.g. the central position, scale, orientation and plastic deformation
  - (b) appearance, such as intensity contrast, profile or surface albedo.

Image Primitives

- Image primitives ≈ tokens.
- An image primitive is a small image patch with a degree d connections or bonds which are illustrated by the half circles.
- Each primitive has a number of attributes for its geometry and appearance.
  - Geometric attributes: position, orientation, scale, relative positions of the bonds with respect to the centre.
  - The appearance is described by the intensity profiles around the centre and along the directions perpendicular to the line-segment connecting the centre and the bonds.

d	Name of primitive
0	Blobs
1	Terminators
2	Edges, ridges or L-junctions
3	T-junctions
4	Cross-junctions

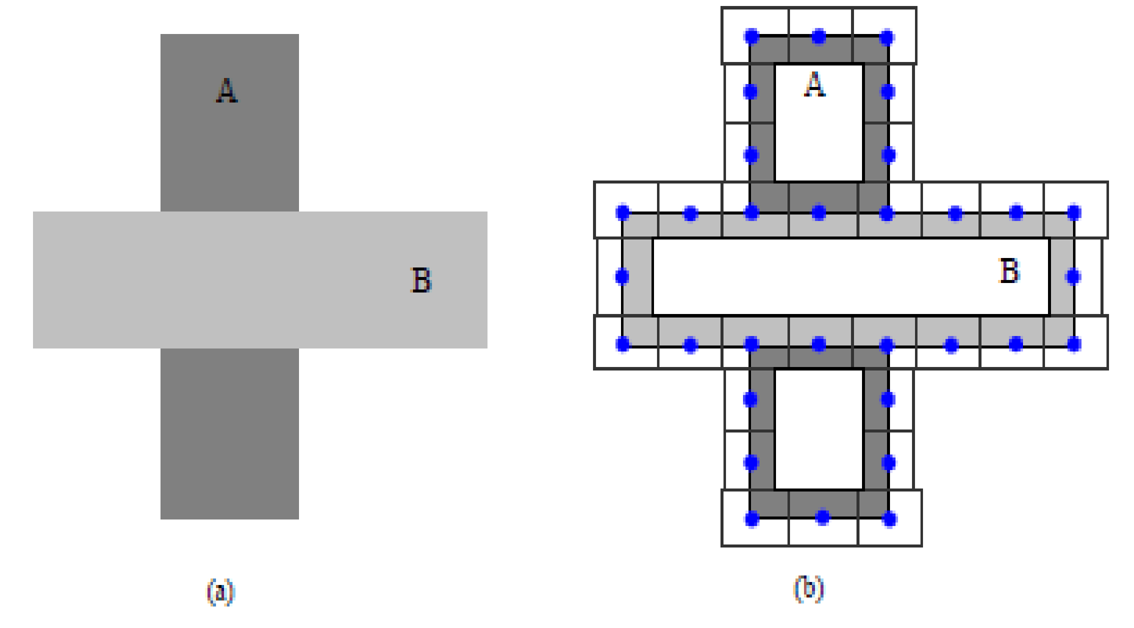


Low level visual vocabulary (image primitives)

(a) Some examples of image primitives: blobs, terminators, edges, ridges, `L'-junctions, `T'-junction, and cross junction etc. These primitives are the elements for composing a bigger graph structure at the upper level of the hierarchy.

(b) An example of composing a big `T'-shape image using 7 primitives.

- In the adjoining figure, the boundaries of the two rectangles are covered by 4 `T'-junctions, 8 'L'-junctions, and 20 step edges. We denote the domain covered by an image primitive φ_{ski} by Λ_{ski}, and the pixels covered by these primitives, which are called the "sketchable part".
- Sketchable part ($\mathbf{I}_{sk}$) is modelled by the image primitives using their intensity profiles.
- The remaining pixels are flat or stochastic texture areas, called non-sketchable, and are clustered into a few homogeneous texture areas. They can be reconstructed through Markov random field models conditional on $\mathbf{I}_{sk}$

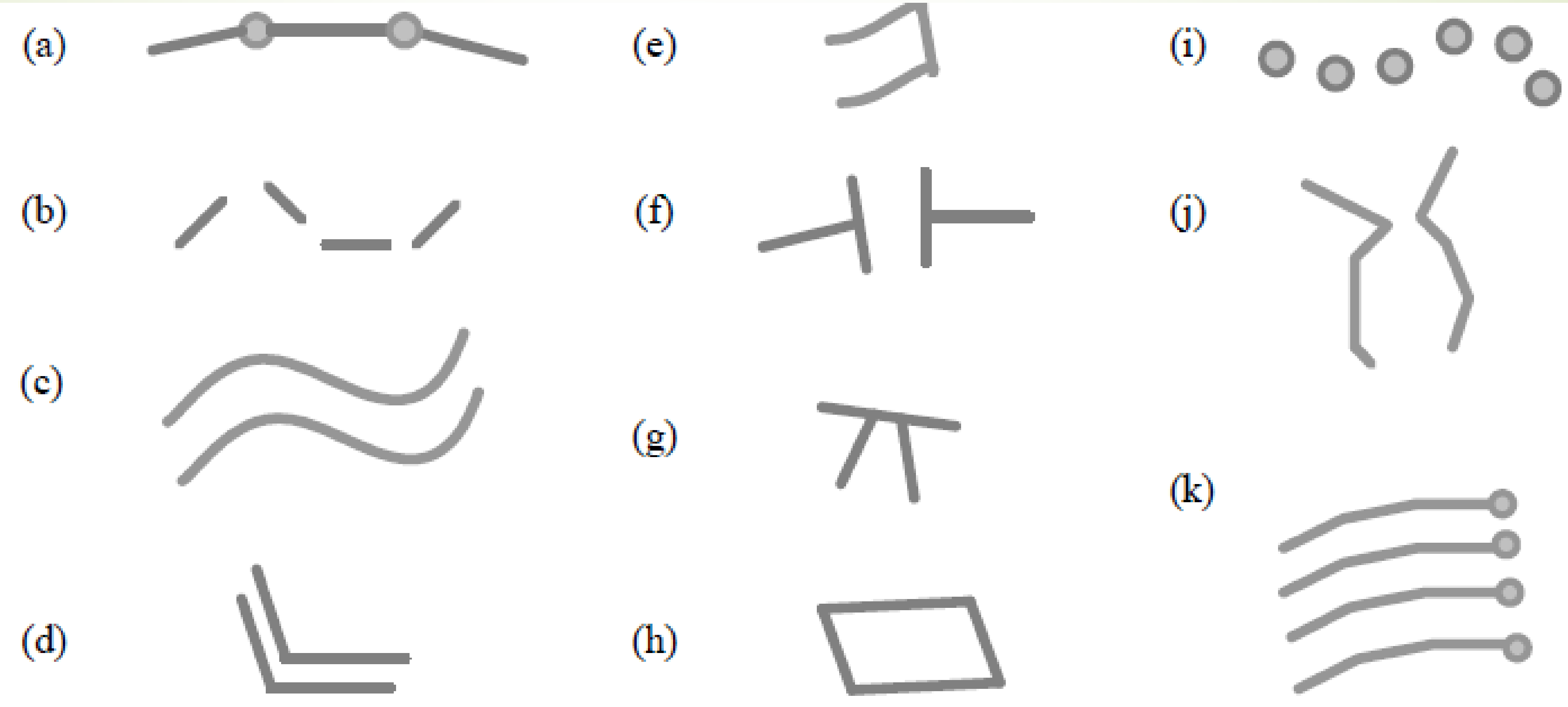


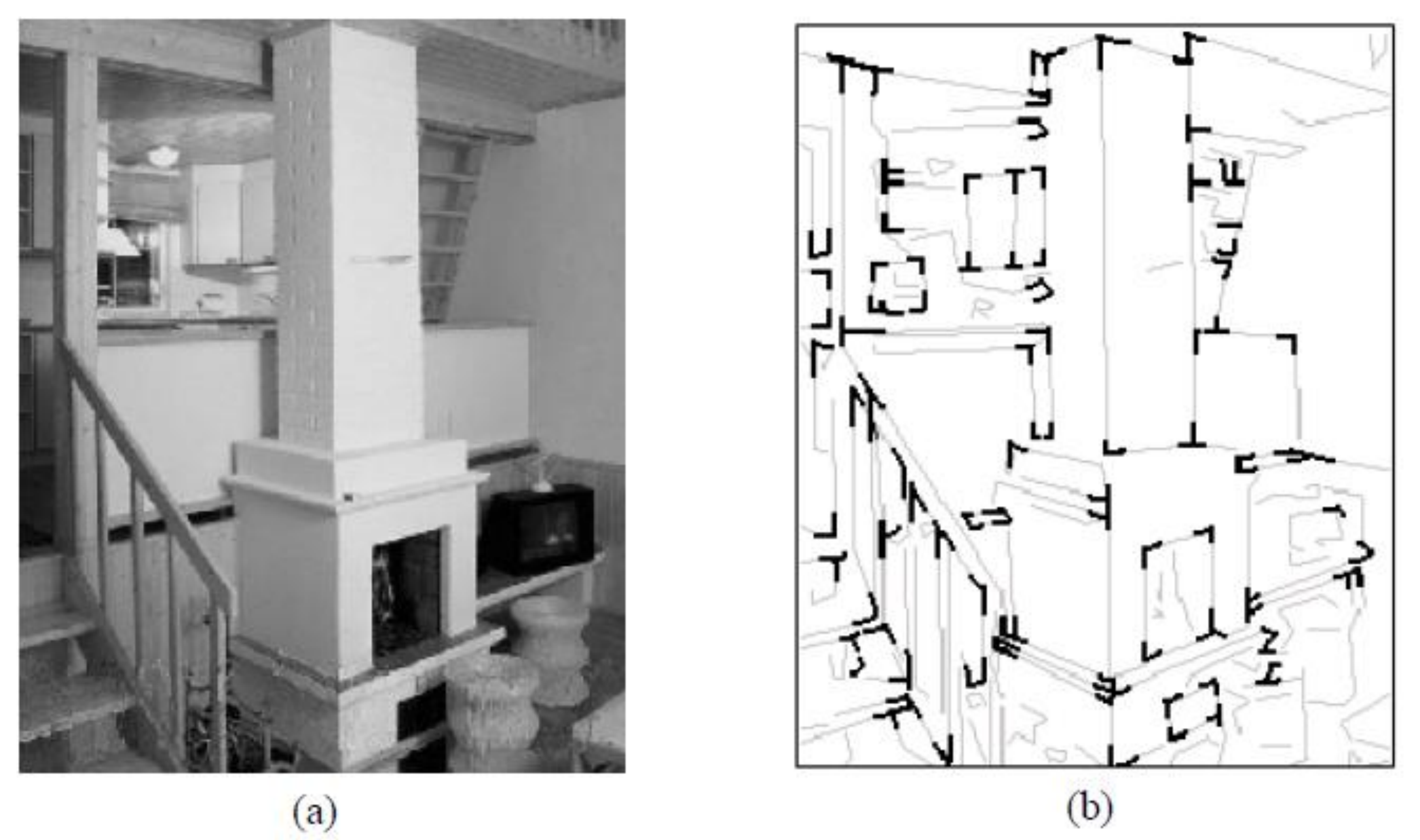
Composing primitives into a graph configuration. (a) is a simple image, and (b) is a number of primitives represented by rectangles which cover the structured parts of the image. The remaining part of the image can be reconstructed through simple heat diffusion

Basic Geometric Groupings

- Alignment, parallelism and symmetry, especially as created by occlusions, are the driving forces behind the grouping of lower level parts into larger parts. These groupings occur at every scale.
- Symmetry: larger scale feature
  - Occurs very often in nature - highly detectable by people even in cluttered scenes.
  - Parallel lines also occur frequently in nature, e.g. in tree trunks.
- Occlusion is especially important:
  - common
  - Strongest clue in a static 2D image to the 3D structure of the scene.
  - Implies the existence of an `amodal' or occluded contour representing the continuation of the left and right edges behind the central bar.
- Graphlets learned through clustering and binding the image primitives. They are generic 2D patterns, and some of them could be interpreted as object parts.

Middle level visual vocabulary: common groupings found in images. (a) extended curves, (b) curves with breaks and imperfect alignment, (c) parallel curves, (d) parallels continuing past corners, (e) ends of bars formed by parallels and corners, (f) curves continuing across paired T-junctions (the most frequent indication of occlusion), (g) a bar occluded by some edge, (h) a square, (i) a curve created by repetition of discrete similar elements, (j) symmetric curves and (k) parallel lines ending at terminators forming a curve

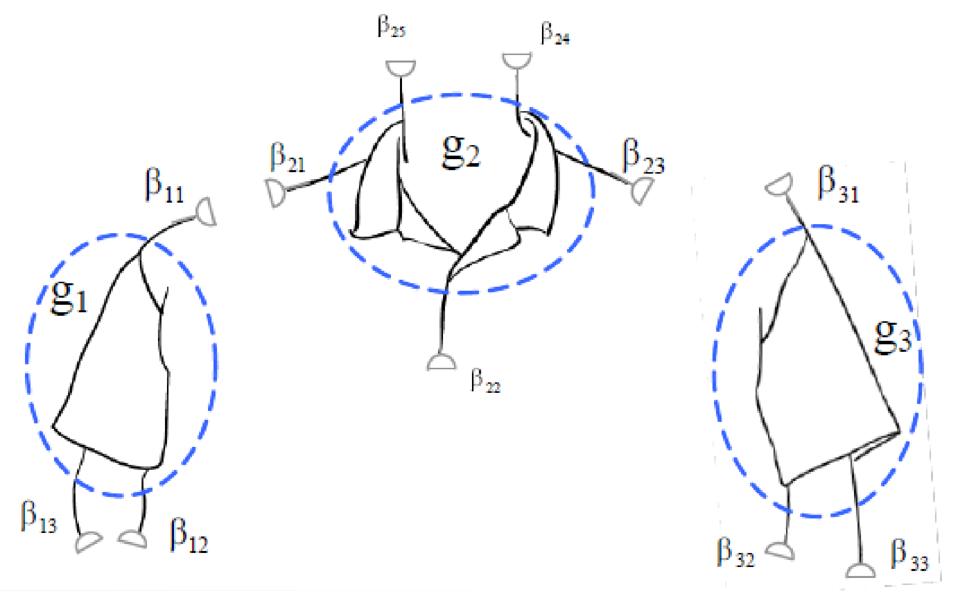




An example of graphlets in natural image. The graphlets are highlighted in the primal sketch. These graphlets can be viewed as larger pieces of lego.

Parts and Objects

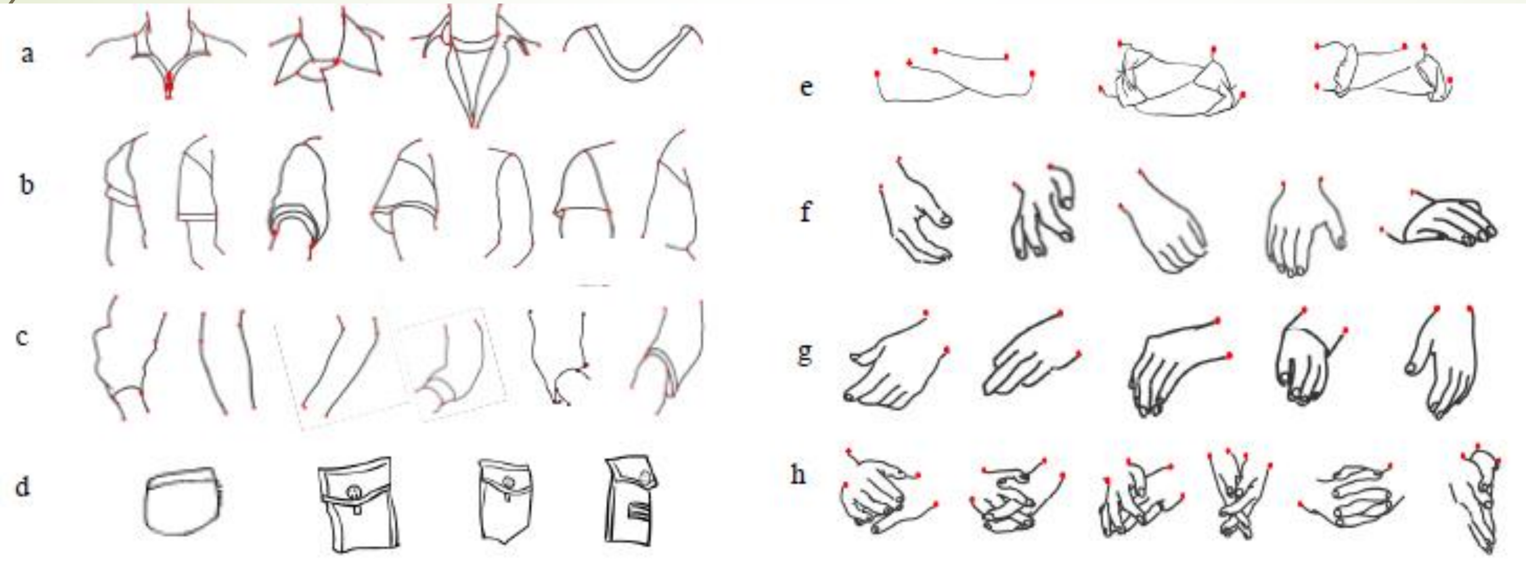
- Background objects -> dictionary will be object parts - object parts significant in that category, reusable by a few categories but low overall frequency .
  - Less entropically significant than the image primitives at the low level.
  - Open bonds for connecting to other objects → represented as address variables that point to other bonds.
- Bigger and more structured than image primitives; forms continuous spectrum for vision vocabulary from low to high level.
- Analogy to OOP.



Example: Object category ‘clothes’

A shirt has 3 parts: a collar and left and right sleeves. Each part is represented by an attribute graph with open bonds, like graphlets. For example, the collar part has 5 bonds, and the two sleeves have 3 bonds to be connected with the arms and collar. By decomposing a number of instances in the clothes category together with upper body and shoes, one can obtain a dictionary of parts.

High level visual vocabulary - the objects and parts. We show an example of upper body clothes made of three parts: a collar, a left and a right short sleeves. Each part is again represented by a graph with bonds



The dictionary of object parts for cloth and body components. Each element is a small graph composed of primitives and graphlets and has open-bonds for connecting with other parts