Evaluating Information Contributions of Bottom-up and Top-down Processes

Xiong Yang[1,3]   Tianfu Wu[2,3]   Song-Chun Zhu[2,3]

[1]IPRAI, HUST, China, [2]Department of Statistics, UCLA and [3]Lotus Hill Research Institute (LHI), China

xyang@hust.edu.cn , {tfwu, sczhu}@stat.ucla.edu

Abstract

This paper presents a method to quantitatively evaluate information contributions of individual bottom-up and top-down computing processes in object recognition. Our objective is to start a discovery on how to schedule bottom-up and top-down processes. (1) We identify two bottom-up processes and one top-down process in hierarchical models, termed α, β and γ channels respectively ; (2) We formulate the three channels under an unified Bayesian framework; (3) We use a blocking control strategy to isolate the three channels to separately train them and individually measure their information contributions in typical recognition tasks; (4) Based on the evaluated results, we integrate the three channels to detect objects with performance improvements obtained. Our experiments are performed in both low-middle level tasks, such as detecting edges/bars and junctions, and high level tasks, such as detecting human faces and cars, together with a group of human study designed to compare computer and human perception.

1. Introduction

In object detection and recognition, hierarchical models and contextual information are widely used [20, 5, 8], and there are two types of computing processes for them: bottom-up and top-down processes [20, 12, 11, 1]. The main objective of this paper is to numerically evaluate information contributions for individual bottom-up process and top-down process and start to discover how to schedule them in different vision tasks. We identify two bottom-up processes and one top-down process in hierarchical model, and propose to (1) separately train their models through a blocking strategy, (2) quantitatively evaluate their individual information contributions in typical recognition tasks, and (3) integrate them for performance improvements.

In the literature, bottom-up and top-down processes are studied with three main kinds of viewpoints: (i) pure bottom-up feed-forward computing, such as works from Poggio's group at MIT [5] and many detection methods such as AdaBoost [14], (ii) pure top-down computing, such as template matching [18], and (iii) one pass of bottom-up followed by a top-down phase in a separate manner, such as DDMCMC [11], compositional boosting algorithm [15], compositional detection method [19] and recent active basis model [16]. The pure bottom-up methods are fast but often local and ambiguous, and the pure top-down methods can be global but too slow in searching solution space. Optimally combining them is a desired way and remains a long-standing problem in vision [12]. Recent cognitive and neuroscience experiments do show that in human perception bottom-up and top-down seem to operate in a complex interactive way [9]. But in order to mimic that with computer algorithm, we think that one key step is to numerically evaluate how much information individual bottom-up process and the top-down process would contribute for various objects and datasets, respectively.

This paper studies bottom-up and top-down computing processes in the hierarchical And-Or graph (AoG) representation[20] as an example, and we hope the results obtained here can be generalized for other types of (hierarchical) representations. An AoG can be represented by a graph $G = <V, E>$ where V represents a set of nodes and E a set of edges linking the nodes. A portion of the AoG of human faces is illustrated in the right panel of Fig.1 (a). Let A be a node in V representing the object-of-interest such as face, eye, head-shoulder, etc. In Fig.1 (a), we study the human face node (A = human faces) and consider how to detect faces in the image shown in the left panel of the figure. As illustrated by the arrows in three colors in the figure, we can define three computing ways for node A:

Definition 1: (the α-channel). node A can be detected alone without taking advantage of surrounding contexts while its children or parts are not recognizable alone in cropped patches. Those faces in Fig.1 (b) can be detected in this channel.

Definition 2: (the β-channel). some of node A's child nodes can be recognized in cropped patches while node A itself (also without surrounding contexts included) is not recognizable alone mainly due to occlusions. We can detect those faces in Fig.1 (c) by binding their parts such as eyes and/or nose whose α channels are on.
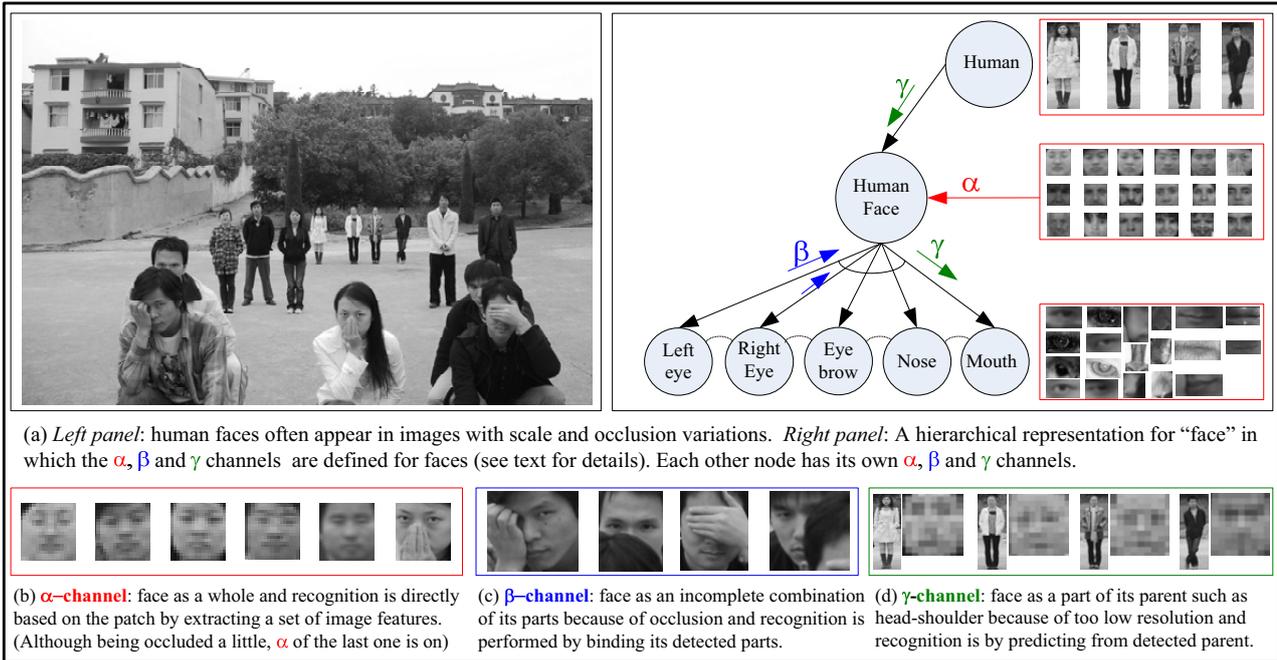
(a) *Left panel*: human faces often appear in images with scale and occlusion variations. *Right panel*: A hierarchical representation for "face" in which the α, β and γ channels are defined for faces (see text for details). Each other node has its own α, β and γ channels.

(b) α–channel: face as a whole and recognition is directly based on the patch by extracting a set of image features. (Although being occluded a little, α of the last one is on)

(c) β–channel: face as an incomplete combination of its parts because of occlusion and recognition is performed by binding its detected parts.

(d) γ-channel: face as a part of its parent such as head-shoulder because of too low resolution and recognition is by predicting from detected parent.

Figure 1. Illustration of the α, β and γ computing channels. In (a), the left panel is a typical image for human faces, and the right panel is a portion of the And-Or graph [20] for human faces. For the human face node, three computing channels can be defined as illustrated by the arrows in red, blue and green colors respectively. The three computing channels handle different types of image data, such as the human face as a whole in (b) for the α-channel, as an incomplete combination of parts in (c) for the the β-channel and as a part of its parent in (d) (see texts for their definitions and details).

Definition 3: (the γ-channel). node A can not be recognized alone in isolation, so does its parts, mainly due to too low resolution, but it can be recognized through its detected parent nodes. Here, we let the parent node pass the context information, such as information from some sibling nodes. Those faces in Fig.1 (d) can be predicted from some parents such as the head-shoulder whose α-channel was on.

The α and β channels are bottom-up processes and the γ-channel is a top-down process. They all contribute to the detection of node A, for example, face detection in Fig.1 (a). In the hierarchical AoG, each node has its own α, β and γ channels, and the definitions of the three channels are recursive. In computing, each channel has two states, "on" or "off". The root node's γ channel and all leaf node's β channels are always off. Intuitively, the three processes can be formulated as three kinds of conditional probabilities modeling different types of image data as the faces shown in Fig.1 (b), (c) and (d).

The motivations of this paper are in three-fold: (1) it would be computationally impossible to apply the three channels of all the nodes in an AoG in detection tasks with the sliding window technique, especially when the AoG was big, so it entails scheduling orders among them, (2) different object categories in different (task-dependent) datasets would have different computing orders which must be eval-

uated before the testing stage in order to improve performance and speed up computing, and (3) we hope the method presented here can be applied to various object categories and datasets to explore information contributions therein.

For an overview, some results obtained in this paper are:

(1) *Strong β-channel in low-middle vision tasks*, we test the α and β channels for detecting five generic image patterns, say, flat/homogenous region, edge/bar, L-junction, T/Y/arrow-junction and cross-junction, and the evaluated information contributions show that the β-channel contributes much more to detect them than the α-channel as shown in Fig.4.

(2) *In high level vision tasks, human faces have strong α-channels while cars seem to have strong β-channels mainly because of occlusions of car in street and the strong α channel of the wheel.* The results are shown in Fig.6 and Fig.7. This may help us explain that why the AdaBoost method in [14] works very well on frontal faces but not yet as effective on other objects such as cars.

(3) *The three channels contribute differently under different tasks due to scale changes and occlusions.* We evaluate their information contributions at multiple scales and under different kinds occlusions as the results shown in Fig.4, Fig.6 and Fig.7.

(4) *Performance improvements are obtained by integrat-*

ing the three computing channels. Because objects can appear in images at different scales and with different degree of occlusions, this entails the integration of the three channels for better recognition rates. Some improvements obtained in our experiments shown in Fig.4, Fig.6 and Fig.7.

2. Evaluating the α, β and γ channels

2.1. Isolating the α, β, γ channels

In order to measure their information contributions individually, we need to separately train the three channels through isolating the three channels of node A. We do that through *scaling* and *masking* the image patches with respect to node A:

(1) *Isolating the α-channel.* The γ-channel is turned off by *cropping* image patches of node A out of its context, as the face patches shown in Fig.1 (b), and the image patches are *down-sampled* up to some suitable scales at which the parts, if cropped in isolation, can not be recognized, thus the β-channel be blocked.

(2) *Isolating the β-channel.* The γ-channel is blocked through *cropping* the compact image patches of node A. The α-channel is blocked by adding suitable *occlusions* onto some parts, as the face patches shown in Fig.1 (c). But some parts are recognizable alone, if cropped.

(3) *Isolating the γ-channel.* For this aim, it is to only remain the contexts of node A in its image patches, as the face patches shown in Fig.1 (d).

2.2. Measuring the information contributions

In general, we denote the α, β and γ channels as testing functions, $T()$, to define their information contributions. Let $D^+(A)$ be a set of positive images of node A and $D^-(A)$ a set of negative images. Based on the isolating strategy stated above, we can separately generate training and testing datasets for the three channels from $D^+(A)$, say, $D_\alpha^+(A)$, $D_\beta^+(A)$ and $D_\gamma^+(A)$. The training and inference algorithm is addressed in Sec.3. Here, we present the method to measure the information contributions.

Definition 4: (Information Contributions (IC)). The information contribution of testing function $T()$ is measured by the *uncertainty reduction* after applying $T()$ in its testing datasets D^+ and D^-. We denote D_T^+ and D_T^- as the result datasets after testing.

Let $q(D^+ \cup D^-)$, $q_T^+(D_T^+)$ and $q_T^-(D_T^-)$ be the *population distribution* of positive and negative samples in the original dataset and the two datasets after testing, respectively. Therefore, their entropies can be calculated, denoted as $\mathbb{H}(q)$, $\mathbb{H}(q_T^+)$ and $\mathbb{H}(q_T^-)$.

The *uncertainty* is defined as the product of the population size and the entropy, so the *information contribution* is

measured by,

$$
\begin{aligned}
IC(T) = & (|D^+| + |D^-|) \times \mathbb{H}(q) \\
& - |D_T^+| \times \mathbb{H}(q_T^+) - |D_T^-| \times \mathbb{H}(q_T^-) \quad (1)
\end{aligned}
$$

The information contribution defined in Eqn.1 is empirical. In our experiments, we use this measurement in both computer experiments and human study to compare computer and human perception. In the literature, an alternative approach from some theoretical viewpoints for measuring $T()$ is studied in [1].

3. Implementing the α, β and γ channels

3.1. Problem formulation

Given a node A in the AoG, it can be represented by a graph $G_A = \langle V_A, E_A \rangle$, where $V_A = V_A^P \cup \{A\} \cup V_A^C$ correspond to the parent node(s), A itself and the child nodes, and $E_A = E_A^\updownarrow \cup E_A^\leftrightarrow$ represent vertical decomposition edges and horizontal spatial relation edges. More details about the AoG is referred to [20].

Given an input image I, the task is to detect all the instances of each node in V_A and output their parse graphs pg_A by maximizing a posterior probability under the Bayesian framework,

$$
pg_A^* = \arg\max_{pg_A} p(pg_A|I) = \arg\max_{pg_A} p(I|pg_A)p(pg_A)
\tag{2}
$$

where $p(I|pg_A)$ is the likelihood model and $p(pg_A)$ the prior. The likelihood model is based on the primal sketch model [20] and the prior can be modeled by stochastic context free grammar [2, 20];

Above, as all information in I about node A can be divided into the α, β and γ channels, I would be equivalently represented by their corresponding proposal maps, denoted as $\alpha(A, I)$, $\beta(A, I)$ and $\gamma(A, I)$, respectively. Then, we would instead maximize,

$$
\begin{aligned}
pg_A^* &= \arg\max_{pg_A} p(I|pg_A)p(pg_A) \\
&= \arg\max_{pg_A} p(\alpha(A,I), \beta(A,I), \gamma(A,I)|pg_A)p(pg_A) \quad (3)
\end{aligned}
$$

where $\alpha(A, I)$, $\beta(A, I)$ and $\gamma(A, I)$ are independent given pg_A, so $p(\alpha(A,I), \beta(A,I), \gamma(A,I)|pg_A)$ can be factorized into $p(\alpha(A,I)|pg_A)p(\beta(A,I)|pg_A)p(\gamma(A,I)|pg_A)$.

Next, we first separately train the α, β and γ channels, then evaluate their information contributions as stated in Sec.2. Based on the training and evaluating results, we can *integrate* the three channels in Eqn.3 to do inference for performance improvements.

3.2. Training the individual α, β and γ channels

The training data preparation is as stated as in Sec.2.1 and Sec.2.2. For node A, we have three types of positive datasets, $D_\alpha^+(A)$, $D_\beta^+(A)$ and $D_\gamma^+(A)$ for training the α, β and γ channels separately. Negative images $D^-(A)$ are collected hugely as a common dataset. The same process can be prepared for any node $v \in V_A^P \cup V_A^C$.

3.2.1 Training the α channel

For each image patch $I_\alpha \in D_\alpha^+(A)$, its β and γ channels are blocked, so $\beta(A, I_\alpha) = \Phi$ and $\gamma(A, I_\alpha) = \Phi$. Then, we have $pg_A = \{A\}$ and $\alpha(A, I_\alpha) = F(I_\alpha)$ where $F(I_\alpha)$ could be a set of image features such as responses of a filter bank [14] or Gabor wavelets [16] directly computed on the image patch I_α.

$$pg_A^* = \{A\}^* = \arg\max_{pg_A} p(pg_A | \alpha(A, I_\alpha))$$

$$\approx \arg\max \log \frac{p(A|F(I_\alpha))}{p(\overline{A}|F(I_\alpha))} = \arg\max \log \frac{p(F(I_\alpha)|A)p(A)}{p(F(I_\alpha)|\overline{A})p(\overline{A})} \quad (4)$$

where $\overline{A}$ represents competitive hypotheses and $\frac{p(A)}{p(\overline{A})}$ is often treated as a constant. That is what the AdaBoost method [14] and the active basis model [16] have done.

The α channel of each node in V_A is trained and evaluated. Given a observed image I^{obs}, for each $v \in V_A$, we could generated its proposal map $\alpha(v, I^{obs})$.

Next, before training the β and γ channel for node A, we first train the α channels of its parent node(s) $v^P \in V_A^P$ and its child nodes $v^C \in V_A^C$.

3.2.2 Training the β channel

For each image patch $I_\beta \in D_\beta^+(A)$, its α and γ channels are blocked, so $\alpha(A, I_\beta) = \Phi$ and $\gamma(A, I_\beta) = \Phi$. For any $v \in V_A^C$, we can compute the proposal map $\alpha(v, I_\beta)$, training the β channel of node A is to find the meaningful alignments of its child nodes captured by deformable part models [2, 16, 18, 20]. Here, we have $\beta(A, I_\beta) = <e_{u,v} \in E_A^\updownarrow, u, v \in V_A^C>$ and $pg_A = <v \in V_A^C>$, so,

$$pg_A^* = <v \in V_A^C>^* = \arg\max_{pg_A} p(pg_A | \beta(A, I_\beta))$$

$$= \arg\max_{pg_A} p(E_A^\updownarrow | V_A^C) p(V_A^C)$$

$$= \arg\max_{pg_A} \prod_{e_{u,v}} p(e_{u,v} | \alpha(u, I_\beta), \alpha(v, I_\beta)) \prod_{t \in V_A^C} p(\alpha(t, I_\beta)) \quad (5)$$

Here we can train different $\beta(A, I_\beta)$ in terms of the dataset $D_\beta^+(A)$ in which the α channels of different $v \in V_A^C$ turn on and others are blocked. Then we evaluate these different compositions of child nodes.

（a）two-layer hierarchical model
（b）Some examples
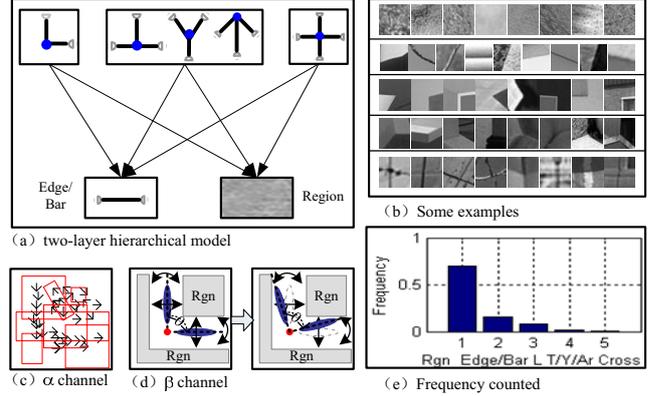（c）α channel
（d）β channel
（e）Frequency counted

Figure 2. (a) is a two layer And-Or graph representation for flat region, edge/bar, L-, T/Y- and cross junction. (b) shows some positive examples and (e) shows their frequencies counted in training dataset. Here, flat regions and edges/bars are treated as leaf nodes, so a L-junction is composed by two edges and two regions, and so on. (c) and (d) illustrate the α and β channels. The α-channel is based on statistics of response of a filter bank and the β-channel uses explicit binding model as in [15].

Region
Edge/Bar
L-junction
T/Y/arrow-junc
Cross-junction
Junctions Map

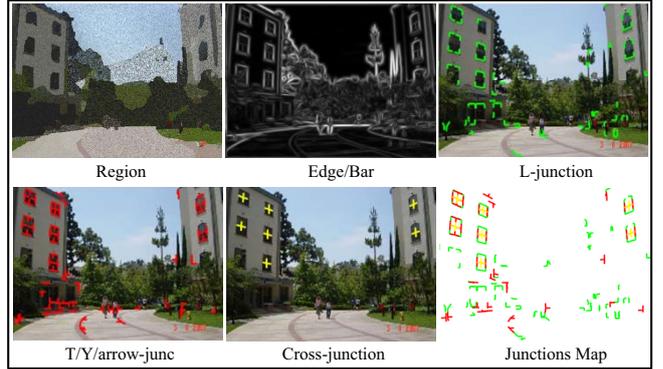Figure 3. One example of detection results of the five low-middle level primitives in our experiments.

3.2.3 Training the γ channel

The γ channel can be trained in the same way as training the β channel of the parent node(s) of node A using the dataset $D_\gamma^+(A)$ and $D^-(A)$ due to the recursive definition of the three channels.

4. Experiments

4.1. Five primitives at low-middle level vision

Regions, including flat region, texture region, etc., and sketches, including edgelet, bar, different kinds of junctions, etc., are two complementary kinds of groups of primitives in representing image at low-middle level vision [15] or as building blocks for high-level objects [16]. In our first experiment, we evaluate five primitives: flat/homogenous
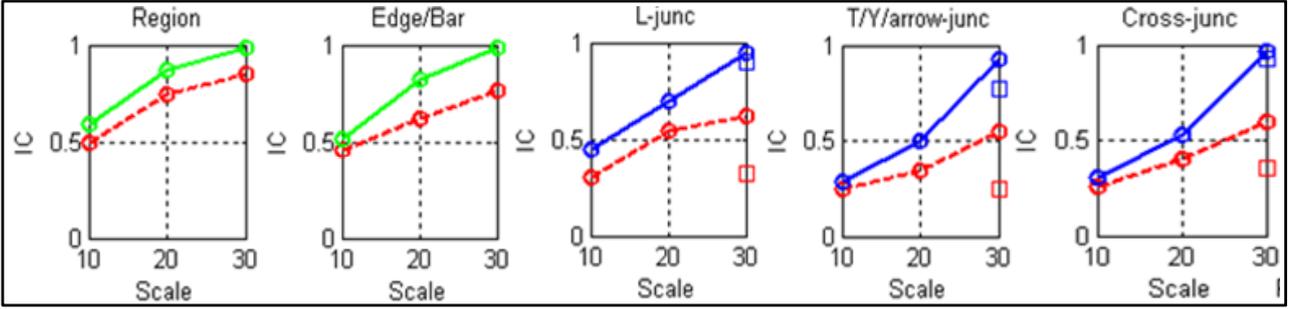
(a) Information contributions (IC) evaluated for the five low-level elements.
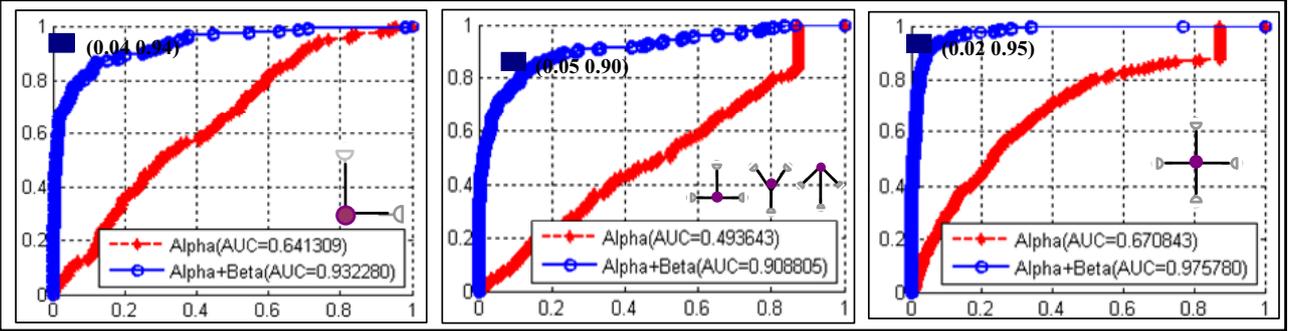
(b) ROCs of α(in red) and $\alpha+\beta$(in blue) for L, T/Y/arrow and cross junctions respectively

Figure 4. The information contributions and testing ROCs. (a) The evaluation of information gain are performed under three scales 10×10, 20×20 and 30×30. The red lines are for α-channels, blue for $\alpha+\beta$-channels and green for $\alpha+\gamma$ channels. The square points represent the evaluated results of computer algorithm and circle points for results obtained in human perception. (b) We also plot the testing ROCs for the three kinds of junctions with red curves for α-channel and blue for $\alpha+\beta$ channel. The performance of human on ROCs is point plotted as square points in the figure.

texture region, edge/bar, L-junction, T/Y/arrow-junction and cross-junction as shown in Fig.2 (b).

The data and the frequency. A set of 200 natural images from LHI image database [17] is used in which the sketches and regions are manually labeled, and randomly divided into two equal subsets as training and testing datasets. Based on the labeled perfect sketches and region segmentations, the frequency of the five nodes are counted in training dataset using the method in [6, 20], as shown in Fig.2 (e). The order of the frequency are very intuitive.

Training and testing. We model the five elements with the AoG representation as shown in Fig.2 (a). We learn the α-channel models for each node where four region process are used as in [11], the classifiers for edge/bar and the other three junctions are, as illustrated in (c), trained using feature statistics (local histograms) of filter responses, such as first and second derivative Gaussian filters, LoG (Laplacian of Gaussian) filters, DoG (difference of Gaussian) and elongated DooG (different of offset Gaussian), all being extracted at 3 scales and 15 orientation (if had). The β-channels are, as illustrated in (d), based on active basis model [16] to do explicit binding from edges/bars and region processes to junctions, similar to the approach in [15]. In testing, in order to handle rotations (except region

processes), the algorithm searches different angles (15 orientations) for α-channel and for β-channel the algorithm firstly selects a proposed edge/bar, then searches the others (edges/bars and regions) in the allowed range of activities (say, the range of relative angles) to do binding.

The results: strong β binding property at the low-middle level vision. Fig.4 shows the calculated information contributions of the three channels of the five nodes, which clearly indicates that the strong β-channel at the low-middle level by both computer algorithm and human perception experiments. In addition, the ROCs also justify this point.

4.2. Faces and cars at high level vision

In our second experiment, we test human faces and cars which are two of the most studied patterns in the literature [14, 16, 13, 3, 7]. Fig.7 (c) shows the face detection results for Fig.1 (a) using the same features and AdaBoost method in [14], which only work well on the frontal faces whose α channels are on.

The data. Here, we need the data with ground truth for both objects and their parts, and we found the data from LHI database [17] which fits the requirement. For faces, we get training data from LHI dataset and use MIT+CMU

dataset [3] as testing set. For cars (side view at the present), both training and testing data are from LHI dataset, and in testing data, different degree of occlusions and smoothing operations are randomly added to explicitly test the β and γ channels.

Training and testing. Given the AoG in Fig.5, we learn α channels for each node using the AdaBoost method in [14] and the active basis model in [16], both are trained under three scales. The learnt active basis models are shown in the Fig.5. For the β channel, we learn the conditional geometric models as β binding models on attributes such as the relative locations, scales and orientations, as illustrated in Fig.5, which will be used to bind the hypotheses of parts. For the γ channel, we use head-shoulder as the parent node for human face node and learn its predicting model also expressed in a conditional geometric model as illustrated in Fig.5 (a), for cars, we do not test its γ channel at the present because the data we have are not the natural setting for cars, such as cars in street scenes [10, 4].

The results. Human faces have a strong α channel, but under occlusions, the β-channel would help a lot, and under too low resolutions such as in video surveillance, the γ channel would dominate. On the contrary, cars have a strong β channels binding from the wheel, especially together with windshield. In real situations, the image data include different kinds of scale changes and occlusions, entailing the integration of the three channels as shown by the explored information contributions and also the performance improvements by the integration of the three channels in Fig.7.

5. Conclusion

This paper numerically evaluates information contributions of individual bottom-up processes and top-down processes. We identify two bottom-up processes and one top-down process in the hierarchical And-Or graph representation, termed α, β and γ computing channels respectively. We separately train their models by a blocking strategy and individually measure their information contributions in typical vision tasks such as detecting junctions, human faces and cars. Based on the evaluated results, we integrate the three computing channels in our detection experiments with performance improvements obtained. This work is starting to discover how to schedule bottom-up and top-down processes in object recognition in the on-going works, and we hope that the proposed method can be generalized to various objects and datasets.
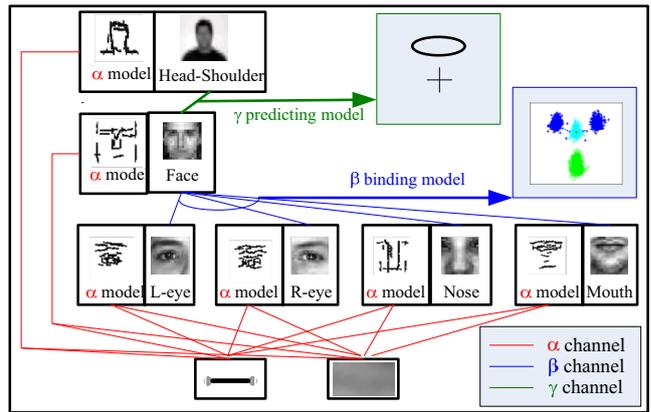
(a) And-Or graph representation for faces
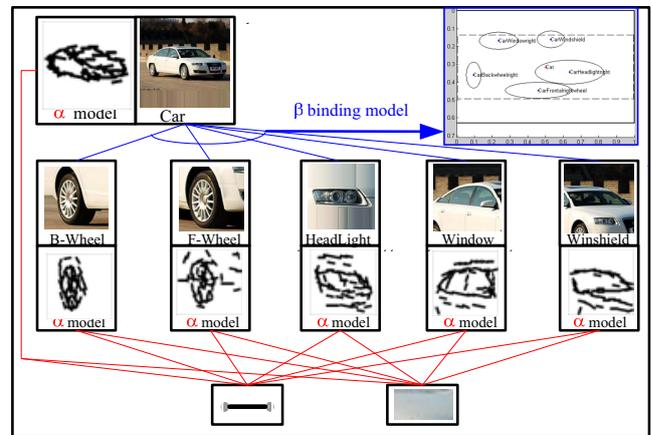
(b) And-Or graph representation for cars

Figure 5. The And-Or graph representation for faces in (a) and cars in (b) with the learnt active basis models [16] for α channels displayed for each node. Also, we train an another AdaBoost classifier for each node using the same method in [14]. The β binding models and the γ predicting model are also learnt as illustrated in the figure.
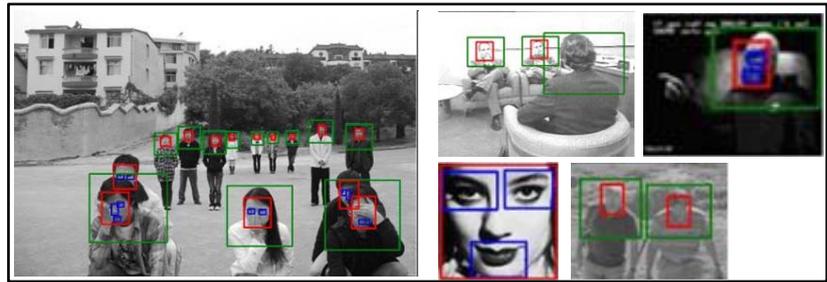
References

[1] G. Blanchard and D. Geman. Hierarchical testing designs for pattern recognition. *Ann. Statist.*, 33(3):1155–1202, 2005. 1, 3

[2] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 3, 4

[3] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. *IJCV*, 74(2):167–181, 2007. 5, 6

[4] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. In *CVPR*, volume 2, pages 2137 – 2144, June 2006. 6
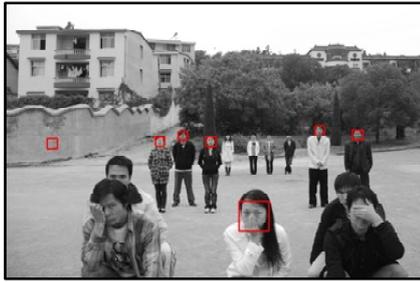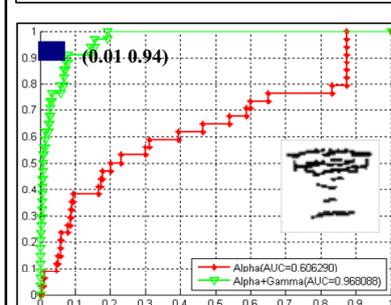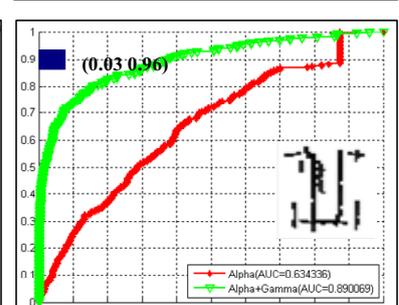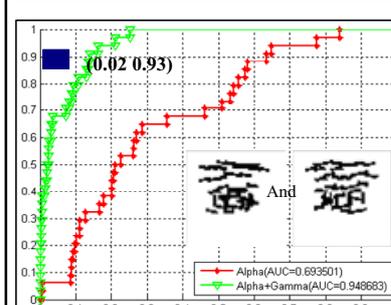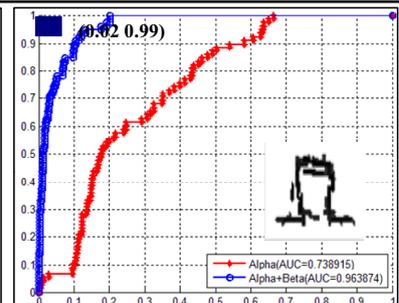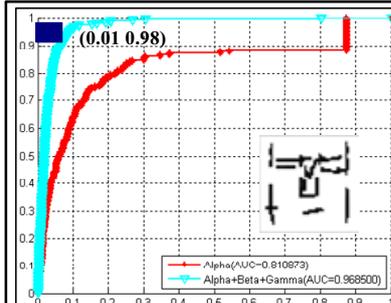
(a) Results of Viola's AdaBoost method [18].　　(b) Results by integrating α+β+γ channels in which α is Viola's method and active basis model
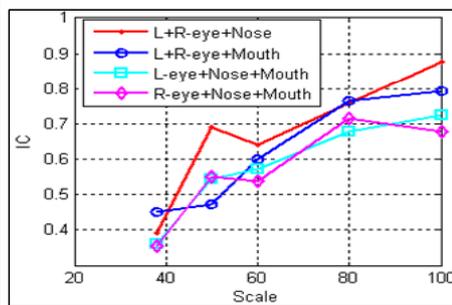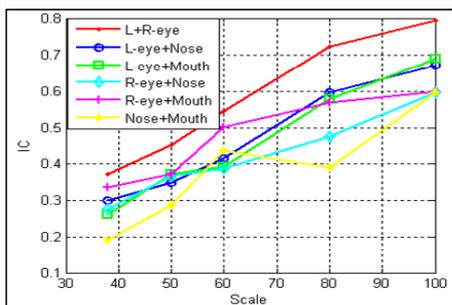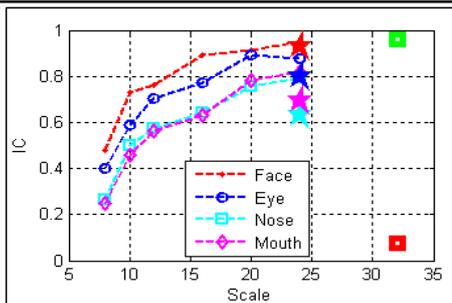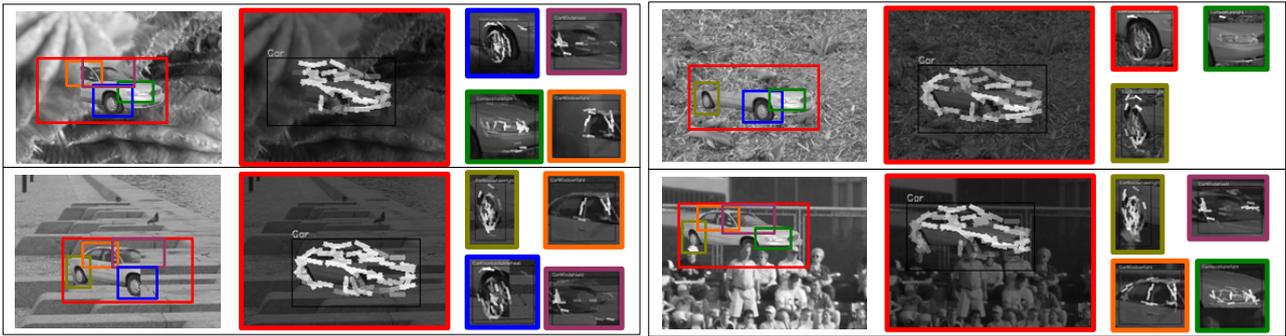
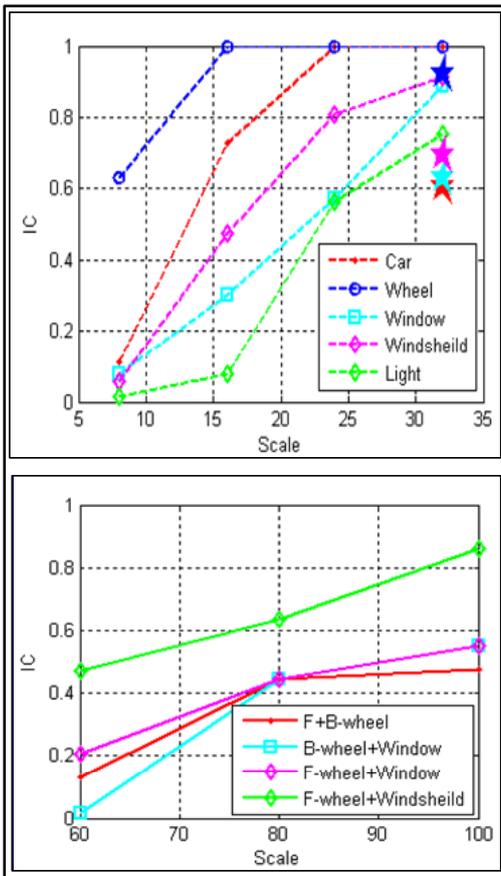(c) Information contributions　　　　　　　　(d) ROCs for different nodes

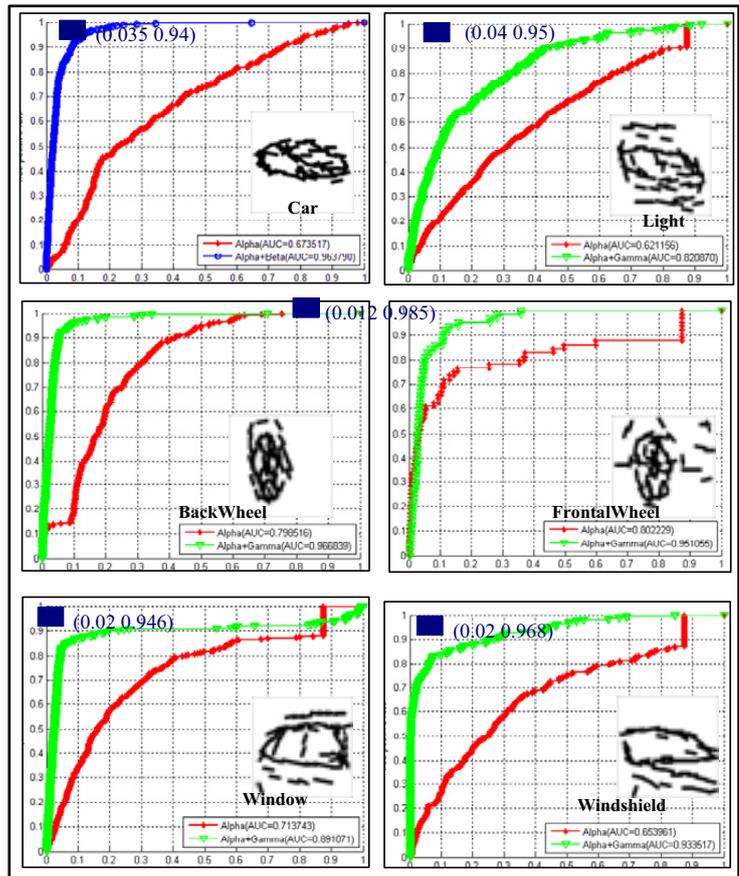Figure 6. The information contributions and testing ROCs for faces.

[5] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999. 1

[6] K. Shi and S. C. Zhu. Mapping natural image patches by explicit and implicit manifolds. *CVPR*, 2007. 5

[7] B. O. Y. Sinha, P. Balas and R. Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, 2006. 5

[8] D. P. Stuart Geman and Z. Chi. Composition systems. *Quart.*

[9] M. C. Thorpe S, Fize D. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996. 1

[10] A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 6

[11] Z. Tu and S. C. Zhu. Image segmentation by data-driven markov chain monte carlo. *PAMI*, 24(5):657–673, 2002. 1, 5

[12] S. Ullman. Visual routines. *Cognition*, 18:97–159, 1984. 1

[13] S. Ullman, M. V. Naquet, and E. Sali. Visual features of in-

Appl. Math, 60(4):707–736, 2002. 1

(a) Results of car detection using $\alpha + \beta + \gamma$ computing channels and the cars are under different kinds of occlusions

(b) Information contributions

(c) ROCs

Figure 7. The information contributions and testing ROCs for cars.

termediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002. 5

[14] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1, 2, 4, 5, 6

[15] T. F. Wu, G. S. Xia, and S. C. Zhu. Compositional boosting for computing hierarchical image structures. In *CVPR*, 2007. 1, 4, 5

[16] Y. N. Wu, Z. Z. Si, C. Fleming, and S. C. Zhu. Deformable template as active basis. *ICCV*, 2007. 1, 4, 5, 6

[17] B. Yao, X. Yang, and S. C. Zhu. Introduction to a large scale general purpose ground truth dataset: methodology, annotation tool, and benchmarks. *EMMCVPR*, 2007. 5

[18] A. L. Yuille, D. S. Cohen, and P. W. Hallinan. Feature extraction from faces using deformable templates. *IJCV*, 8:99–111, 1992. 1, 4

[19] L. Zhu and A. L. Yuille. A hierarchical compositional system for rapid object detection. *NIPS*, 2005. 1

[20] S. C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362, 2006. 1, 2, 3, 4, 5