Noname manuscript No.
(will be inserted by the editor)

A Numerical Study of the Bottom-up and Top-down Inference Processes in And-Or Graphs

Tianfu Wu · Song-Chun Zhu

Received: date / Accepted: date

Abstract This paper presents *a numerical study* of the bottom-up and top-down inference processes in hierarchical models using the And-Or graph as an example. Three inference processes are identified for each node A in a recursively defined And-Or graph: the $\alpha(A)$ process detects node A directly based on image features, the $\beta(A)$ process computes node A by binding its child node(s) bottom-up and the $\gamma(A)$ process predicts node A top-down from its parent node(s). All the three processes contribute to computing node A from images in complementary ways. The objective of our numerical study is to explore how much information each process contributes and how these processes should be integrated to improve performance. We study them in the task of object parsing using And-Or graph formulated under the Bayesian framework. Firstly, we isolate and train the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes separately by blocking the other two processes. Then, information contributions of each process are evaluated individually based on their discriminative power, compared with their respective human performance. Secondly, we integrate the three processes explicitly for robust inference to improve performance and propose a greedy pursuit algorithm for object parsing. In experiments, we choose two hierarchical case studies: one is junctions and rectangles in low-to-middle-level vision and the other is human faces in high-level vision. We observe that (i) the effectiveness of the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes depends on the scale and occlusion conditions, (ii) the α(face) process is stronger than the α processes of facial components, while β(junctions) and β(rectangle) work much better than their α processes, and (iii) the integration of the three processes improves performance in ROC comparisons.

Keywords Bottom-up/Top-down · Information Contribution · Hierarchical Model · And-Or Graph · Object Parsing

T.F. Wu[†,*] and S.-C. Zhu[†,‡,*]
Department of [†]Statistics and [‡]Computer Science, University of California, Los Angeles, USA
[*]Lotus Hill Research Institute (LHI), Ezhou, China
E-mail: {tfwu, sczhu}@stat.ucla.edu

1 Introduction

1.1 Motivations and objectives

In the literature of object detection, recognition and parsing, hierarchical models and contextual information are widely used and shown to improve performance (Geman et al, 2002; Riesenhuber and Poggio, 1999; Ullman et al, 2002; Schneiderman and Kanade, 2002; Todorovic and Ahuja, 2008b; Wu et al, 2009; Sudderth et al, 2008; Felzenszwalb et al, 2009; Fidler et al, 2008; Torralba, 2003; Divvala et al, 2009). In hierarchical models, we observe that certain nodes, such as the human face, are often interpreted in a top-down fashion. One does that because it is much more effective to detect the full human face than individual facial components. In contrast, some other nodes such as junctions and handwriting digits are more effectively computed through bottom-up binding. For example, it is very difficult to detect rectangles directly. Instead, we can detect parallel lines or L-junctions first and then bind those compatible parallel lines or compatible L-junctions under some constraints. Furthermore, if we take scale and occlusion into account, one may have to adapt different computing strategies for different object instances. Fig.1 shows three cases in detecting human faces: the first case is a normal situation in which human faces are at middle

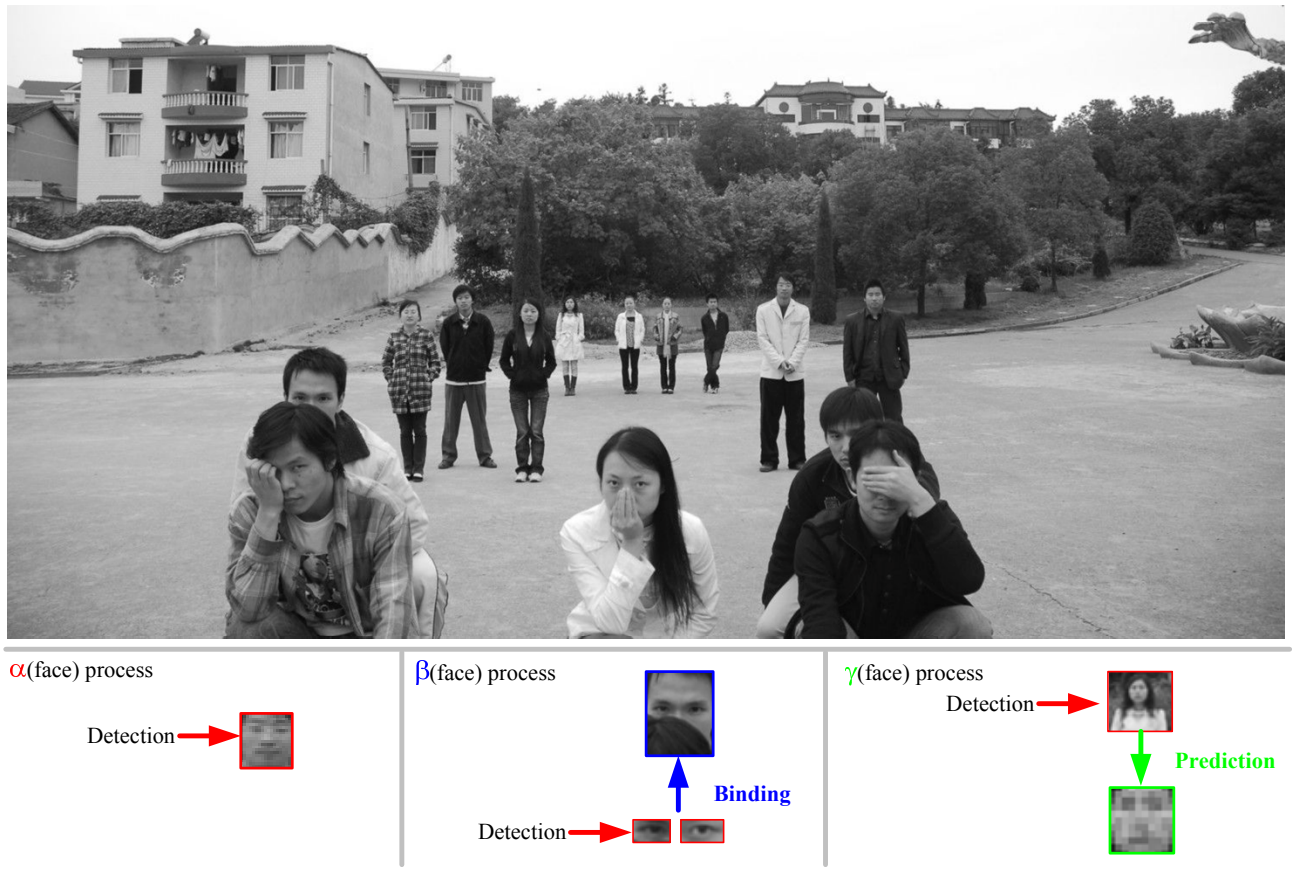

Fig. 1 Motivation of the α, β and γ inference processes using human face detection as an example. There are three cases of human faces appearing in the top image, each of which entails a different inference process, termed the α(face), β(face) and γ(face) processes respectively, as illustrated in the bottom panel. General identifications of the three processes are illustrated in Fig.2 and formal definitions of the three processes are introduced in Sec.1.2.

resolution without occlusion, the second contains human faces at higher resolution but with occlusion and the third contains human faces at extremely low resolution. Intuitively, these three cases entail three different inference processes as illustrated in the bottom of Fig.1: human faces in the first case can be detected directly based on image data features, but those features work for the first case would fail in the second and the third cases due to occlusion and low resolution respectively. Human faces in the second case can be computed by binding those detectable facial components such as eyes and mouth, etc., and those in the third case can be predicted from their detectable surrounding contexts such as the head-shoulders. It is natural to ask the following three questions.

(i) What inference processes, bottom-up and top-down, can be identified for nodes in hierarchical models?
(ii) How much information does each of them contribute for different nodes?
(iii) How should they be integrated to improve detection performance?

In this paper, we present a framework to study these three questions in the task of object parsing. We formulate object parsing under the Bayesian inference framework. We choose the And-Or graph (AoG) (Zhu and Mumford, 2006) as our hierarchical model to represent object grammar. The AoG is a recursive structure. First of all, we identify three inference processes for each node A in an AoG, termed the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes. The three processes account for the three cases as the human face example has shown in Fig.1. Then, through scaling and masking image patches of node A, we isolate and train the three processes separately by blocking the other two processes and evaluate their information contributions individually by both computers and humans based on their discriminative power. Secondly, we integrate the three processes explicitly for robust inference to improve performance and propose a greedy pursuit algorithm for object parsing. We choose two hierarchical case studies in our object parsing experiments, one is junctions and rectangles in low-to-middle-level vision and the other is human faces in high-level vision. We observe that (i) the effectiveness

of the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes depends on the scale and occlusion conditions, (ii) the α(face) process is stronger than the α processes of facial components, while β(junctions) and β(rectangle) work much better than their α processes, and (iii) the integration of the three processes improves performance in ROC comparisons. In our on-going work, we are studying how these numerical observations lead to improved computational efficiency through scheduling.

In the following, we shall briefly introduce the AoG and the α, β and γ inference processes, and then give an overview of the numerical study.

1.2 Overview of the AoG and the α, β and γ processes

The AoG representation is a hierarchical model recursively defined for effective visual knowledge representation which embodies a stochastic context sensitive image grammar (SCSG) (Zhu and Mumford, 2006). The SCSG combines the reconfigurability of stochastic context free grammar (SCFG) with the contextual constraints of graphical Markov random field (MRF) models. Generally, an AoG can represent the structural, geometric, appearance, and probabilistic information for an object category. There are three types of nodes in an AoG (see Fig.4): *And-nodes* represent decomposition and are denoted by solid circles, *Or-nodes* represent alternative structures and are denoted by dash circles and *terminal nodes* link to image data and are denoted by solid rectangles. Each And-node in the AoG can also directly terminate to image data (through a terminate node) when it is at low resolution. Traditional hierarchical models do not have Or-nodes and allow only leaf nodes to link to image data (Riesenhuber and Poggio, 1999; Aycinena et al, 2008). We will introduce the definition of the AoG in Sec.2.1.

The α, β and γ processes in AoG. Fig.2 shows a portion of an AoG using the face example discussed in Fig.1 where node A represents human face, node P represents head-shoulder and node C_i's represent facial components ($i = 1, 2, 3$). As an AoG is recursively defined, we can consider the α, β and γ processes of And-node A in Fig.2 without loss of generality.

Definition 1: (the α process). The $\alpha(A)$ process handles situations in which node A is at middle resolution without occlusion. Node A can be **detected directly** (based on its compact image data) and **alone** (without taking advantage of surrounding context) while its children or parts are not recognizable alone in cropped patches. An example of α(face) process is shown in the left-bottom panel of Fig.1. Most of the sliding window detection methods in computer vision literature belong to this process. It can be viewed as either bottom-up or top-down. By bottom-up, it means that discriminative models are used to train the α process, such as the Adaboost classifiers (Viola and Jones, 2004). By top-down, it means that generative models are used, such as the active basis model (Wu et al, 2009).

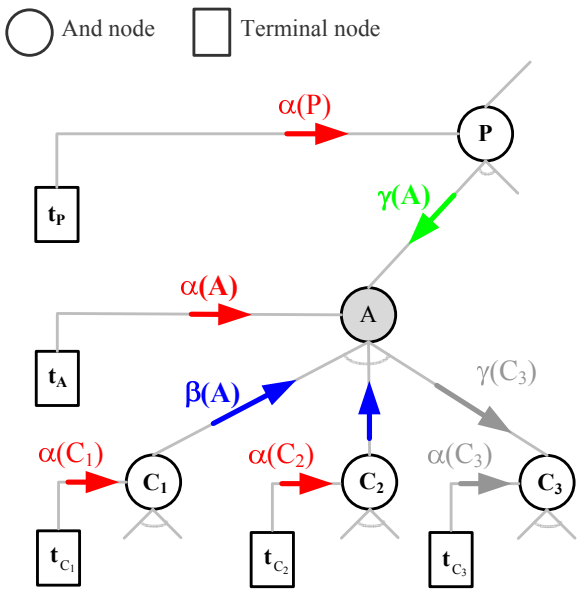

Fig. 2 Illustration of identifying the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ inference processes for each And-node A in an AoG (see texts in Sec.1.2 for detail definitions). The $\alpha(A)$ process is directly based on the compact image data of node A (either bottom-up or top-down), the $\beta(A)$ process generates hypotheses of node A by bottom-up binding the α processes of some child node(s) (for example, $(\{\alpha(C_1), \alpha(C_2)\} \rightarrow \beta(A))$, and the $\gamma(A)$ process predicts hypotheses of node A from the α processes of some parent node(s) (for example, $\alpha(P) \rightarrow \gamma(A)$ or $\beta(A) \rightarrow \gamma(C_3)$ in a top-down fashion). In computing, each process has two states: "on" or "off", for example, $\alpha(C_3)$ process is off and we show it in grey. As an AoG is defined recursively, each And-node has its own α, β and γ processes (except that the root node's γ processes and the β-processes of leaf nodes are always off).

Definition 2: (the β process). When node A is at high resolution, it is more likely to be occluded in a scene. Node A itself is not detectable in terms of the $\alpha(A)$ process due to occlusion. A subset of node A's child nodes can be detected in cropped patches (say, their α processes are activated). Then, the $\beta(A)$ process computes node A by **binding** the detected child nodes **bottom-up** under some compatibility constraints. An example of β(face) process is illustrated in the middle-bottom panel of Fig.1. Most of component (Biederman, 1987; Heisele et al, 2007), fragment (Ullman et al, 2002) or part (Amit and Trouvé, 2007; Schneiderman and Kanade, 2002) based methods, the constellation models (Fei-Fei et al, 2006; Fergus et al, 2007) and the pictorial models (Felzenszwalb and Huttenlocher, 2005) belong to this process.

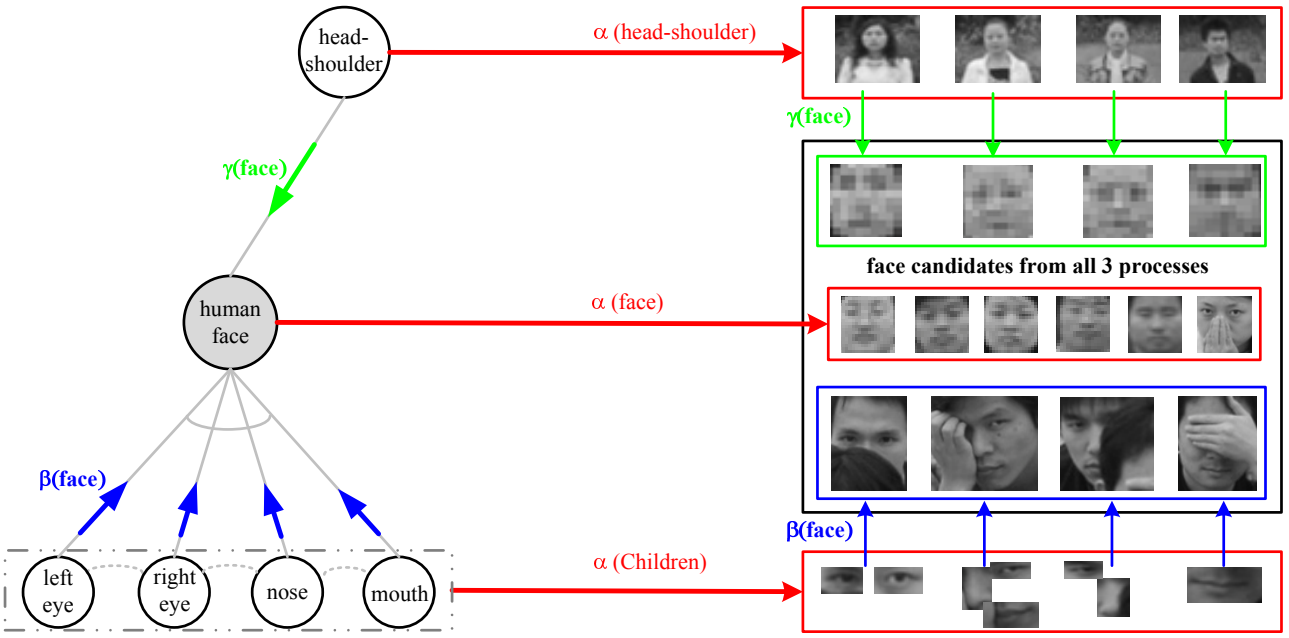

Fig. 3 Illustration of integrating the α(face), β(face) and γ(face) in the human face AoG for face detection. The three inference processes are effective in complementary ways relatively depending on the scale and occlusion conditions. The typical situations shown here are common to other object categories.

Definition 3: (the γ process). The $\gamma(A)$ process handles situations in which node A is at very low resolution. Node A can not be detected alone in isolation based on $\alpha(A)$, and neither can its parts. Then, the $\beta(A)$ process also fails. An example of γ(face) process is illustrated in the right-bottom panel of Fig.1. So, information outside of the local window must be incorporated. The $\gamma(A)$ process **predicts** node A **top-down** from a parent node whose α process is activated. In this paper, we let the parent node pass contextual information, such as information from some sibling nodes or other spatial context. Most of the context-based methods (Torralba, 2003; Hoiem et al, 2008; Fink and Perona, 2003) belong to this process.

For node A, all the three inference processes, $\alpha(A)$, $\beta(A)$ and $\gamma(A)$, contribute to computing it from images in complementary ways. The effectiveness of each process depends on the scale and occlusion conditions. As shown in Fig.3, the three cases of human faces shown in Fig.1 can be handled by the α(face), β(face) and γ(face) respectively. Intuitively, for robust inference we should integrate them. As an AoG is a recursive structure, the three inference processes are also defined recursively and each And-node has its own α, β and γ inference processes (except that the γ process of the root node and the β processes of leaf nodes are always disabled).

Motivation for training the α, β and γ processes separately. In this paper, we train the three processes separately based on their respective isolated training data. We introduce the isolation method in Sec.4.1. Here, we propose the motivation and necessity to do that. Suppose we want to learn a human face classifier (ie. the α(face) process). There are two choices in selecting positive examples: (i) only face examples like those pointed by the α(face) arrow in Fig.3, or (ii) a set of human face examples mixing all those shown in the right middle box in Fig.3. In the literature, people often get positive examples by cropping image patches only based on the labelled bounding boxes. When labelling the bounding box for an object instance, however, one often already takes advantage of all the information coming from the α, β and γ processes. Often, most of existing work often trained a classifier based on a set of mixed positive examples (especially, mixing the α case and the γ case). Then, the learned classifier could be contaminated depending on the mixing rate implicitly (Torralba and Murphy, 2007; Fink and Perona, 2003; Avidan, 2006). We can explain the contamination. Generally, whether a feature is selected into a classifier depends on how different the feature responses of positive examples and those of negative examples are. The feature responses of a set of mixing positive examples do not reflect the true discriminative power of the feature, however.

1.3 Overview of the numerical study

Our numerical study consists of the following four steps.

I. Isolating the processes. To measure their individual information contributions, we isolate the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes by *scaling* and *masking* the image patch of node A based on the labeling information to match their definitions stated in Sec.1.2. Fig.7 illustrates the isolation procedure in Sec.4.1. The labeling information used in this paper is the manually labeled parse graphs (say, instances of the AoG representation) which are available in the LHI image database (Yao et al, 2007). Based on the isolation, we generate the training and evaluation dataset for each process in Sec.4.2.

II. Learning. The three processes are trained separately and the learning procedure is based on the MLE framework. In this paper, we choose the recently proposed active basis model (Wu et al, 2009) as the $\alpha(A)$ process in Sec.4.3. For the $\beta(A)$ and $\gamma(A)$ processes, both of them include two components, one is the α processes of node A's child nodes (for $\beta(A)$) or parent node(s) (for $\gamma(A)$) and the other is the relation model which constrains the configuration of all the nodes appeared in the $\beta(A)$ or $\gamma(A)$ process. We consider three types of relations in the configuration, relative locations, scales and orientations respectively. They are parameterized as Gaussian distributions and learned from the training dataset in Sec.4.4 and Sec.4.5.

III. Evaluation of the information contribution. We evaluate the individual information contribution of each process based on their discriminative power and we also study the human performance of the three processes individually in Sec.5. The evaluation procedure is similar to the decision tree framework (Breiman et al, 1984). The information contribution of each process is defined as the impurity reduction obtained by applying it in the evaluation dataset. In the human study, we use the psychology toolbox (Brainard, 1997) to set up our experimental environments. In order to reduce the amount of data to be observed by human subjects, we use the false positives in the computer experiments as the negative samples in the human study. We control the observing time as a additional isolation method for humans.

IV. Integration for improving performance. As illustrated in Fig.1 and Fig.3, we know that all the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes contribute to detecting node A in complementary ways which depend on the scale and occlusion conditions. Given an input image, the objective of object parsing is to output the parse graph of each object instance of node A (for example the human face) on the fly and we often do not know the specific situation of node A in advance. We formulate object parsing using AoG under the Bayesian framework in Sec.2. For robust inference, we integrate the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes explicitly and propose a greedy pursuit algorithm for object parsing under the Bayesian framework in Sec.3. The experimental results show performance improvement from the integration.

1.4 Related work and our main contributions

In hierarchical models, bottom-up and top-down are two basic computing mechanisms and are often used with three strategies:

(i) pure bottom-up inference which passes messages in a feed-forward manner in the hierarchy, starting from data-driven features (Riesenhuber and Poggio, 1999; Serre et al, 2007; Aycinena et al, 2008).
(ii) pure top-down inference which passes messages in a feed-back manner in the hierarchy, starting from template matching (Todorovic and Ahuja, 2008a; Demirci et al, 2009).
(iii) one pass of bottom-up inference followed by one phase of top-down inference (Tu et al, 2005; Epshtein et al, 2008; Borenstein and Ullman, 2008; Levin and Weiss, 2009; Demirci et al, 2006).

In the recent vision literature, it is well acknowledged that both bottom-up and top-down inference processes contribute to object detection, recognition and parsing, and they should be combined (Lee and Mumford, 2003; Jin and Geman, 2006). Despite many efforts, it has been unclear how to combine bottom-up and top-down inference processes in a robust and effective way. The first numerical evaluation of top-down versus bottom-up is the ROC comparisons addressed in (Han and Zhu, 2009). Our previous work on compositional boosting (Wu et al, 2007) proposed to separate the implicit testing (ie, the α process) and explicit testing (ie, the β process) and then combine them under the compositional boosting. This paper presents a more general framework and formulation to integrate the bottom-up and top-down inference processes (say, the α, β and γ processes) in an *explicit* way so that we can compare different kinds of integrations numerically, benefitting from the isolation and separate training procedures.

Our contributions. In comparison to previous work, this paper has the following novel aspects:

(i) It presents a numerical study of the bottom-up and top-down inference processes in hierarchical models using the AoG as an example. To the best of our knowledge, it is the first time this is done in the vision literature.

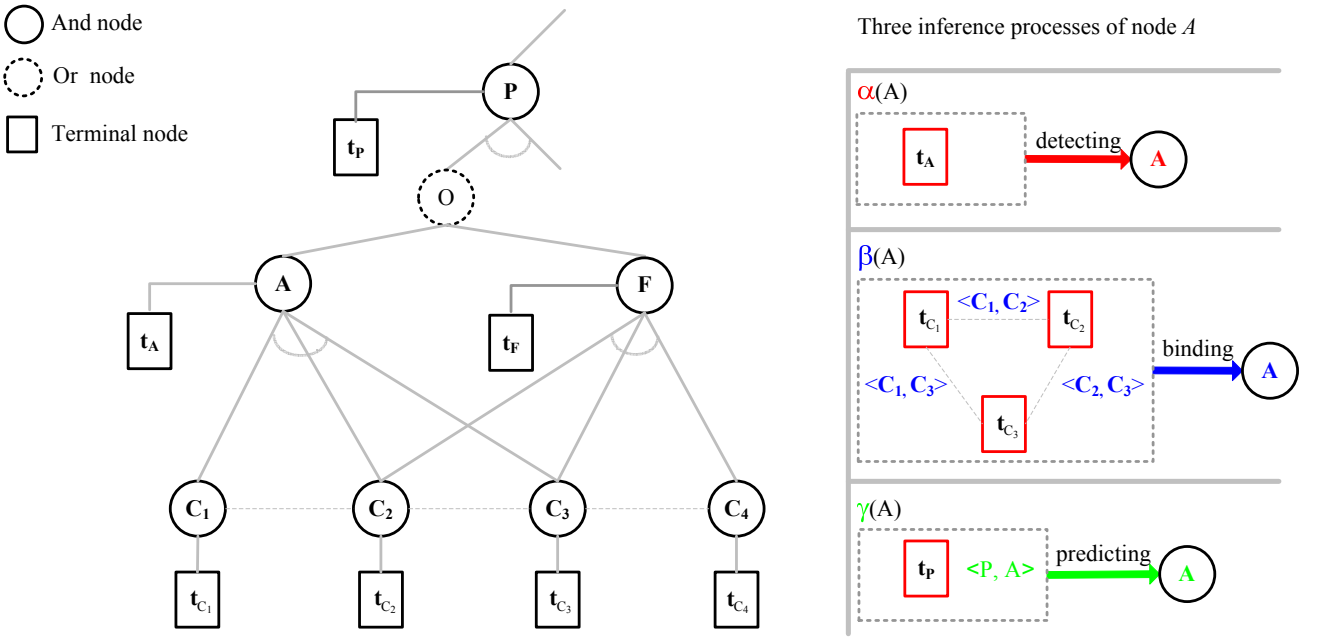

Fig. 4 Illustration of the And-Or graph (AoG) representation. There are three types of nodes: And-nodes for decompositions, Or-nodes for alternative structures and terminal nodes for image data link. In an AoG, all And-nodes can directly terminate to image data through a terminate node when it is at low resolution. In traditional hierarchical models, only leaf nodes can link to image data. The α, β and γ processes are specified for each And-node A in an AoG as illustrated in the right panel.

(ii) It trains the identified α, β and γ processes separately to reduce contamination by using an isolation procedure. It evaluates information contributions of the identified α, β and γ processes individually in both computer and human experiments.

(iii) It proposes a pursuit algorithm for object parsing using AoG which integrates the α, β and γ processes explicitly under the Bayesian framework for robust inference. The algorithm presents a way to link discriminative learning to the Bayes.

(iv) It observes that the effectiveness of the α, β and γ processes depends on the scale and occlusion conditions. The α(face) process is stronger than the α processes of facial components, while β(junctions) and β(rectangle) work much better than their α processes.

(v) Potentially, these numerical studies could shed some lights on how to schedule the α, β and γ processes of different nodes in an AoG, especially when we have a big AoG with hundreds of nodes.

1.5 Paper organization

The remainder of this paper is organized as follows. In Sec.2, we define the AoG representation and formulate object parsing using AoG under the Bayesian framework. In Sec.3, we propose a greedy pursuit algorithm for object parsing and connect the identified α, β and γ processes with the Bayesian inference explicitly. In Sec.4, we present the isolation method and separate learning procedures for the three processes. In Sec.5, we propose the method of evaluating the information contribution of each process. In Sec.6, we show two series of experiments, one is for the information contribution evaluation of the three process and the other is for the object parsing experiments. Finally, in Sec.7, we summarize the paper and discuss some on-going work of the scheduling problem.

2 Problem formulation

In this section, we introduce the AoG for object representation (Zhu and Mumford, 2006) and formulate object parsing using AoG under the Bayesian framework. Then, we derive the α, β and γ processes in the Bayesian formula.

2.1 The AoG representation

Fig.4 shows a fragment of an AoG. An AoG embodies a stochastic context sensitive image grammar and is specified by a quadruple,

$$\mathcal{G} = (V_N, V_T, E, \mathcal{P}) \quad (1)$$

$V_N = V_{and} \cup V_{or}$ is a set of nonterminal nodes with an And-node set V_{and} representing decompositions (shown

Symbol	Interpretation
Λ	image lattice
I_Λ	an image I defined on the lattice Λ
$\mathcal{G} = (V_N, V_T, E, \mathcal{P})$	a 4-tuple representing SCSG $\mathcal{G}$ embedded in the AoG
$V_N = V_{and} \cup V_{or}$	a set of nonterminal nodes including a And-node set V_{and} and a Or-node set V_{or}, a nonterminal node is represented by a capital letter, such as $O \in V_{or}, P, A, C_1 \in V_{and}$.
V_T	a set of terminal nodes having one-to-one correspondence with And-nodes in V_{and}, a terminal node is represented by t with a subscript of its corresponding And-node, such as t_P, t_A, t_{C_1}
$E = E_{or} \cup E_{dec} \cup E_t \cup E_{rel}$	a set of edges including four types, a switching edge set E_{or}, a decomposition edge set E_{dec}, a terminal edge set E_t and a relation edge set E_{rel}
$V_{and}^{ch}, V_{and}^{prt} \subset V_{and}$	subsets of And-nodes which have child node(s) and parent node(s) respectively
$ch(), prt()$	children node set and parent node set of a And-node such as $ch(A), prt(A) \subset V_{and}$
$X(), x()$	a vector of attributes of a And-node such as $X(A)$ or a terminal node such as $x(t_A)$ respectively, often including the relative location, scale and orientation information
$pg = (V_N^{pg}, V_T^{pg}, E^{pg}, p(pg))$	a parse graph which is an instance of the AoG or a valid configuration of the grammar $\mathcal{G}$
$\mathcal{C}(pg) = \{t, x(t) : t \in V_T^{pg}\}$	a configuration which is collapsed from a parse graph to an image lattice
$\alpha(A; \theta)$	α inference process of an And-node $A \in V_{and}$, also use $\alpha(A)$ for simplicity
$\beta(A\|\mathbf{c}; \phi)$	β inference process of an And-node $A \in V_{and}$ given a (complete or partial) child node set $\mathbf{c} \subseteq ch(A)$
$\gamma(A\|P; \varphi)$	γ inference process of an And-node $A \in V_{and}$ given a parent node $P \in prt(A)$
Tst()	a general notation for the α, β or γ process as a test function during evaluation
IC(Tst)	information contribution of Tst() (say, α, β or γ process)
$D_\alpha^+(A)$, $D_\alpha^-(A)$	positive and negative dataset for α process of node A
$D_\beta^+(A\|\mathbf{c})$, $D_\beta^-(A\|\mathbf{c})$	positive and negative dataset for β process of node A given a child node set $\mathbf{c}$
$D_\gamma^+(A\|P)$, $D_\gamma^-(A\|P)$	positive and negative dataset for γ process of node A given a parent node P
w_A^α, $w_A^{\beta(\mathbf{c})}$, $w_A^{\gamma(P)}$	the α, β and γ weights
$w_{\mathbf{c}}^{\text{bind}}, w_P^{\text{predict}}$	the compatibility weight in binding in $w_A^{\beta(\mathbf{c})}$ and the compatibility weight in prediction in $w_A^{\gamma(P)}$
$w_A = w_A^\alpha + w_A^{\beta(\mathbf{c})} + w_A^{\gamma(P)}$	the pursuit index in our algorithm

Table 1 The table of main notations used in this paper

by solid circles in Fig.4) and an Or-node set V_{or} representing alternative structures (shown by dash circles in Fig.4). A nonterminal node is denoted by capital letters, for example, $P, A, C_1 \in V_{and}$, $O \in V_{or}$.

V_T is a set of terminal node (shown by rectangles in Fig.4). In an AoG, each And-node can directly terminate to image data through a terminal node when it is at low resolution. In traditional hierarchical models, however, only leaf nodes can link to image data. A terminal node is denoted by lowercase t with the subscript letter of the corresponding nonterminal node, for example, $t_A, t_{C_1} \in V_T$.

$E = E_{or} \cup E_{dec} \cup E_t \cup E_{rel}$ is a set of edges including four types:

(i) $E_{or} = \{< O, A >: O \in V_{or}, A \in V_{and}\}$ is a set of vertical switching edges which link Or-nodes to corresponding And-nodes as alternatives.

(ii) $E_{dec} = \{< A, C >: A \in V_{and}, C \in ch(A), ch(A) \neq \emptyset\}$ is a set of vertical decomposition edges which connect And-nodes to their child And-nodes. $ch(A) \subset V_{and}$ denotes the set of child nodes of node A.

(iii) $E_t = \{< A, t_A >: A \in V_{and}, t_A \in V_T\}$ is a set of vertical terminating edges which connect And-nodes to their corresponding terminal nodes.

(iv) $E_{rel} = \{< A, F >: A, F \in V_{and}, prt(A) \cap prt(F) \neq \emptyset\}$ is a set of horizontal relation edges which connect among And-nodes at the same layer, often pairwise. $prt(A)$, $prt(F) \subset V_{and}$ denotes the sets of parent nodes of nodes A and B respectively.

$\mathcal{P}$ is the probability defined over the space of all valid parse graphs which are defined below.

In an AoG $\mathcal{G}$, each Or-node $O \in V_{or}$ has a switching variable indicating the occurring frequency of its branches, denoted by $p(A|O)$ $(A \in V_N)$. Both And-nodes $A \in V_{and}$ and terminal nodes $t \in V_T$ have a vector of attributes denoted by $X(A)$ and $x(t)$ respectively. For a subset $\mathbf{v} \subset V_{and}$, we denote $X(\mathbf{v})$ as the concatenation of attributes for And-nodes in the subset $\mathbf{v}$. The attributes include location, scale, orientation, etc. As illustrated in the right panel of Fig.4, the attributes of an And-node can be passed from attributes of other nodes in three ways: (i) the corresponding terminal node directly, (ii) child And-node(s) during binding process or (iii) parent node(s) during prediction process.

Parse graph. A parse graph, pg, is one instance of the AoG by selecting variables at the Or-nodes and specifying the attributes for And-nodes through the three

ways stated above. We have,

$$pg = (V_N^{pg}, V_T^{pg}, E^{pg}, p(pg)) \tag{2}$$

where $V_N^{pg} = V_{and}^{pg} \cup V_{or}^{pg}$ ($V_N^{pg} \subset V_N$) is the nonterminal node set of the parse graph, $V_T^{pg} \subset V_T$ is the terminal node set and $E^{pg} = E_{or}^{pg} \cup E_{dec}^{pg} \cup E_t^{pg} \cup E_{rel}^{pg}$ ($E^{pg} \subset E$) is the edge set. We have a parse tree if we omit the horizontal relation edges $E_{rel}^{pg} \subset E^{pg}$ in a parse graph.

$p(pg)$ is the prior probability of parse graph pg, measuring the occurring probability of each switching edge $< O, A > \in E_{or}^{pg}$ and the compatibility probabilities among the attributes of And-nodes (pairwise used in this paper) in V_{and}^{pg} with vertical decomposition edge $< P, A > \in E_{dec}^{pg}$ and horizontal relation edge $< C_i, C_j > \in E_{rel}^{pg}$. So, we have,

$$p(pg) = \frac{1}{Z} \exp\{-\mathcal{E}(pg)\} \tag{3}$$

where $Z = \sum_{pg} \exp\{-\mathcal{E}(pg)\}$ is the partition function and $\mathcal{E}(pg)$ is the total energy,

$$\begin{aligned} \mathcal{E}(pg) = & - \sum_{<O,A> \in E_{or}^{pg}} \log p(A|O) \\ & - \sum_{<P,A> \in E_{dec}^{pg}} \log p(X(A)|X(P)) \\ & - \sum_{<C_i,C_j> \in E_{rel}^{pg}} \log p(X(C_i), X(C_j)) \end{aligned} \tag{4}$$

and $p(A|O)$ is the switching probability estimated by the occurring frequency in training data (Zhu and Mumford, 2006), $p(X(A)|X(P))$ captures the top-down prediction model and $p(X(C_i), X(C_j))$ captures the compatibilities in the bottom-up binding model. They will be specified in the learning algorithm in Sec.4.

Given an input image I with domain defined on lattice Λ, the inference of AoG is to construct a parse graph for each object instance and its structure is not predefined but inferred on the fly.

Configuration. A configuration $\mathcal{C}$ is the set of all terminal nodes in a valid parse graph pg, flattened in an image lattice.

$$\mathcal{C}(pg) = \{(t, x(t)) : t \in V_T^{pg}\} \tag{5}$$

The image data likelihood of a parse graph pg, $p(I|pg)$, is measured based on the terminal nodes in V_T^{pg} (since they link to image data). Further, if there was no occlusion between different terminal nodes (which is true for roughly rigid object categories such as the human face), we can factorize the likelihood as,

$$p(I|pg) = p(I|\mathcal{C}(pg)) = \prod_{t \in V_T^{pg}} p(I_{\Lambda_t}|t) \tag{6}$$

where $\Lambda_t \in \Lambda$ is the image domain occupied by the terminal node t.

In inference, we do not need compute $p(I|pg)$ exactly, instead we measure the likelihood ratio between $p(I|pg)$ and a reference background model which is made implicitly in our derivation.

2.2 Bayesian formulation of object parsing using AoG

An AoG represents the object grammar of an object category. Given an input image I_Λ, it contains an unknown number K object instances at different scales. Some object instances may be occluded. Each object instance is represented by a parse graph pg_k ($k = 1, \cdots, K$). For the human face parsing, Fig.1 shows a typical testing image and the left-bottom panel in Fig.5 shows a number of inferred parse trees of human face instances.

The goal of object parsing using AoG is to construct a parse graph for each object instance in I_Λ on the fly. We seek a world representation W for image I_Λ,

$$W = (K, \{pg_k\}_{k=1}^K) \tag{7}$$

Under the Bayesian framework, we infer W by maximizing a posterior probability,

$$W^* = \arg\max_{W \in \Omega} p(W|I_\Lambda) = \arg\max_{W \in \Omega} p(W) p(I_\Lambda|W) \tag{8}$$

where Ω is the solution space.

The prior probability $p(W)$ is,

$$p(W) = p(K) \prod_{k=1}^K p(pg_k) \tag{9}$$

where $p(K)$ is the prior distribution for the number of object instances (for example, an exponential model $p(K) \propto \exp\{-\lambda_0 K\}$) and $p(pg_k)$ is the prior model of a parse graph already addressed in Eqn.3 in Sec.2.1.

The likelihood $p(I_\Lambda|W)$. Let Λ_{pg_k} be the image lattice occupied by the parse graph pg_k ($1 \le k \le K$). Denote $\Lambda_{fg} = \cup_{k=1}^K \Lambda_{pg_k}$ as the foreground lattice and $\Lambda_{bg} = \Lambda \setminus \Lambda_{fg}$ as the remaining background lattice. $I_\Lambda = (I_{\Lambda_{fg}}, I_{\Lambda_{bg}})$. Let $q(I)$ be the generic background model which will be made implicit in our derivation. The likelihood $p(I_\Lambda|W)$ is,

$$\begin{aligned} p(I_\Lambda|W) &= p(I_{\Lambda_{fg}}|W) q(I_{\Lambda_{bg}}) \\ &= \frac{p(I_{\Lambda_{fg}}|W) q(I_{\Lambda_{bg}}) q(I_{\Lambda_{fg}})}{q(I_{\Lambda_{fg}})} \\ &= q(I_\Lambda) \frac{p(I_{\Lambda_{fg}}|W)}{q(I_{\Lambda_{fg}})} \\ &= q(I_\Lambda) \prod_{k=1}^K \frac{p(I_{\Lambda_{pg_k}}|pg_k)}{q(I_{\Lambda_{pg_k}})} \end{aligned} \tag{10}$$

where $p(I_{\Lambda_{pg_k}}|pg_k)$ means that the domain Λ_{pg_k} is explained away by the parse graph pg_k and conversely, $q(I_{\Lambda_{pg_k}})$ explains the domain Λ_{pg_k} as background. These models compete with each other to perform parsing.

So, Eqn.8 can be reproduced as,

$$\begin{aligned} W^* &= \arg\max_{W\in\Omega} p(K)q(I_\Lambda)\prod_{k=1}^{K}[p(pg_k)\frac{p(I_{\Lambda_{pg_k}}|pg_k)}{q(I_{\Lambda_{pg_k}})}] \\ &= \arg\max_{W\in\Omega} p(K)\prod_{k=1}^{K}[p(pg_k)\frac{p(I_{\Lambda_{pg_k}}|pg_k)}{q(I_{\Lambda_{pg_k}})}] \end{aligned} \tag{11}$$

3 Object parsing in a greedy pursuit manner

In the literature, there are several ways to infer W^* in Eqn.11, such as the data-driven Markov chain Monte Carlo (DDMCMC) method used in (Tu and Zhu, 2002). In this paper, our goal is to pursue object instances appearing in an input image and construct corresponding parse graphs. Often, the number of object instances K is typically not too large. So, we adopt the best-first-search algorithm (Dechter and Pearl, 1985) directly to pursue parse graphs sequentially by maximizing Eqn.11 in a greedy manner. Our pursuit inference algorithm integrates the α β and γ processes and includes two aspects: (i) generating proposals (hypotheses) for possible parse graphs and (ii) verifying parse graph proposals in a greedy pursuit manner.

3.1 Connecting the α, β and γ processes with Bayesian inference

We pursue parse graphs sequentially based on Eqn.11 starting from an empty $W_0 = \emptyset$,

$$W_0 = \emptyset \to W_1 \to \cdots \to W_k \to \cdots \to W_K = W^*$$

At each step we pursue a parse graph and at the step k (≥ 1) of pursuit, let $\Lambda_k = \Lambda \setminus \cup_{i=1}^{k-1}\Lambda_{pg_i}$. We pursue the k-th parse graph pg_k by,

$$pg^* = \arg\max_{pg\in\Omega_{pg}} p(pg)p(I_{\Lambda_{pg}}|pg) \tag{12}$$

where Ω_{pg} is the proposal space of parse graphs and we omit k in pg_k hereafter in the derivation for simplicity when there is no confusion.

Similar to derive Eqn.10, we have,

$$p(I_{\Lambda_{pg}}|pg) = q(I_{\Lambda_k})\frac{p(I_{\Lambda_{pg}}|pg)}{q(I_{\Lambda_{pg}})} \tag{13}$$

So, Eqn.12 can be rewritten as,

$$\begin{aligned} pg^* &= \arg\max_{pg\in\Omega_{pg}} p(pg)\frac{p(I_{\Lambda_{pg}}|pg)}{q(I_{\Lambda_{pg}})} \\ &= \arg\max_{pg\in\Omega_{pg}} [\log p(pg) + \log\frac{p(I_{\Lambda_{pg}}|pg)}{q(I_{\Lambda_{pg}})}] \end{aligned} \tag{14}$$

which is consistent with Eqn.11.

Recall that the prior probability $p(pg)$ is defined in Eqn.3 in general. For object categories with roughly rigid configuration such as the human face, we can assume that there are no occlusion among different nodes at the same layer in a parse graph so that we can factorize the likelihood ratio $\frac{p(I_{\Lambda_{pg}}|pg)}{q(I_{\Lambda_{pg}})}$ with respect to Eqn.6,

$$\log\frac{p(I_{\Lambda_{pg}}|pg)}{q(I_{\Lambda_{pg}})} = \sum_{t\in V_T^{pg}} \log\frac{p(I_{\Lambda_t}|t)}{q(I_{\Lambda_t})} \tag{15}$$

Without loss of generality, we consider the AoG illustrated in Fig.4. Node A represents the object of interest such as the human face. $V_{and} = \{P, A, C_1, C_2, C_3\}$. Further, we consider a parse graph pg in which $V_{and}^{pg} = \{P, A, C_1, C_2\}$ and $V_T^{pg} = \{t_P, t_A, t_{C_1}, t_{C_2}\}$. In terms of Eqn.3, we have,

$$\begin{aligned} \log p(pg) = &\log p(A|O) + \log p(X(A)|X(P)) \\ &+ \sum_{i=1}^{2}\log p(X(C_i)|X(A)) \\ &+ \log p(X(C_1), X(C_2)) + \log Z \end{aligned} \tag{16}$$

By combining Eqn.15 and Eqn.16, Eqn.14 can be rewritten as,

$$\begin{aligned} pg^* = \arg\max_{pg}\{&\log p(A|O) + \sum_{i=1}^{2}\log p(X(C_i)|X(A)) \\ &+ \underbrace{\log\frac{p(I_{\Lambda_{t_A}}|t_A)}{q(I_{\Lambda_{t_A}})}}_{\alpha(A)\text{ process}} \\ &+ \underbrace{[\underbrace{\sum_{i=1}^{2}\log\frac{p(I_{\Lambda_{t_{C_i}}}|t_{C_i})}{q(I_{\Lambda_{t_{C_i}}})}}_{\alpha(t_{C_i})\text{ process}} + \underbrace{\log p(X(C_1), X(C_2))}_{\text{binding model}}]}_{\beta(A)\text{ process}} \\ &+ \underbrace{[\underbrace{\log\frac{p(I_{\Lambda_{t_P}}|t_P)}{q(I_{\Lambda_{t_P}})}}_{\alpha(P)\text{ process}} + \underbrace{\log p(X(A)|X(P))}_{\text{prediction model}}]}_{\gamma(A)\text{ process}}\} \end{aligned} \tag{17}$$

where $p(X(C_i)|X(A))$ is the prediction model for the child node C_i of node A. From Eqn.17, we can see that

our pursuit algorithm integrates the α, β and γ processes explicitly.

Define w_A^{α}, $w_A^{\beta(\mathbf{c})}$ and $w_A^{\gamma(P)}$ as the weights computed from the α, β and γ processes respectively,

$$w_A^{\alpha} = \log \frac{p(I_{\Lambda_{t_A}}|t_A)}{q(I_{\Lambda_{t_A}})} \tag{18}$$

$$w_A^{\beta(\mathbf{c})} = \sum_{i=1}^{2} \log \frac{p(I_{\Lambda_{t_{C_i}}}|t_{C_i})}{q(I_{\Lambda_{t_{C_i}}})} + \log p(X(C_1), X(C_2)) \tag{19}$$

$$w_A^{\gamma(P)} = \log \frac{p(I_{\Lambda_{t_P}}|t_P)}{q(I_{\Lambda_{t_P}})} + \log p(X(A)|X(P)) \tag{20}$$

which will be specified by the learning algorithm in Sec.4. Then, we can rewrite Eqn.17 as,

$$pg^* = \arg\max_{pg \in \Omega_{pg}} \{\log p(A|O) + \sum_{i=1}^{2} \log p(X(C_i)|X(A)) + \underbrace{w_A^{\alpha}}_{\alpha(A) \text{ process}} + \underbrace{w_A^{\beta(\mathbf{c})}}_{\beta(A) \text{ process}} + \underbrace{w_A^{\gamma(P)}}_{\gamma(A) \text{ process}} \} \tag{21}$$

Now, we can introduce the formal specifications of the α, β and γ processes by connecting the general identifications of the three processes in Sec.1.2 with the Bayesian inference in terms of Eqn.17.

I. The α process detects node A by applying a log-likelihood ratio test, $\log \frac{p(I_{\Lambda_{t_A}}|t_A)}{q(I_{\Lambda_{t_A}})}$ (Eqn.18), directly based on image features when node A is at middle resolution without occlusion. The α process can be viewed as either bottom-up (feature classifiers such as the Adaboost method) or top-down (template matching such as the active basis model) inference process. For each And-node $A \in V_{and}$, the α process, denoted by $\alpha(A;\theta)$, is instantiated by a corresponding terminal node $t_A \in V_T$, where θ is a set of parameters. For example, in Fig.4, we have,

$$\alpha(A;\theta) : t_A \to A \text{ and } x(t_A) \Rightarrow X(A) \tag{22}$$

where $t_A \to A$ is calculated by w_A^{α} in Eqn.18 and will be specified in Eqn.32, and $x(t_A) \Rightarrow X(A)$ is used to activate the γ processes of node A's child nodes, $p(X(C_i)|X(A))$, in Eqn.17 and the β process of node A's parent node, $p(X(P)|X(A))$.

II. The β process computes node A by applying a bottom-up binding test, for example $\log p(X(C_1), X(C_2))$ (in Eqn.19), of its child nodes $\mathbf{c} = (C_1, C_2)$ which have been detected in a given step based on the log-likelihood ratio tests of their own α processes, $w_{C_i}^{\alpha} = \log \frac{p(I_{\Lambda_{t_{C_i}}}|t_{C_i})}{q(I_{\Lambda_{t_{C_i}}})}$. The β process handles the situation in which node A is at high resolution but with occlusion (the occlusion disable the $\alpha(A)$ process). Let $V_{and}^{ch} \subset V_{and}$ be the set of And-nodes which have children. For each node $A \in V_{and}^{ch}$, the β process of node A can be defined, denoted by $\beta(A|\mathbf{c};\phi)$ where $\mathbf{c} \subseteq ch(A)$ and ϕ is a set of parameters. Given different $\mathbf{c}$'s, we obtain different β processes for node A. Consider $\mathbf{c} = (C_1, C_2)$ in Fig.4, we have,

$$\begin{aligned} \beta(A|\mathbf{c};\phi) : & \, t_{C_1} \to C_1 \text{ and } x(t_{C_1}) \Rightarrow X(C_1) \\ & \, t_{C_2} \to C_2 \text{ and } x(t_{C_2}) \Rightarrow X(C_2) \\ & \, (C_1, C_2) \to A \text{ and } (X(C_1), X(C_2)) \Rightarrow X(A) \end{aligned} \tag{23}$$

where $t_{C_i} \to C_i$ are calculated by $w_{C_i}^{\alpha}$ $(i = 1, 2)$, and $(X(C_1), X(C_2)) \Rightarrow X(A)$ will activate the β binding process of node A. Then, we calculate $w_A^{\beta(\mathbf{c})}$ which will be specified by Eqn.45. The $\beta(A|\mathbf{c};\phi)$ will, in turn, activate the γ processes of the other child nodes of node A and the β process of node A's parent node. Actually, this procedure is activated recursively in testing.

III. The γ process computes node A by applying a top-down prediction test, $\log p(X(A)|X(P))$ (in Eqn.20), from its parent node P which has been already detected in a given step based on the log-likelihood ratio test of the α process of node P, $w_P^{\alpha} = \log \frac{p(I_{\Lambda_{t_P}}|t_P)}{q(I_{\Lambda_{t_P}})}$. The γ process handles the situation in which node A is under very low resolution so both $\alpha(A)$ and $\beta(A)$ are disabled. Let $V_{and}^{prt} \subset V_{and}$ be the set of And-nodes which have parent node(s). For each node $A \in V_{and}^{prt}$, we can define its γ process, denoted by $\gamma(A|P;\varphi)$ where $P \in prt(A)$ is a parent node and φ is a set of parameters. Similarly, in Fig.4 we have,

$$\begin{aligned} \gamma(A|P;\varphi) : & \, t_P \to P \text{ and } x(t_P) \Rightarrow X(P) \\ & \, P \to A \text{ and } X(P) \Rightarrow X(A) \end{aligned} \tag{24}$$

where similarly, $t_P \to P$ is calculated by w_P^{α} and $X(P) \Rightarrow X(A)$ will activate the γ process of node A. Then, we calculate $w_A^{\gamma(P)}$ which will be specified by Eqn.52, and then we run the inference process recursively.

3.2 The algorithm

Our greedy pursuit algorithm is straightforward based on Eqn.17. Fig.5 shows a running example of human face parsing by the proposed algorithm. On the whole, the algorithm first runs all α processes (see the top panel in Fig.5) and applies thresholds to obtain candidates for each node. Then, to pursue object instances of node A, the algorithm recursively runs all β processes and γ processes to do bottom-up binding and top-down

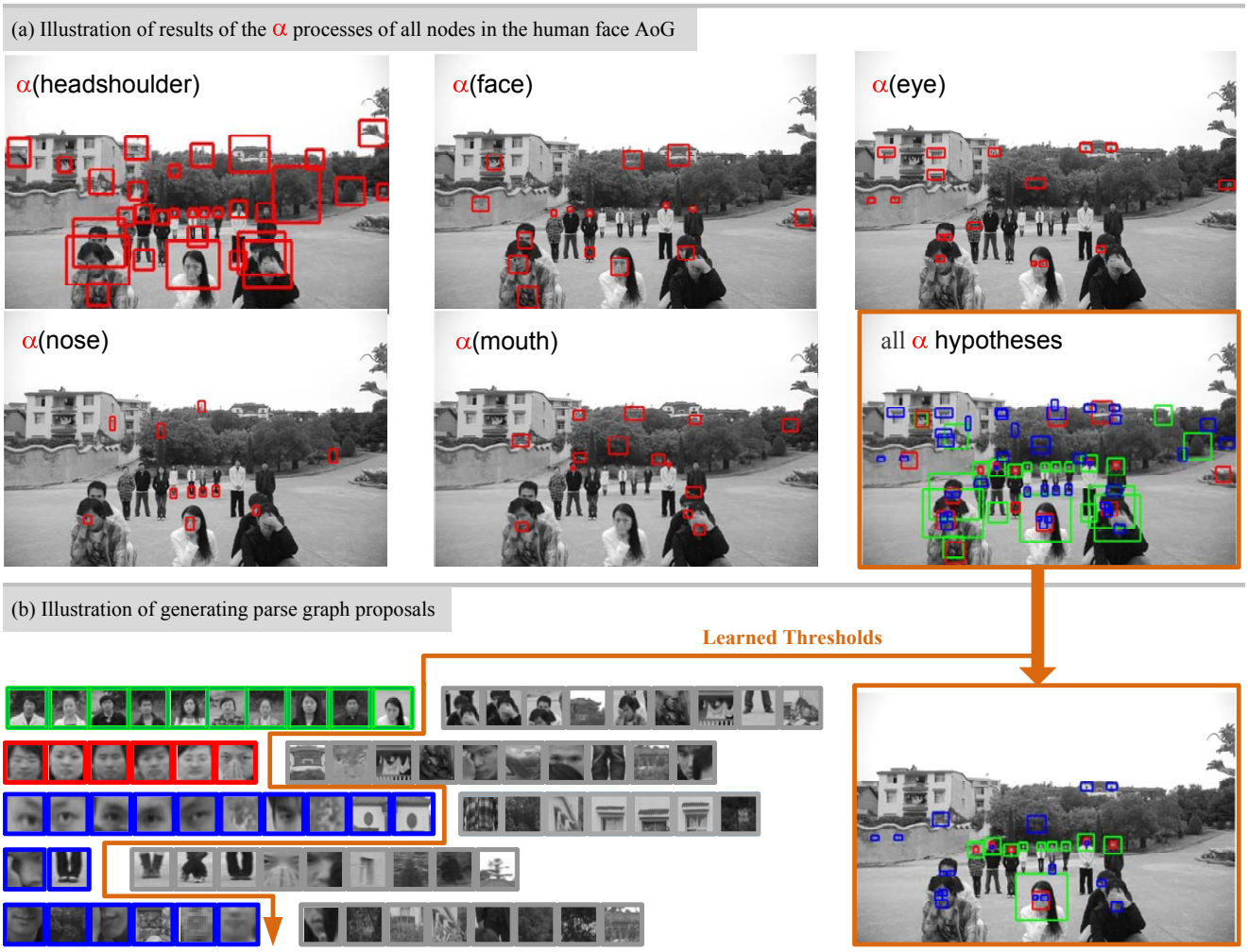

Fig. 5 A running example of pursuing human faces and constructing corresponding parse graphs in a typical image by integrating the α, β and γ processes. The learning algorithms of the three processes are specified in Sec.4. In this figure, (a) shows the results of running all the α processes with a number of candidates of each node. (b) illustrates how we generate parse graph proposals. After applying the learned thresholds, we get promising candidates of each node and then run all the β and γ processes to propose possible parse graphs. (c) shows the results of pursuing object instances and constructing their parse graphs on the fly by integrating the α, β and γ processes in the proposed algorithm. In a greedy manner, we can get the pursuit indexes of all the object instances. For each object instance, we know the construction of the parse graph explicitly, which goes beyond only a bounding box for each detected object instance in traditional object detection. (Best viewed in color)

prediction and then generate the parse graph proposals (see the middle panel in Fig.5). The pursuit is based on the pursuit index in Eqn.25. The parse graph for each pursued instance of node A is constructed by retrieving all the used α candidates (see the bottom panel in Fig.5). We summarize the algorithm in Fig.6.

Proposal generation Ω_{pg}. In testing, in order to find the object instances which appear at different locations, scales and orientations, we need search these three spaces to find possible proposals. For searching the scale space, we create the testing image pyramid for I_Λ by a certain down-sampling factor (we use 0.9 in our experiments) until the image size is smaller than the minimum of sizes of the learned active basis templates. We have the pyramid with L layers $\{I_{\Lambda_0} = I_\Lambda, I_{\Lambda_1}, \cdots, I_{\Lambda_L}\}$. For different orientations, we rotate the learned active basis templates. We use the sliding window method to search all locations in the testing pyramid for all active basis templates. That is, we run all the α processes first and by applying the thresholds we get a list of candidates for each node in the AoG (which could be empty). Based on the list, we can generate parse graph proposals Ω_{pg}. Since we used fixed relative scales in training the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes (as stated in Sec.4.2), for example, given a hypothesis of node A (such as the human face), we roughly know the locations, scales and orientations of other nodes in terms of the learned β binding models $p(X(C_i)|X(A))$ and γ prediction models $p(X(A)|X(P))$. Then, we put all proposals in an open list which could be complete or partial parse graphs. Further, different parse graphs could also overlap to compete to explain the corresponding image domain. The open list would be explored to do the proposal verification and the ambiguity is solved based on the pursue index addressed below.

Pursuit index. For each proposal, we compute its total weight,

$$w_A = w_A^\alpha + w_A^{\beta(\mathbf{c})} + w_A^{\gamma(P)} \tag{25}$$

which is the pursuit index we are seeking for proposal verification. Similarly, we can estimate the threshold $\text{Th}(w_A)$ in an evaluation dataset.

Performance comparisons. According to Eqn.25, we can explicitly compare the performance of different integrations for each node A, for example, by plotting ROCs based on w_A^α, $w_A^\alpha + w_A^{\beta(\mathbf{c})}$, $w_A^\alpha + w_A^{\gamma(P)}$ and w_A respectively. Some comparison results are shown in Fig.16 for junctions and rectangles and Fig.17 for human faces. From these ROCs, we can see that how and how much the integration improves performance.

Object parsing by integrating the α, β and γ processes

Input: an image I_Λ and an AoG $\mathcal{G}$.
Output: parse graphs pg_i $(i = 1, \cdots, K)$.

1. α **map generation:** $I_{\alpha(U)}, \forall U \in V_{and}$
   run $\alpha(U; \theta_U)$ and compute the weight w_U^α.

2. α **hypotheses generation:** open list $\text{OP}(U)$ from $I_{\alpha(U)}$
   apply thresholds $\text{Th}(w_U^\alpha)$ and local inhibitions.

3. β **bindings and merging.**
   (1) run $\beta(A|\mathbf{c}; \phi)$ and compute the weight $w_A^{\beta(\mathbf{c})}$
   (2) apply $\text{Th}(w_A^{\beta(\mathbf{c})})$ to generate $\beta(A)$ hypotheses and insert them into $\text{OP}(A)$ decreasingly;
   (3) merge with compatible $\alpha(A)$ hypotheses and compute weight $w_A^\alpha + w_A^{\beta(\mathbf{c})}$.

4. γ **predictions and merging.**
   (1) run $\gamma(A|P; \varphi)$ and compute the weight $w_A^{\gamma(P)}$
   (2) apply $\text{Th}(w_A^{\gamma(P)})$ to generate $\gamma(A)$ hypotheses and insert them into $\text{OP}(A)$ decreasingly;
   (3) merge with compatible $\alpha(A)$, $(\alpha + \beta)(A)$ hypotheses and compute weight $w_A^\alpha + w_A^{\gamma(P)}$ or $w_A^\alpha + w_A^{\beta(\mathbf{c})} + w_A^{\gamma(P)}$.

5. **Object pursuing and parsing.**
   In $\text{OP}(A)$, pursue node A according to w_A, construct parse graphs by retrieving all the α hypotheses. Stop pursuing based on $\text{Th}(w_A)$.

Fig. 6 The greedy pursuit algorithm for object parsing using AoG by integrating the α, β and γ processes

4 Learning the α, β and γ processes

In this section, we introduce the learning algorithm under the MLE framework for the α, β and γ processes to specify w_A^α in Eqn.18, $w_A^{\beta(\mathbf{c})}$ in Eqn.19 and $w_A^{\gamma(P)}$ in Eqn.20 respectively. We train the three processes separately due to the observation that the effectiveness of the three processes depends on the scale and occlusion conditions as illustrated in Fig.1 and Fig.3, and for the purpose of evaluating the information contribution of each process individually. To that end, we propose an isolation method to block one process from the other two processes. The isolation is based on the manually labeled parse graphs in this paper which are available in the LHI image database (Yao et al, 2007).

4.1 Isolating the α, β and γ processes.

Scale and occlusion are the two main causes entailing the α, β and γ processes. So, each of the three processes of node A can be blocked through *scaling* and/or *masking* image patches of node A in terms of the labeled

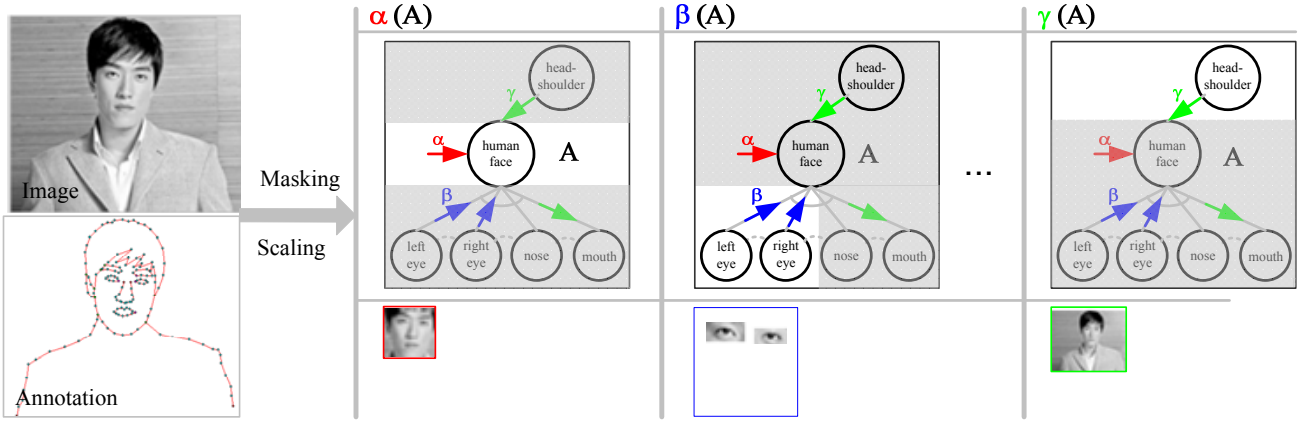

Fig. 7 Illustration of isolating the α, β and γ processes of node A in an AoG. Consider the human face example in this figure. The isolation is performed by *scaling* and *masking* the image patch (the left-top panel) in terms of its annotation (the left-bottom panel). The annotation used in this paper is the manually labelled parse graph. Details are specified in Sec.4.1. Based on the isolation, we generate training and evaluation data for each process in Sec.4.2.

parse graphs. Then, we isolate one process by blocking the other two processes. Fig.7 illustrates the procedures with an example for human face (node A).

I. Isolating the $\alpha(A)$ process. Block both the $\beta(A)$ and $\gamma(A)$ processes will isolate the α process. First, we crop only the compact image patches of node A out of its context in terms of the annotations. Then, the image patches are down-sampled up to a certain scale at which the parts can not be recognized if cropped in isolation.

II. Isolating the $\beta(A)$ process. We have different $\beta(A)$ processes depending on the given subset of node A's child nodes, $\mathbf{c} \subseteq ch(A)$. Then, to isolate the $\beta(A|\mathbf{c})$ process is is to block both the $\alpha(A)$ and $\gamma(A)$ processes meanwhile keep the α processes of child nodes in $\mathbf{c}$ on. We first crop only the compact image patches of node A but just keep those patches whose resolutions are above a predefined value. Then, we scale all the image patches to the same size (also above the predefined value) and mask those portions of all the image patches with respect to the child node(s) not in $\mathbf{c}$.

III. Isolating the $\gamma(A)$ process. We may have different γ processes given different parent nodes P. To isolate the $\gamma(A|P)$ process is to block the $\alpha(A)$ and $\beta(A)$ while keeping the $\alpha(P)$ process on. So, it is equivalent to isolate the $\alpha(P)$ process. First, we crop the compact image patches of the parent node P. Then, we down-sample the image patches up to a certain scale at which the node A itself can not be recognized if cropped in isolation.

By changing the testing image dataset of an object category with these isolating methods, we can cause most of existing object detection or recognition methods to fail. To achieve robust performance, we train each process separately first and then integrate them explicitly (see Eqn.17) for our numerical study.

Next, we generate training and evaluation data for each process in terms of the isolating procedures.

4.2 Training data for the α, β and γ processes

Suppose we have a set of m positive images for an object of interest, $D^+ = \{(I_1, pg_1), \cdots, (I_m, pg_m)\}$, where pg_i is the annotated parse graph for image I_i. Based on the parse graph, we can generate training and testing datasets for the three processes. For simplicity of notations, we assume that each node of interest appears in each image I_i with good resolution.

I. The α process training dataset. Let $D^+_\alpha(A)$ denote the positive training dataset for the $\alpha(A)$ process of node A. Through the isolation method of the α process, for each $I_i \in D^+$, we obtain the α image patch of node A, denoted by $I_i^{(A)}$. So, we have,

$$D^+_\alpha(A) = \{I_i^{(A)} : i = 1, 2, \cdots, m\}$$

II. The β process training dataset. Let $D^+_\beta(A|\mathbf{c})$ denote the positive training dataset for the $\beta(A|\mathbf{c})$ process. Through the isolation method of the β process, for each $I_i \in D^+$, we obtain the β image patch of node A given child node(s) in $\mathbf{c}$, denoted by $I_i^{(\mathbf{c})}$. Then, we get,

$$D^+_\beta(A|\mathbf{c}) = \{(I_i^{(\mathbf{c})}, X(\mathbf{c}|I_i^{(\mathbf{c})})) : i = 1, 2, \cdots, m\}$$

where $X(\mathbf{c}|I_i^{(\mathbf{c})}) = \{X(C_j) : C_j \in \mathbf{c}, j = 1, \cdots, |\mathbf{c}|\}$ is the concatenation of attributes for child node(s) in $\mathbf{c}$ measured in $I_i^{(\mathbf{c})}$, which will be used to learn the bottom-up β binding model $p(X(C_j), X(C_k))$'s for node A given child nodes in $\mathbf{c}$ $(j \neq k, j, k = 1, \cdots, |\mathbf{c}|)$.

III. The γ process training dataset. Let $D^+_\gamma(A|P)$ denote the positive training dataset for the $\gamma(A|P)$ process. Through the isolation method of the γ process, for each $I_i \in D^+$, we can get the γ image patch of node A given the parent node P, denoted by $I_i^{(P)}$. So, we have,

$$D^+_\gamma(A|P) = \{(I_i^{(P)}, X(A|I_i^{(P)})) : i = 1, 2, \cdots, m\}$$

where $X(A|I_i^{(P)})$ is the attributes of node A measured in $I_i^{(P)}$, which will be used to learn the top-down γ prediction model $p(X(A)|X(P))$ of node A given the parent node P.

When node P only has one child node A, we can transform the γ prediction model of node A given the parent node P $p(X(A)|X(P))$ into the β binding model of node P given the child node A equivalently.

In the same way, we can get the attributes $X(C_j|I_i^{(A)})$ to learn top-down γ prediction models $p(X(C_j)|X(A))$'s for child nodes C_j's of node A where $C_j \in ch(A)$, $I_i^{(A)} \in D^+_\alpha(A)$.

Correspondingly, we collect negative datasets $D^-_\alpha(A)$, $D^-_\beta(A|\mathbf{c})$ and $D^-_\gamma(A|P)$ by randomly cropping image patches from generic background images.

Scale specifications in experiments. In the experiments for evaluating the information contributions in Sec.6.1, we prepare the data for multiple scales to observe how the information contributions change with scales. In the experiments for object parsing by integrating the α, β and γ processes of node A in Sec.6.2, we use fixed relative scales for the three processes. Consider the scales of the compact image patches of node A in these three processes, denoted by $s_{\alpha(A)}$, $s_{\beta(A)}$ and $s_{\gamma(A)}$ respectively. We set $s_{\alpha(A)} = b \times s_{\gamma(A)} = \frac{1}{b} \times s_{\beta(A)}$ ($b = 2$ used in our current experiments).

Given the data, we specify the training procedure under the MLE framework in the next section.

4.3 Learning the α process

Learning the α process involves selecting a modeling scheme for $\alpha(A;\theta)$ and estimating the parameters θ by maximizing the data likelihood on $D^+_\alpha(A)$. For example, in discriminative boosting methods, θ_A is the learned strong classifier which consists of a set of boosted weak classifiers and the corresponding weights (Viola and Jones, 2004), and in generative model-based methods such as the active basis model, θ is the set of parameters specify the learned deformable template (Wu et al, 2009). Given $D^+_\alpha(A)$, we have,

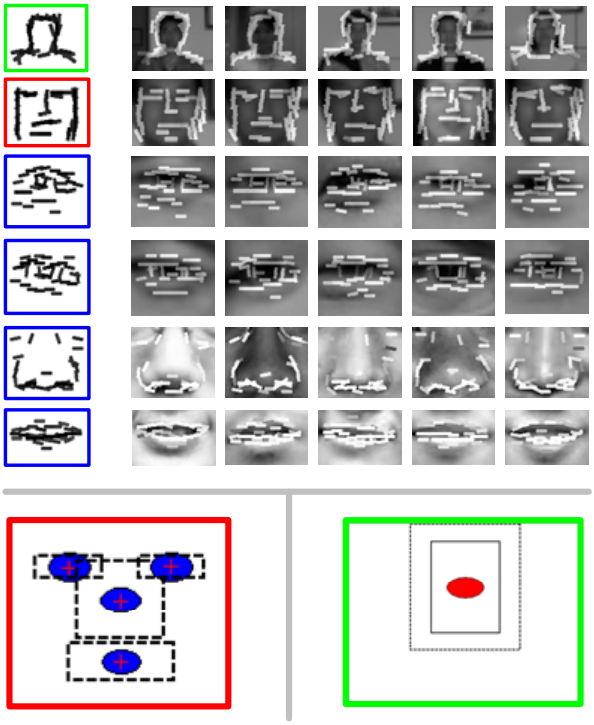

Fig. 8 Illustration of learned α, β binding and γ prediction models for human face. The top panel shows the learned active basis model for the α process of each terminal node. The left-bottom panel illustrates the binding model for the β process in which the outside red box is the bounding box of face and the inside dash boxes are for the parts and the ellipses represent the location following a Gaussian distribution. The right-bottom panel shows the prediction model for the γ process in which the outside green box is the bounding box of head-shoulder and the inside solid and dash boxes represent the changeable size of the bounding box of face and the red ellipse represent the location of face following a Gaussian distribution.

$$\begin{aligned}\alpha(A;\theta^*) &= \arg\max_\theta p(D^+_\alpha(A)|A;\theta) \\ &= \arg\max_\theta \sum_{i=1}^{m} \log p(I_i^{(A)}|A;\theta) \end{aligned} \tag{26}$$

Solving $\alpha(A;\theta^*)$ depends on choosing a specific modeling scheme for $p(I|A;\theta)$. In this paper, we use the active basis model which is briefly introduced here for this paper to be self-contained.

Active basis model. The active basis model is a deformable model which consists of a small number of Gabor wavelet elements (as visual primitives for modeling object category) at selected locations and orientations. These Gabor wavelet elements can slightly perturb their locations and orientations before they are linearly combined to generate the observed image. Let Λ be the domain of the image patch I and $\{B_{x,y,s,o}\}$ the dictionary of Gabor wavelet elements. The (x, y, s, o) are densely

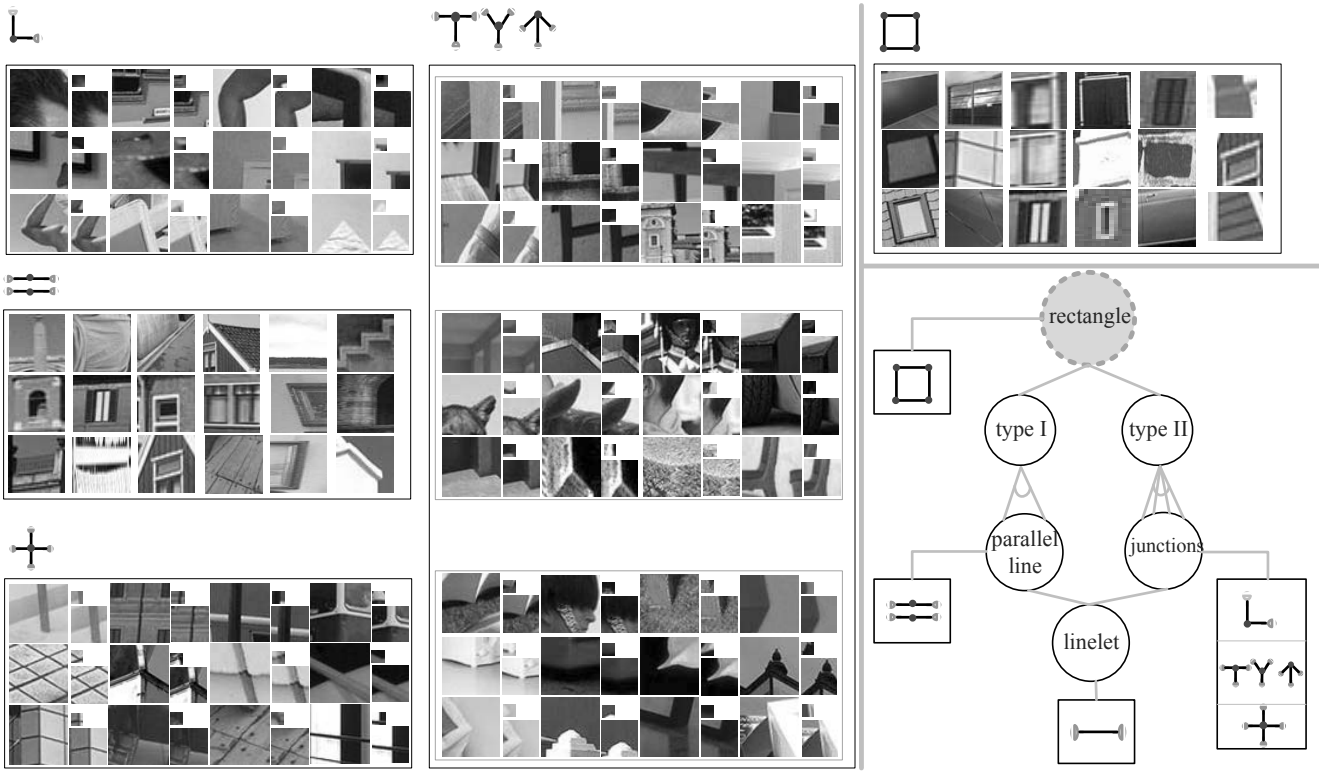

Fig. 9 Illustration of the AoG for junctions and rectangles. The left panel shows some positive examples of L junction, cross junction, parallel line, T/Y/Arrow junction. Each sample of L, cross and T/Y/Arrow junctions is shown under three different scales (10×10 pixels, 20×20 pixels and 30×30 pixels) from which we can intuitively see that the α process would be very weak. The right-top panel shows some samples of rectangle in which we can also know that the α process would not work well due to the variabilities. The right-bottom panel shows the AoG for rectangle.

sampled: $(x, y) \in \Lambda$, s is a fixed size (often about 1/10 of the length of Λ) and $o \in \{i\pi/N, i = 0, \cdots, N-1\}$ (e.g.. $N = 15$). The dictionary forms an over-complete dictionary for modeling I_Λ. Then we obtain the sparse coding scheme $I = \sum_{i=1}^{n} a_i B_i + U$ where n is the number of selected bases, $B_i = B_{x_i, y_i, s, 0_i}$, a_i's are the coefficients and U is the unexplained residual image. In the matrix form, we have $I = B\mathbf{a} + U$ (where $B = (B_1, \cdots, B_n)$ and $\mathbf{a} = (a_1, \cdots, a_n)^{'}$). In terms of linear decomposition, we know that U resides in the remaining subspace orthogonal to B and we can write $U = \bar{B}\bar{\mathbf{a}}$ (where columns in $\bar{B}$ are orthogonal to columns in B and both $\bar{B}$ and $\bar{\mathbf{a}}$ would be made implicit in the active basis model). So, we have $I = B\mathbf{a} + \bar{B}\bar{\mathbf{a}}$. Then, we can specify the distribution of I given B as

$$p(I|B) = p(\mathbf{a}, \bar{\mathbf{a}}) \det(J) = p(\mathbf{a}) p(\bar{\mathbf{a}}|\mathbf{a}) \det(J) \tag{27}$$

where J is the Jacobi matrix of the linear transform from I to $(\mathbf{a}, \bar{\mathbf{a}})$ and $\det(J)$ the determinant of J.

On the other hand, let $q(I)$ be a reference distribution (which has a few choices discussed in (Wu et al, 2009)), and similarly, we can have

$$q(I) = q(\mathbf{a}, \bar{\mathbf{a}}) \det(J) = q(\mathbf{a}) q(\bar{\mathbf{a}}|A) \det(J) \tag{28}$$

In the active basis model, we want to construct $p(I|B)$ by modifying $q(I)$ and assume $q(\bar{\mathbf{a}}|\mathbf{a}) = p(\bar{\mathbf{a}}|\mathbf{a})$, so we have

$$p(I|B) = q(I) \frac{p(\mathbf{a})}{q(\mathbf{a})} = q(I) \frac{p(a_1, \cdots, a_n)}{q(a_1, \cdots, a_n)} \tag{29}$$

Further, by applying the local inhibition principle, we can treat the selected Gabor wavelet elements independently, that is,

$$p(I|B) = q(I) \prod_{i=1}^{n} \frac{p(a_i)}{q(a_i)} \tag{30}$$

where $p(a_i)$ is parameterized as an exponential family model $p(a_i; \lambda_i) = \frac{1}{Z(\lambda_i)} \exp\{\lambda_i h(r_i)\} q(a_i)$ ($r_i = | < I, B_i > |^2$ is the local energy of Gabor filter response and $h(r_i) = sigmoid(r_i) = \zeta[\frac{2}{1+e^{-2r_i/\zeta}} - 1]$ is a sigmoid transformation function with ζ being the saturation level such as $\zeta = 6$), and $q(a_i)$ is pooled from generic background images at an off-line stage. The resulting model is

$$p(I|B) = q(I) \prod_{i=1}^{n} \frac{1}{Z(\lambda_i)} \exp\{\lambda_i h(r_i)\} \tag{31}$$

In testing, the matching score (the α weight of a hypothesis of node A) is the log-likelihood ratio,

$$w_A^{\alpha} = \log \frac{p(I|B)}{q(I)} = \sum_{i=1}^{n} [\lambda_i h(r_i) - \log Z(\lambda_i)] \tag{32}$$

Active basis model can be also used to learn mixed image template modeling both shape and texture (Wu et al, 2009; Si et al, 2009).

The threshold $\mathrm{Th}(w_A^{\alpha})$ of the α process $\alpha(A;\theta)$ can be estimated in a validation α dataset.

The top panel of Fig.8 shows the learned active basis for each terminal node of a face AoG.

4.4 Learning the β process

Learning the β process involves specifying $\beta(A|\mathbf{c};\phi)$ and estimating the parameters ϕ by maximizing the data likelihood on $D_{\beta}^{+}(A|\mathbf{c})$. $\beta(A|\mathbf{c};\phi)$ composes (or binds) a (complete or partial) set of child nodes in $\mathbf{c}$, to generate hypotheses of node A. In the literature, component-based (Biederman, 1987; Heisele et al, 2007), fragment-based (Ullman et al, 2002) and other part-based methods (Amit and Trouvé, 2007; Wu et al, 2007) can be treated as this kind of process. Given $D_{\beta}^{+}(A|\mathbf{c})$, we obtain,

$$\begin{aligned}
&\beta(A|\mathbf{c};\phi^*) \\
&= \arg\max_{\phi} p(D_{\beta}^{+}(A|\mathbf{c})|\mathbf{c};\phi) \\
&= \arg\max_{\phi} \sum_{i=1}^{m} \log p(I_i^{(\mathbf{c})}, X(\mathbf{c}|I_i^{(\mathbf{c})})|\mathbf{c};\phi)
\end{aligned} \tag{33}$$

The $\beta(A|\mathbf{c};\phi)$ process includes two components, one is the α process of each child node in $\mathbf{c}$ with parameters $\theta_{\mathbf{c}}$ and the other is the binding model for the given children $\mathbf{c}$ with parameters $\triangle$. So, $\phi = (\theta_{\mathbf{c}}, \triangle)$ and we have,

$$\begin{aligned}
&p(I^{(\mathbf{c})}, X(\mathbf{c}|I^{(\mathbf{c})})|\mathbf{c};\phi) \\
&= p(I^{(\mathbf{c})}|\mathbf{c};\theta_{\mathbf{c}}) \times p(X(\mathbf{c}|I^{(\mathbf{c})});\triangle)
\end{aligned} \tag{34}$$

where for notation simplicity we use $I^{(\mathbf{c})}$ to represent $I_i^{(\mathbf{c})}$ generally.

In this paper, we consider three types of attributes for binding, that is, the location $L_{\mathbf{c}}$, scale $S_{\mathbf{c}}$ and orientation $O_{\mathbf{c}}$ respectively. So, $\triangle = (\triangle_L, \triangle_S, \triangle_O)$. And, we model them in a pairwise manner. Consider $\mathbf{c} = (C_i, C_j)$ $(C_i, C_j \in ch(A))$, we have,

$$X(\mathbf{c}|I^{(\mathbf{c})}) = \{X(C_i), X(C_j)\} = (L_{\mathbf{c}}, S_{\mathbf{c}}, O_{\mathbf{c}}|I^{(\mathbf{c})})$$

and

$$L_{\mathbf{c}} = (L_{C_i}, L_{C_j});\ S_{\mathbf{c}} = (S_{C_i}, S_{C_j});\ O_{\mathbf{c}} = (O_{C_i}, O_{C_j})$$

For the α processes of child nodes in $\mathbf{c}$, we have,

$$\begin{aligned}
&\log p(I^{(\mathbf{c})}|\mathbf{c}) = \log p(I^{(C_i,C_j)}|C_i, C_j) \\
&= \log p(I^{(C_i)}|C_i;\theta_{C_i}) + \log p(I^{(C_j)}|C_j;\theta_{C_j}) \\
&= \alpha(C_i;\theta_{C_i}) + \alpha(C_j;\theta_{C_j})
\end{aligned} \tag{35}$$

For binding child nodes in $\mathbf{c}$, we obtain,

$$\begin{aligned}
&p(X(\mathbf{c}|I^{(\mathbf{c})});\triangle) = P(X(C_i), X(C_j);\triangle) \\
&= p(L_{\mathbf{c}}, S_{\mathbf{c}}, O_{\mathbf{c}}|I^{(\mathbf{c})};\triangle) \\
&= p(L_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_L) \times p(S_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_S) \times p(O_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_O)
\end{aligned} \tag{36}$$

Solving $\beta(A|\mathbf{c};\phi^*)$ depends on choosing a specific modeling scheme for $p(L_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_L)$, $p(S_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_S)$ and $p(O_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_O)$. In this paper, the three terms are modeled as Gaussian distributions in their respective transformed spaces (Felzenszwalb and Huttenlocher, 2005).

Thus, each pairwise binding $\mathbf{c} = (C_i, C_j)$ is characterized by the expected relative location $\mu_{L_{\mathbf{c}}}$, scale $\mu_{S_{\mathbf{c}}}$ and orientation $\mu_{O_{\mathbf{c}}}$ and corresponding full covariance matrices $\Sigma_{L_{\mathbf{c}}}$, $\Sigma_{S_{\mathbf{c}}}$ and $\Sigma_{O_{\mathbf{c}}}$. So, we have $\triangle_L = (\mu_{L_{\mathbf{c}}}, \Sigma_{L_{\mathbf{c}}})$, $\triangle_S = (\mu_{S_{\mathbf{c}}}, \Sigma_{S_{\mathbf{c}}})$ and $\triangle_O = (\mu_{O_{\mathbf{c}}}, \Sigma_{O_{\mathbf{c}}})$ which can be estimated from the dataset $D_{\beta}^{+}(A|\mathbf{c})$. Then, we have,

$$\begin{aligned}
p(L_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_L) &= p(L_{C_i}, L_{C_j};\mu_{L_{\mathbf{c}}}, \Sigma_{L_{\mathbf{c}}}) \\
&= \mathcal{N}(L_{C_i} - L_{C_j};\mu_{L_{\mathbf{c}}}, \Sigma_{L_{\mathbf{c}}})
\end{aligned} \tag{37}$$

$$\begin{aligned}
p(S_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_S) &= p(S_{C_i}, L_{C_j};\mu_{S_{\mathbf{c}}}, \Sigma_{S_{\mathbf{c}}}) \\
&= \mathcal{N}(S_{C_i} - S_{C_j};\mu_{S_{\mathbf{c}}}, \Sigma_{S_{\mathbf{c}}})
\end{aligned} \tag{38}$$

and

$$\begin{aligned}
p(O_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_O) &= p(O_{C_i}, O_{C_j};\mu_{O_{\mathbf{c}}}, \Sigma_{O_{\mathbf{c}}}) \\
&= \mathcal{N}(O_{C_i} - O_{C_j};\mu_{O_{\mathbf{c}}}, \Sigma_{O_{\mathbf{c}}})
\end{aligned} \tag{39}$$

Further, we specify the three Gaussian distribution above in a transformed space to have zero means and diagonal covariances. To that end, we first compute the singular value decompositions of the three covariance matrices and then define the transformations. For example, for the location (the same is for the scale and orientation), we have ,

$$\Sigma_{L_{\mathbf{c}}} = U_{L_{\mathbf{c}}} D_{L_{\mathbf{c}}} U_{L_{\mathbf{c}}}^T \tag{40}$$

$$T_{ij}(L_{C_i}) = U_{L_{\mathbf{c}}}^T (L_{C_i} - \mu_{L_{\mathbf{c}}}) \tag{41}$$

$$T_{ji}(L_{C_j}) = U_{L_{\mathbf{c}}}^T (L_{C_j}) \tag{42}$$

So, we can rewrite Eqn.37 as

$$\begin{aligned}
&p(L_{\mathbf{c}}|I^{(\mathbf{c})};\triangle_L) = \mathcal{N}(L_{C_i} - L_{C_j};\mu_{L_{\mathbf{c}}}, \Sigma_{L_{\mathbf{c}}}) \\
&= \mathcal{N}(T_{ij}(L_{C_i}) - T_{ji}(L_{C_j});0, D_{L_{\mathbf{c}}})
\end{aligned} \tag{43}$$

So, we calculate the binding score of nodes in $\mathbf{c}$ as,

$$w_{\mathbf{c}}^{\text{bind}} = \log p(X(\mathbf{c}|I^{(\mathbf{c})}); \triangle) \tag{44}$$

Then, by combining Eqn.36 and Eqn.44, we obtain the weight of a β hypothesis of node A given the children $\mathbf{c}$,

$$\begin{aligned} w_A^{\beta(\mathbf{c})} &= \log p(I^{(\mathbf{c})}, X(\mathbf{c}|I^{(\mathbf{c})})|\mathbf{c}; \phi) \\ &= \log p(I^{(\mathbf{c})}|\mathbf{c}) + \log p(X(\mathbf{c}|I^{(\mathbf{c})}); \triangle) \\ &= w_{\mathbf{c}}^{\text{bind}} + \sum_{C_i \in \mathbf{c}} w_{C_i}^{\alpha} \end{aligned} \tag{45}$$

The threshold $\text{Th}(w_A^{\beta(\mathbf{c})})$ of the β process $\beta(A|\mathbf{c}; \phi)$ can be estimated in a validation β dataset.

4.5 Learning the γ process

Learning the γ process involves specifying $\gamma(A|P, \varphi)$ and estimating the parameters φ by maximizing the data likelihood on $D_\gamma^+(A|P)$. $\gamma(A|P; \varphi)$ predicts hypothesis of node A from the α process of its parent node P. In the literature, context-based methods (Torralba, 2003; Hoiem et al, 2008) can be looked as γ processes. Given $D_\gamma^+(A|P)$, we have,

$$\begin{aligned} &\gamma(A|P; \varphi^*) \\ &= \arg\max_{\varphi} p(D_\gamma^+(A|P)|P; \varphi) \\ &= \arg\max_{\varphi} \sum_{i=1}^{m} \log p(I_i^{(P)}, X(A|I_i^{(P)})|P; \varphi) \end{aligned} \tag{46}$$

Also, $\gamma(A|P; \varphi)$ consists of two components, one is the α process of the parent node P with the parameters θ_P and the other is the predicting model from parent node P to node A itself with parameters $\bigtriangledown$. So, we have $\varphi = (\theta_P, \bigtriangledown)$ and obtain,

$$\begin{aligned} &p(I^{(P)}, X(A|I^{(P)})|P; \varphi) \\ &= p(I^{(P)}|P; \theta_P) \times p(X(A|I^{(P)}); \bigtriangledown) \end{aligned} \tag{47}$$

where we also use $I^{(P)}$ to represent $I_i^{(P)}$ in general.

In the γ process, we want to predict the location L_A, scale S_A and orientation O_A of node A from the parent node P. So, we have $\bigtriangledown = (\bigtriangledown_L, \bigtriangledown_S, \bigtriangledown_O)$ and

$$X(A|I^{(P)}) = X(A)|X(P) = (L_A, S_A, O_A|I^{(P)})$$

For the α process of the parent node P, we have,

$$\log p(I^{(P)}|P; \theta_P) = \alpha(P; \theta_P) \tag{48}$$

In order to predict a hypothesis of node A, we have,

$$\begin{aligned} &p(X(A|I^{(P)}); \bigtriangledown) = p(X(A)|X(P); \bigtriangledown) \\ &= p(L_A, S_A, O_A|I^{(P)}; \bigtriangledown) \\ &= p(L_A|I^{(P)}; \bigtriangledown_L) \times p(S_A|I^{(P)}; \bigtriangledown_S) \times p(O_A|I^{(P)}; \bigtriangledown_O) \end{aligned} \tag{49}$$

Then, solving $\gamma(A|P, \varphi^*)$ depends on how we model $p(L_A|I^{(P)}; \bigtriangledown_L)$, $p(S_A|I^{(P)}; \bigtriangledown_S)$ and $p(O_A|I^{(P)}; \bigtriangledown_O)$. The three terms are also treated as Gaussian distribution. So, $\bigtriangledown_L = (\mu_{L_A}, \Sigma_{L_A})$, $\bigtriangledown_S = (\mu_{S_A}, \Sigma_{S_A})$ and $\bigtriangledown_O = (\mu_{O_A}, \Sigma_{O_A})$. For example, we have,

$$\begin{aligned} &p(L_A|I^{(P)}; \bigtriangledown_L) \\ &= p(L_A|I^{(P)}; \mu_{L_A}, \Sigma_{L_A}) \\ &= \mathcal{N}(L_A; \mu_{L_A}, \Sigma_{L_A}) \end{aligned} \tag{50}$$

where μ_{L_A} is the mean and Σ_{L_A} is the covariance, estimated by the statistics in $D_\gamma^+(A|P)$.

So, we compute the prediction score for node A from its parent node P as,

$$w_P^{\text{predict}} = \log p(X(A)|X(P); \bigtriangledown) \tag{51}$$

Then, the weight of a γ hypothesis of node A is,

$$w_A^{\gamma(P)} = w_P^{\text{predict}} + w_P^{\alpha} \tag{52}$$

Similarly, the threshold $\text{Th}(w_A^{\gamma(P)})$ of the γ process $\gamma(A|P; \varphi)$ can be estimated in a validation γ dataset.

The bottom panel of Fig.8 illustrates the learned Gaussian distributions in the β binding and γ prediction process for human face.

5 Evaluating the information contributions of the α, β and γ processes

We propose a method to evaluate the information contributions of the α, β and γ processes individually based on their discriminative power. Our method is similar to the decision tree framework (Breiman et al, 1984). We also study human performance for the three processes individually for comparisons.

5.1 Evaluating method

For simplicity of notation, we denote the α, β and γ processes as a testing function Tst(). As illustrated in Fig.10, the *information contribution* of Tst(), denoted by IC(Tst), is measured by the *uncertainty or impurity reduction* after applying it on a testing dataset D.

The testing dataset $D = D^+ \cup D^-$ includes a set of positive samples D^+ and a set of negative samples D^-. After applying Tst(), we can obtain two datasets, one is

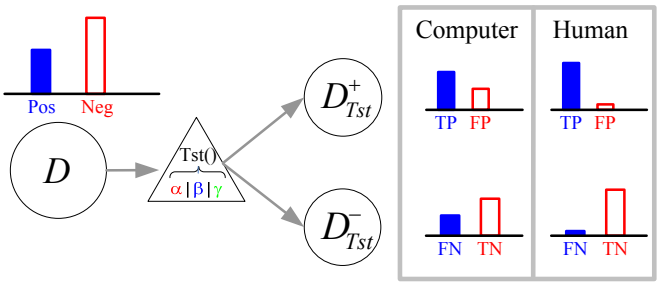

Fig. 10 Illustration of evaluating the information contributions of the α, β and γ processes individually based on their discriminative power. We also study the human performance for the three processes individually for comparisons. $D = D^+ \cup D^-$ are the input testing dataset including a positive sample set D^+ and a negative sample set D^-. After testing, we obtain two subsets, one is the set D^+_{Tst} in which samples pass the test and the other D^-_{Tst} in which samples do not pass the test. D^+_{Tst} consists of TPs and FPs, and D^-_{Tst} includes TNs and FNs. See texts for the calculation of information contribution.

D^+_{Tst} in which samples pass the testing function Tst() and the other is D^-_{Tst} in which samples fail. So, D^+_{Tst} consist of true positives (TPs) and false positives (FPs), and D^-_{Tst} include true negatives (TNs) and false negatives (FNs). We compute the information contribution of Tst() as ,

$$\text{IC}(\text{ Tst}) = 1 - \frac{\mathcal{H}(D^+_{Tst}) + \mathcal{H}(D^-_{Tst})}{\mathcal{H}(D)} \tag{53}$$

where $\mathcal{H}()$ represents the impurity of a dataset which is defined as the product of the size of the dataset (denoted by $|\cdot|$) and its entropy (denoted by $\mathbb{E}()$),

$$\mathcal{H}(D) = |D| \times \mathbb{E}(D) \tag{54}$$

and the entropy is

$$\mathbb{E}(D) = -\frac{|D^+|}{|D|}\log\frac{|D^+|}{|D|} - \frac{|D^-|}{|D|}\log\frac{|D^-|}{|D|} \tag{55}$$

In the same way, we can compute $\mathbb{E}(D^+_{Tst})$, $\mathcal{H}(D^+_{Tst})$, $\mathbb{E}(D^-_{Tst})$ and $\mathcal{H}(D^-_{Tst})$. Then, we calculate IC(Tst).

In the literature, an alternative approach for measuring Tst() is studied in (Blanchard and Geman, 2005) from some theoretical viewpoints.

5.2 Human study

The information contribution defined in Eqn.53 is empirical, so we also study human performance of the information contributions of the α, β and γ processes individually for comparisons.

The human study environment. Based on the psychological toolbox (Brainard, 1997), we develop a GUI interface for the human study. In experiments, we have 7 human subjects with normal sights. We use LCD monitors whose brightness and contrast are adjusted for each subject adaptively. The distance between human subjects and monitors are adjusted around 50cm according to each subject's sight. The outside light environment is also adjusted to a suitable level. In testing, clicking the enter key means the displayed sample is positive and clicking the space key means it is negative.

Observing time setting. In order to study the information contributions individually in the human study, in addition to control the scale of image patch, we control the observing time. For the α process, the observing time is less than 200ms. For the β and γ processes, we do not control the observing time. At same time, the response time of each subject is recorded.

The testing data for human subjects. In order to reduce the amount of human subjects' observing image data, we only use the FPs from computer experiments as the negative samples for human subjects. Fig.12 and Fig.14 show some examples used in evaluations of α and β processes of human face. The assumption is that those TNs would also be correctly classified by human subjects, which is intuitively reasonable. At the same time, each group of data is tested by all 7 subjects to eliminate possible biases made by some subjects. The human subjects can be treated as ideal observers and their overall performance improvement against the computer can be treated as a metric in future work for the computer vision community.

6 Experiments

In the experiments of our numerical study, we choose two hierarchical case studies, one is junctions and rectangles in low-to-middle-level vision and the other is human faces in high-level vision. And, we do two series of experiments, one is to evaluate the individual information contribution of the α, β and γ processes and the other is object parsing by integrating the three processes with performance comparisons.

6.1 Experiment I: evaluating information contributions of the α, β and γ processes individually

Junctions and rectangles. We consider five types of hierarchical image structures in low-to-middle-level vision including L-junction, cross junction, parallel line, T/Y/Arrow junction and rectangle. In our experiments, we treat T/Y/arrow junction as the same type currently due to the similarity. As illustrated in the right-bottom panel of Fig.9, the rectangle node is an Or-node which has two types of decompositions, one is decomposed

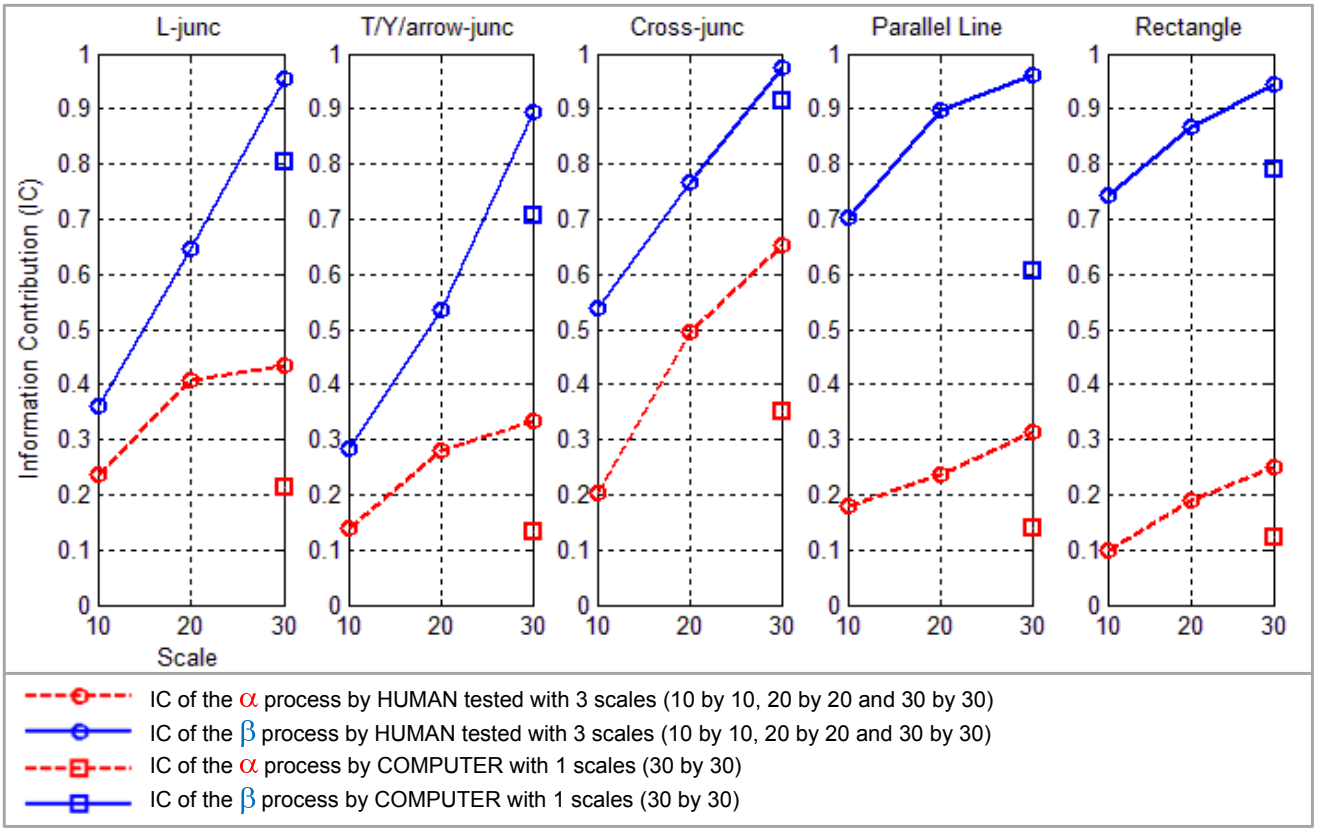

Fig. 11 The information contributions of the α and β processes of junctions and rectangles. We test three scales, 10×10, 20×20 and 30×30 pixels. We can observe that the β processes of junctions and rectangles are much better than their α processes.

into two groups of parallel lines and the other is decomposed into four junctions such as four L-junctions.

The data. A set of 200 natural images from the LHI image database (Yao et al, 2007) is used in which the sketches are manually labeled. Based on the manually labeled sketches, we extract positive examples for the five types of hierarchical image structures and a common set of negative examples. Some positive examples are shown in Fig.9.

Training and testing. For the α process, we use first and second derivative Gaussian filters, LoG (Laplacian of Gaussian) filters, DoG (difference of Gaussian) and elongated DooG (different of offset Gaussian) filters, all with 3 scales (10×10, 20×20 and 30×30 pixels). The α process of line segment uses the primal sketch model (Guo et al, 2007) similar to the implicit testing used in our previous compositional boosting work (Wu et al, 2007). The α processes of the five types of hierarchical image structures use the patch-based active basis model for both shape and texture. In testing, we search different 15 orientations in order to handle the rotation. For the β process, the five types of hierarchical image structures are computed by binding line segments in terms of the explicit testing on their relative locations, angles and distances between their endpoints. Rectangles are computed in two alternative ways, one is by binding two groups of parallel lines in terms of their relative locations and angles, and the other is by binding a set of incomplete (two or three) or complete (four) junctions in terms of their relative locations, angles and distances between the endpoints.

The observation: Fig.11 shows the information contributions of the α (red lines) and β (blue lines) processes of junctions and rectangles from the human study experiments at three scales (10×10, 20×20 and 30×30 pixels). The results of computer experiments are shown by those small rectangles (the red ones for the α and the blue ones for the β process and for clarity only the results tested with the scale 30×30 pixels are shown). We observe that the β binding inference process dominates in low-to-middle-level vision.

Human faces. We consider the AoG of the human face which consists of six nodes, head-shoulder, face, left eye, right eye, nose and mouth, as shown in Fig.8. In our experiments, we treat the left and right eye node as one same type of node due to the similarity.

The data. A set of 1000 images from the LHI database is used in which all the six nodes are at good resolution and the parse graphs are manually labeled (see an

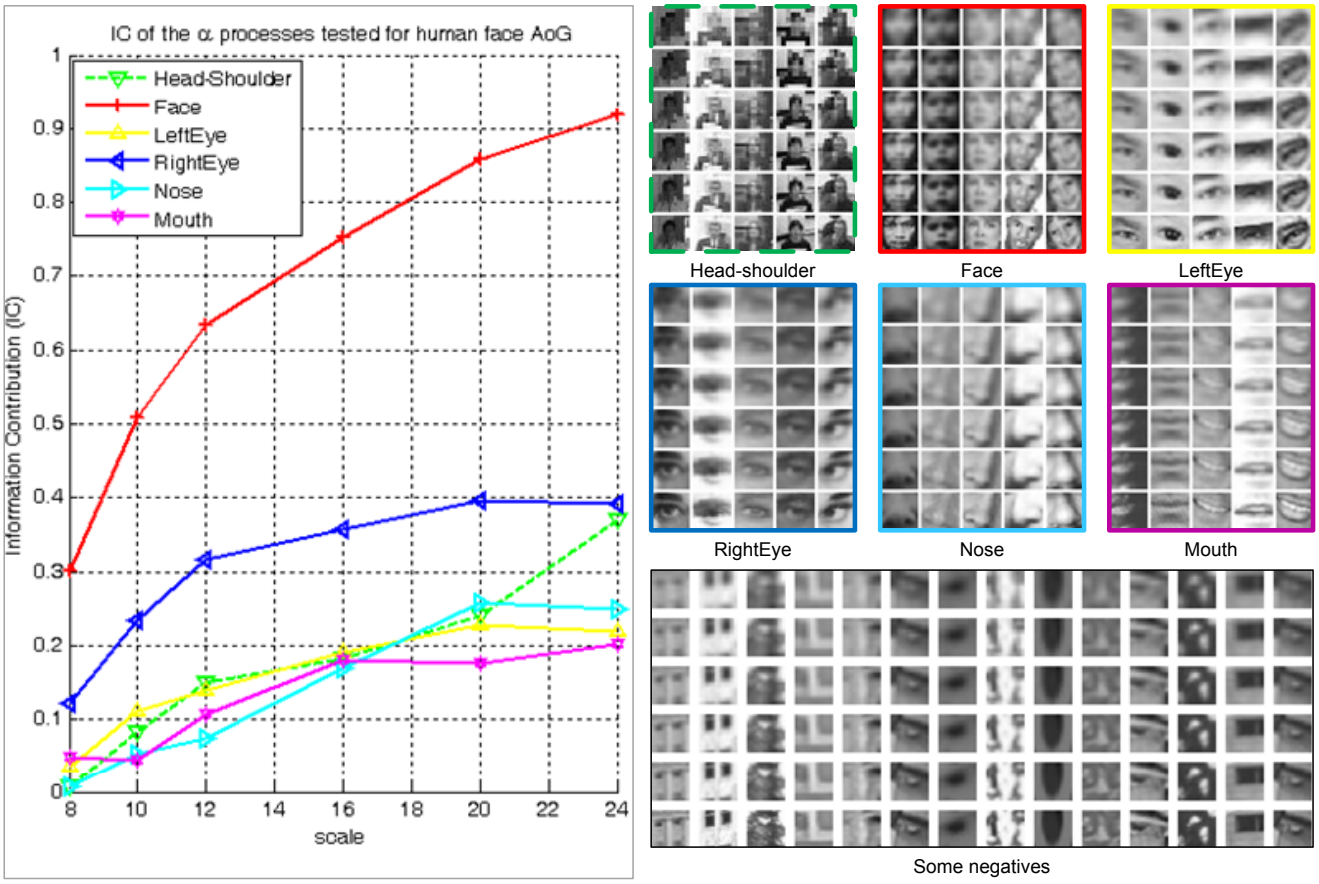

Fig. 12 The left panel shows the information contributions of the α processes of nodes (ie. head-shoulder, face, left eye, right eye, nose and mouth) in human face AoG in the human study. We test five scales (8×8, 10×10, 12×12, 16×16, 20×20 and 24×24 pixels). Some positive examples for each node and some negative examples are shown in the right panel. We can observe that the α(face) process is stronger than the α processes of other nodes in the human face AoG.

example in Fig.7). We generate the training data based on the parse graph.

Training and testing. For the α process, we use the Gabor filter. The learned α, β and γ processes are shown in Fig.8. Here, we test five scales for the α process (8×8, 10×10, 12×12, 16×16, 20×20 and 24×24 pixels), five scales for β process (38×38, 50×50, 60×60, 80×80 and 100×100) and one scale for the γ process of human face (32×32).

The observation: We observe that the α process of the human face node is stronger than those of the other node in the human face AoG. The information contributions are shown in Fig.12.

6.2 Experiment II: object parsing in a greedy pursuit manner by integrating the α, β and γ processes

Rectangles. We test a set of 50 images including 30 city scene images and 20 office scene images. A running example is shown in Fig.15 and more examples are shown in Fig.16. From the ROC comparisons in Fig.16, we can see that the β processes of junctions and rectangles in low-to-middle-level vision dominate with much performance improved against the α processes.

Human faces. We test a set of 500 images in which more than half of human face instances are with occlusion or at very low resolution. A running example of the human face pursuit is shown in Fig.5 and more examples are shown in Fig.17. From the ROC comparisons in Fig.17, we can see that for human face, its α process works better than those of its child nodes such as eyes and nose and its parent node such as head-shoulder.

The ROC comparisons are consistent with the evaluated information contributions in experiment I.

7 Summary and discussion

In this paper, we present a framework for the numerical study of the bottom-up and top-down inference processes in hierarchical models using AoG as an example and choose two hierarchical cases in our experiments,

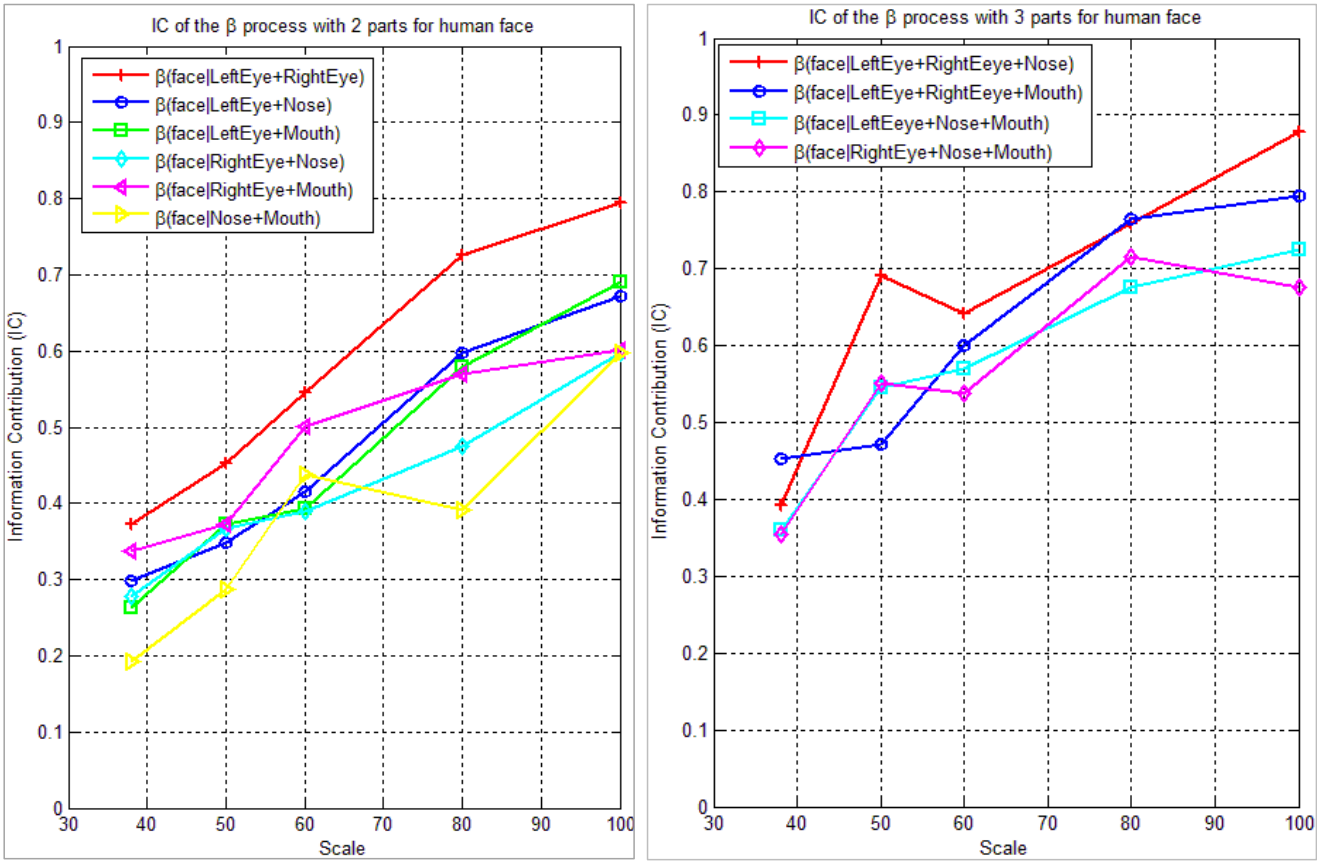

Fig. 13 The information contributions of the β processes of the human face. *The left panel* show the information contributions of the β processes with 2 facial components. *The right panel* is for the β processes with 3 facial components. We test five scales, 38×38, 50×50, 60×60, 80×80 and 100×100 pixels. Some examples are shown in Fig.14. In the β process, we can observe that the left eye and right eye are more informative than other facial components.

one is junctions and rectangles in low-to-middle-level vision and human faces in high-level vision. For each node A in an AoG, we identify three inference processes, termed the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes. The numerical study consists of four components: (i) isolating the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes based on a blocking method, (ii) training their models separately under the MLE framework, (iii) evaluating their information contributions individually based on their discriminative power in both computer experiments and human perception experiments and (iv) integrating them explicitly under the Bayesian framework for robust inference. Based on the numerical study in our experiments, we observe that:

(i) For each node A in an AoG, the $\alpha(A)$, $\beta(A)$ and $\gamma(A)$ processes contribute to compute it in complementary ways. Their effectivenesses depend on the scale and occlusion conditions.

(ii) In low-to-middle-level vision, for junctions (L, T, Y, arrow and cross junctions and parallel lines) and rectangles, their β processes (bottom-up binding processes in the hierarchy) are dominated based on both computer experiments and human perception experiments.

(iii) In high-level vision, the human face node have stronger α process than those of facial components.

(iv) For robust inference of object parsing using AoG, the three processes should be integrated explicitly under the Bayesian formulation. The integration takes advantage of the separation of the learning of the three processes.

Beside accuracy performance, computational efficiency is another important criteria in computer vision, especially when we have a big hierarchical model with 100s (even 1000s) nodes, we can not afford to perform bottom-up detections for all nodes at the beginning. On the other hand, some recent human vision experiments show that humans can recognize scene and object categories as fast as we detect the low level image primitives and the human visual system schedules the computing in an very effective way (Thorpe et al, 1996) (but how the human visual system handles that is still unclear to vision researchers). Actually, the scheduling problem is

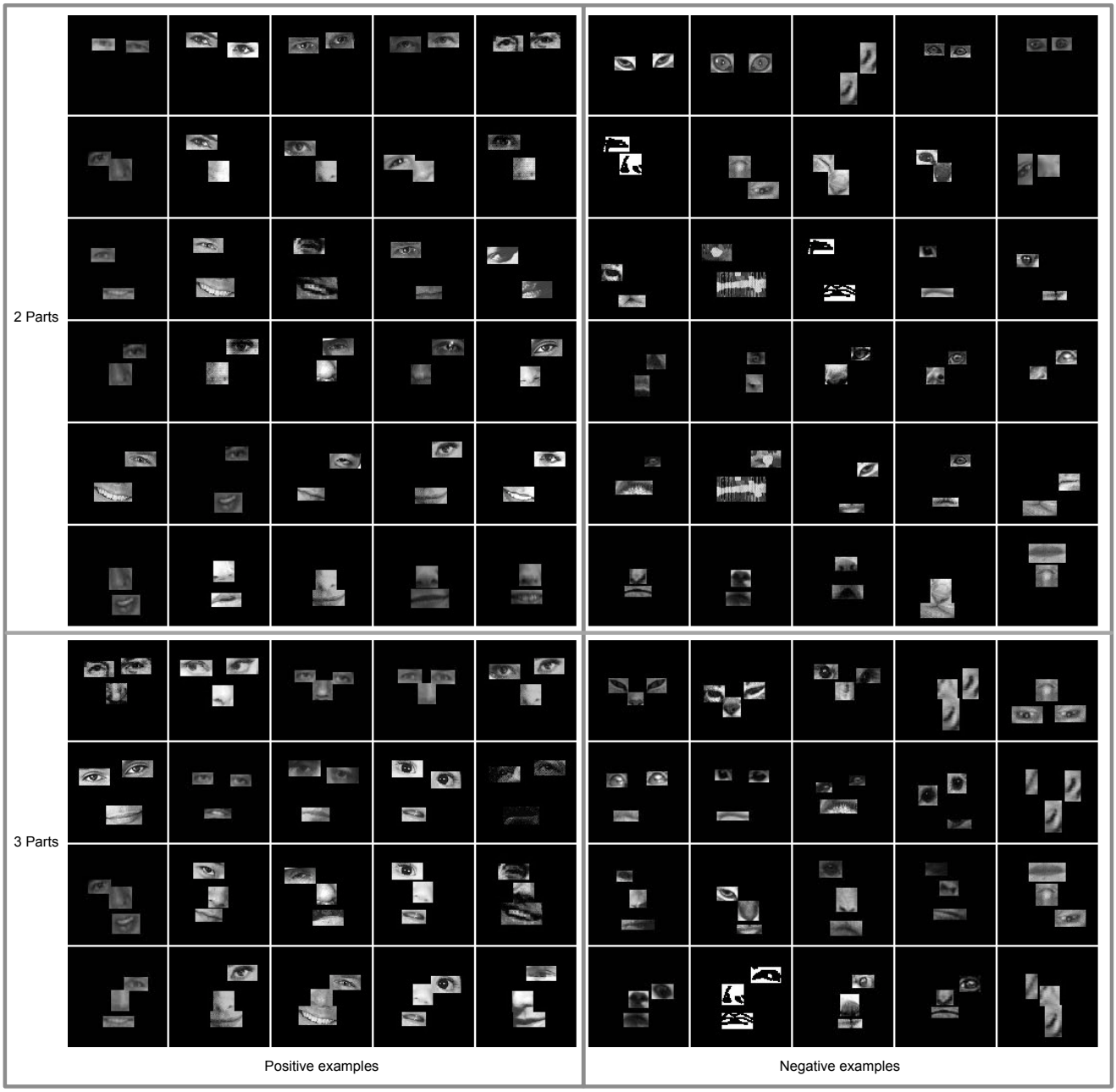

Fig. 14 Some examples used in the human study of evaluating the information contribution of the β processes of human face with 2 and 3 facial components respectively. The left panel shows some positive examples and the right panel shows some negative examples. The examples are at 100×100 pixels for illustration.

a long-standing problem in vision (Ullman, 1984) and often discussed verbally. We think that the answer lies in the numerical study of the bottom-up and top-down inference processes. We should evaluate their respective information contributions in the first place and then obtain some insights on how to schedule them. We leave the scheduling problem in our on-going work.

In the literature, some efficient search algorithms are proposed for computing a single node, such as the coarse-to-fine strategy (Blanchard and Geman, 2005; Fleuret and Geman, 2008), the efficient subwindow search method (Lampert et al, 2009), the dynamic programming methods (Meinshausen et al, 2009) and the A* heuristic algorithm (Felzenszwalb and McAllester, 2007; Kokkinos and Yuille, 2009). They do not handle the scheduling problem among different nodes in hierarchical models.

Consider object parsing using AoG as an example again, the objective of scheduling is to maximizing the accuracy performance and simultaneously minimiz-

Fig. 15 A running example of pursuing junctions and rectangles in a typical image by integrating the α, β and γ processes. The left-top is the original input image. The middle-top is the edge probability map and the right-top shows the detected corners. The images in the second and the third row show the detected results of different kinds of junctions with the type name shown in the left-top in each image. The left-bottom shows the detected rectangles. The middle-bottom shows the final sketch by merging all detected results. Compared with the canny results shown in the right-bottom image, we can see that the final sketch obtained by the proposal algorithm is better.

ing the overall computing cost. In hierarchical models, the computing always starts from some nodes's α processes in general. For computational efficiency, one should compute those most promising α processes first and then pass their messages to their child nodes through the top-down γ processes and to their parent nodes through the bottom-up β processes, and so on, schedule the bottom-up and top-down inference processes (the α, β and γ processes) in an AoG.

Acknowledgements This work at UCLA was supported by NSF grants IIS-0713652, ONR grant N00014-07-M-0287, DMS-0707055, and the work at LHI was supported by China 863 project 2008AA01Z126 and 2009AA01Z331, NSF China grants 60728203 and 60832004. The authors are thankful to the reviewers for their constructive comments, to Dr. Yingnian Wu for the active basis code and extensive discussions, to Xiong Yang and other artistic people at LHI for their helping us prepare the data and human study, to Brandon Rothrock, Haifeng Gong, Zhangzhang Si for their discussions.

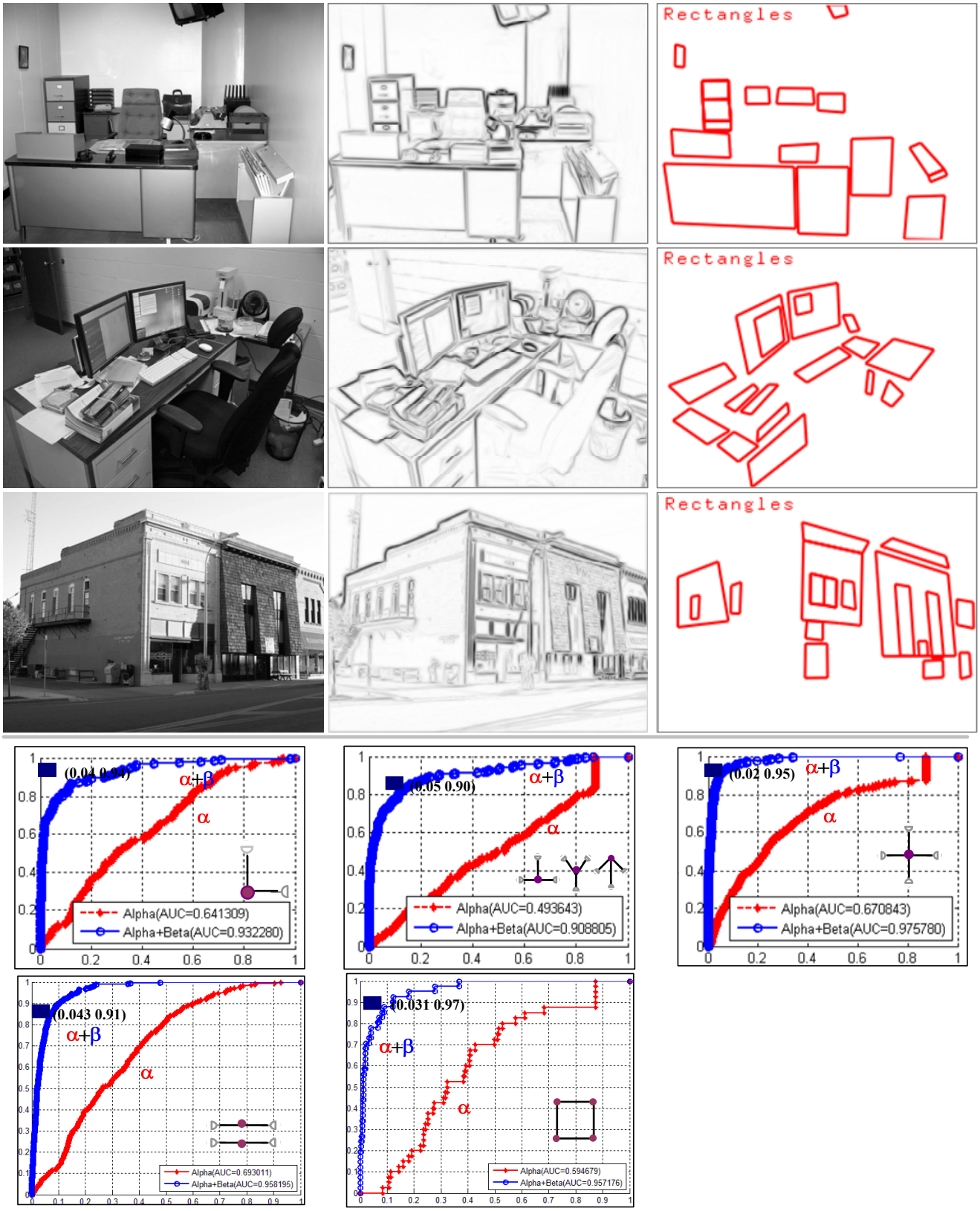

Fig. 16 *The top panel* show more results of rectangle pursuing. *The bottom panel* shows the ROC comparisons of the α process and the integration of the α, β processes for junctions and rectangle. Those small solid rectangles show the performance by humans.

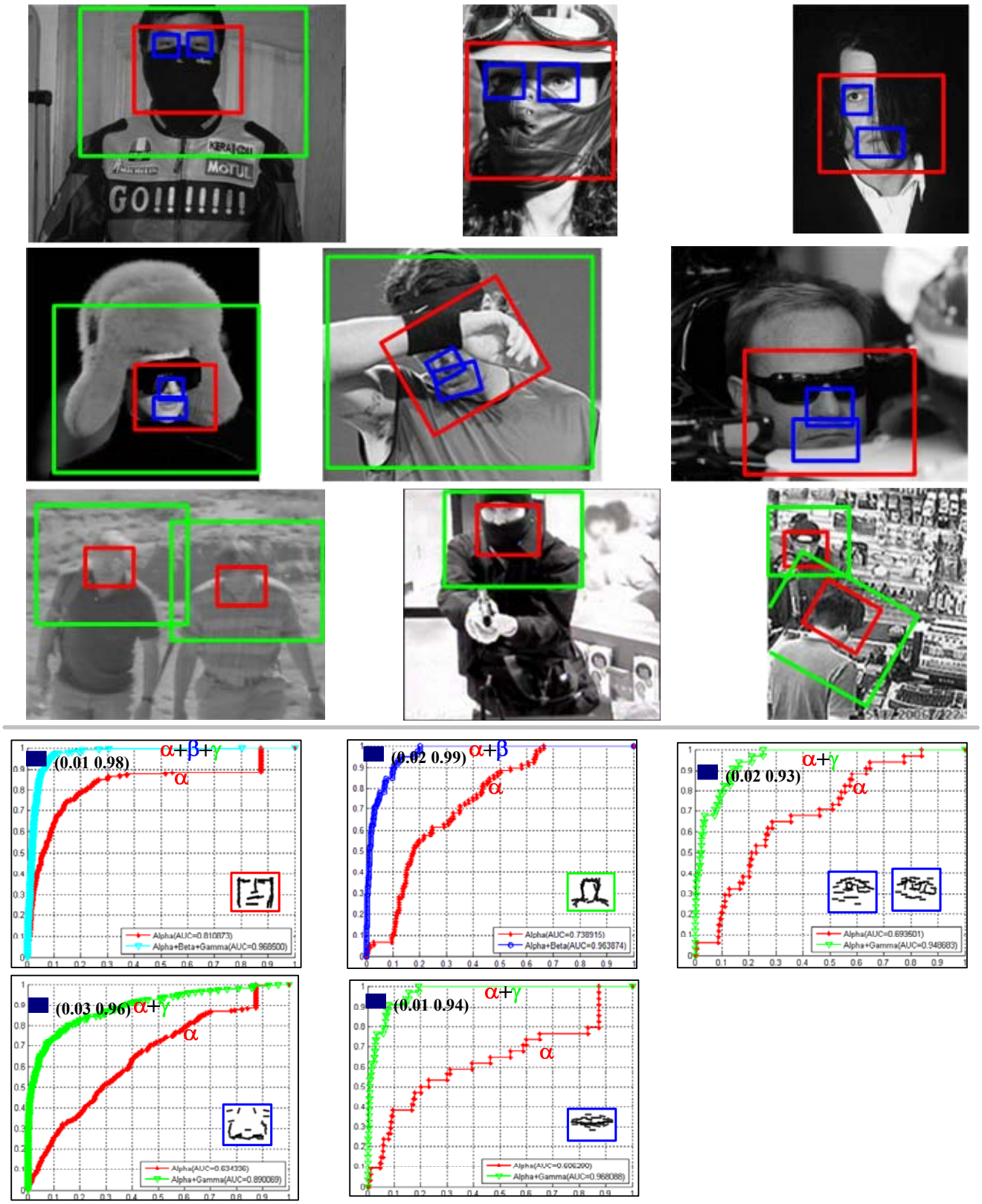

Fig. 17 *The top panel* shows more results of human face pursuing. *The bottom panel* shows the ROC comparisons of the α process and different integrations of the α, β and γ processes. Those small solid rectangles show the performance by humans.

References

Amit Y, Trouvé A (2007) Pop: Patchwork of parts models for object recognition. IJCV 75(2):267–282

Avidan S (2006) Spatialboost: Adding spatial reasoning to adaboost. In: ECCV, pp 386–396

Aycinena M, Kaelbling LP, Lozano-Perez T (2008) Learning grammatical models for object recognition. Tech. rep., MIT CSAIL

Biederman I (1987) Recognition-by-components: A theory of human image understanding. Psychological Review 94:115–147

Blanchard G, Geman D (2005) Hierarchical testing designs for pattern recognition. Ann Statist 33(3):1155–1202

Borenstein E, Ullman S (2008) Combined top-down/bottom-up segmentation. PAMI 30(12):2109–2125

Brainard DH (1997) The psychophysics toolbox. Spatial Vision 10:433–436

Breiman L, Friedman J, Stone CJ, Olshen R (1984) Classification and Regression Trees. Wadsworth and Brooks

Dechter R, Pearl J (1985) Generalized best-first search strategies and the optimality of a*. J ACM 32(3):505–536

Demirci MF, Shokoufandeh A, Keselman Y, Bretzner L, Dickinson S (2006) Object recognition as many-to-many feature matching. IJCV 69(2):203–222

Demirci MF, Platel B, Shokoufandeh A, Florack LL, Dickinson SJ (2009) The representation and matching of images using top points. J Math Imaging Vis 35(2):103–116

Divvala SK, Hoiem D, Hays JH, Efros AA, Hebert M (2009) An empirical study of context in object detection. In: CVPR

Epshtein B, Lifshitz I, Ullman S (2008) Image interpretation by a single bottom-up top-down cycle. PNAS 105(38):14,298–14,303

Fei-Fei L, Rob F, Pietro P (2006) One-shot learning of object categories. PAMI 28(4):594–611

Felzenszwalb P, Huttenlocher D (2005) Pictorial structures for object recognition. IJCV 61(1):55–79

Felzenszwalb P, McAllester D (2007) The generalized a* architecture. JAIR 29:153–190

Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2009) Object detection with discriminatively trained part based models. PAMI

Fergus R, Perona P, Zisserman A (2007) Weakly supervised scale-invariant learning of models for visual recognition. IJCV 71(3):273–303

Fidler S, Boben M, Leonardis A (2008) Similarity-based cross-layered hierarchical representation for object categorization. In: CVPR

Fink M, Perona P (2003) Mutual boosting for contextual inference. In: NIPS

Fleuret F, Geman D (2008) Stationary features and cat detection. JMLR 9:2549–2578

Geman S, Potter D, Chi ZY (2002) Composition systems. Quart Appl Math 60(4):707–736

Guo CE, Zhu SC, Wu YN (2007) Primal sketch: Integrating structure and texture. CVIU 106(1):5–19

Han F, Zhu SC (2009) Bottom-up/top-down image parsing with attribute grammar. PAMI 31(1):59–73

Heisele B, Serre T, Poggio T (2007) A component-based framework for face detection and identification. IJCV 74(2):167–181

Hoiem D, Efros A, Hebert M (2008) Putting objects in perspective. IJCV 80(1):3–15

Jin Y, Geman S (2006) Context and hierarchy in a probabilistic image model. In: CVPR, pp 2145–2152

Kokkinos I, Yuille A (2009) Hop: Hierarchical object parsing. In: CVPR

Lampert CH, Blaschko M, Hofmann T (2009) Efficient subwindow search: A branch and bound framework for object localization. PAMI Epub ahead:1–13

Lee TS, Mumford D (2003) Hierarchical bayesian inference in the visual cortex. J Opt Soc Am A Opt Image Sci Vis 20(7):1434–1448

Levin A, Weiss Y (2009) Learning to combine bottom-up and top-down segmentation. IJCV 81(1):105–118

Meinshausen N, Bickel P, Rice J (2009) Efficient blind search: Optimal power of detection under computational cost constraints. Ann Appl Stat 3(1):38–60

Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. Nature Neuroscience 2:1019–1025

Schneiderman H, Kanade T (2002) Object detection using the statistics of parts. IJCV

Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T (2007) Robust object recognition with cortex-like mechanisms. PAMI 29(3):411–426

Si Z, Gong H, Wu YN, Zhu SC (2009) Learning mixed templates for object recognition. CVPR pp 272–279

Sudderth EB, Torralba A, Freeman W, Willsky A (2008) Describing visual scenes using transformed objects and parts. IJCV 77(1-3):291–330

Thorpe S, Fize D, Marlot C (1996) Speed of processing in the human visual system. Nature 381:520–522

Todorovic S, Ahuja N (2008a) Region-based hierarchical image matching. IJCV 78(1):47–66

Todorovic S, Ahuja N (2008b) Unsupervised category modeling, recognition, and segmentation in images. PAMI 30(12):2158–2174

Torralba A (2003) Contextual priming for object detection. IJCV 53(2):169–191

Torralba A, Murphy K (2007) Sharing visual features for multiclass and multiview object detection. PAMI 29(5):854–869, senior Member-Freeman, William T.

Tu ZW, Zhu SC (2002) Image segmentation by data-driven markov chain monte carlo. PAMI 24(5):657–673

Tu ZW, Chen XR, Yuille A, Zhu SC (2005) Image parsing: Unifying segmentation, detection, and recognition. IJCV 63(2):113–140

Ullman S (1984) Visual routines. Cognition 18:97–159

Ullman S, Naquet MV, Sali E (2002) Visual features of intermediate complexity and their use in classification. Nature Neuroscience 5(7):682–687

Viola P, Jones M (2004) Robust real-time face detection. IJCV 57(2):137–154

Wu TF, Xia GS, Zhu SC (2007) Compositional boosting for computing hierarchical image structures. In: CVPR

Wu YN, Si ZZ, Gong HF, Zhu SC (2009) Learning active basis model for object detection and recognition. IJCV Epub ahead, DOI 10.1007/s11263-009-0287-0

Yao B, Yang X, Zhu SC (2007) Introduction to a large scale general purpose ground truth dataset: methodology, annotation tool, and benchmarks. EMMCVPR

Zhu SC, Mumford D (2006) A stochastic grammar of images. Found Trends Comput Graph Vis 2(4):259–362