

# Circle & Search: Attribute-Aware Shoe Retrieval

JUNSHI HUANG and SI LIU, National University of Singapore  
JUNLIANG XING, Institute of Automation, Chinese Academy of Sciences  
TAO MEI, Microsoft Research Asia  
SHUICHENG YAN, National University of Singapore

Taking the shoe as a concrete example, we present an innovative product retrieval system that leverages object detection and retrieval techniques to support a brand-new online shopping experience in this article. The system, called Circle & Search, enables users to naturally indicate any preferred product by simply circling the product in images as the visual query, and then returns visually and semantically similar products to the users. The system is characterized by introducing *attributes* in both the detection and retrieval of the shoe. Specifically, we first develop an attribute-aware part-based shoe detection model. By maintaining the consistency between shoe parts and attributes, this shoe detector has the ability to model high-order relations between parts and thus the detection performance can be enhanced. Meanwhile, the attributes of this detected shoe can also be predicted as the semantic relations between parts. Based on the result of shoe detection, the system ranks all the shoes in the repository using an attribute refinement retrieval model that takes advantage of query-specific information and attribute correlation to provide an accurate and robust shoe retrieval. To evaluate this retrieval system, we build a large dataset with 17,151 shoe images, in which each shoe is annotated with 10 shoe attributes e.g., heel height, heel shape, sole shape, etc.). According to the experimental result and the user study, our Circle & Search system achieves promising shoe retrieval performance and thus significantly improves the users' online shopping experience.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering*; I.2.6 [Artificial Intelligence]: Learning—*Knowledge acquisition*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Shoe retrieval, object detection, attribute learning

## ACM Reference Format:

Junshi Huang, Si Liu, Junliang Xing, Tao Mei, Shuicheng Yan. 2014. Circle & search: Attribute-aware shoe retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 1, Article 3 (August 2014), 21 pages.  
DOI: <http://dx.doi.org/10.1145/2632165>

## 1. INTRODUCTION

Nowadays, online shopping is becoming increasingly popular. Many Web sites, such as Amazon, eBay, Taobao, etc., provide convenient and economical platforms for people to buy their favorites. On these Web sites, fashion-related commodities make a huge market, within which shoes take a considerable proportion. However, some problems emerge when employing the current retrieval techniques on the online shopping Web sites. One of the most severe problems is the lack of semantic information in the representation of products.

---

This work is supported by the Singapore Ministry of Education under research grant MOE2010-T2-1-087. Dr. J. Xing was partially supported by the National Science Foundation of China under grant no. 61303178. Author's addresses: J. Huang (corresponding author), S. Liu, National University of Singapore, Singapore; email: a0092558@nus.edu.sg; J. Xing, Institute of Automation, Chinese Academy of Sciences, China; T. Mei, Microsoft Research Asia, China; S. Yan, National University of Singapore, Singapore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2014 ACM 1551-6857/2014/08-ART3 \$15.00

DOI: <http://dx.doi.org/10.1145/2632165>

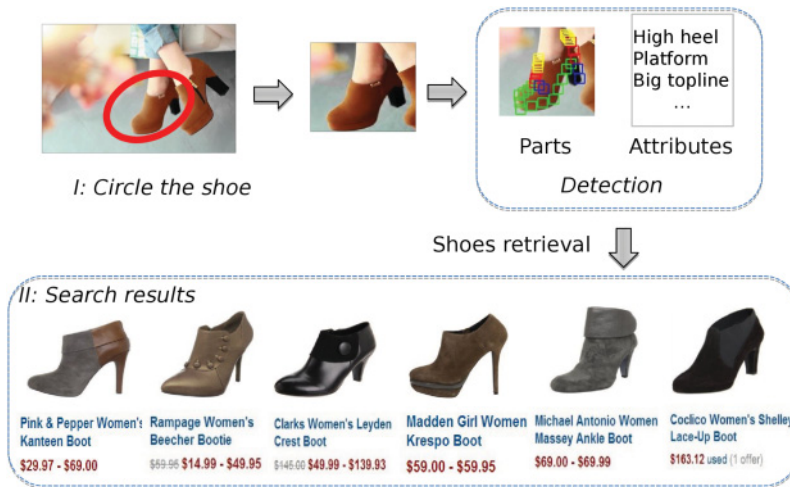


Fig. 1. Scenario illustration of the Circle & Search shoe retrieval system. The user browses a Web site and circles a shoe. The visually and semantically similar shoe images will be returned by the proposed system from the retrieval repository. All of the figures in this article are best viewed in original PDF file.

In this article, we propose an innovative shoe retrieval system named Circle & Search. The application scenario of our retrieval system is illustrated in Figure 1. Imagine the scenario that a user is browsing the online shopping Web sites and gets attracted by a pair of shoes, impelling him/her to buy a similar pair. A common way to find the similar product is to type some proper keywords into the search engine and select one's preference among the retrieval results. However, it is likely that one encounters difficulties when working out accurate descriptions as the keywords. One possible solution is to search the shoes with visual queries. In other words, with the circled shoe from query images, the top similar shoe images will be returned from the retrieval repository.

In our system, the query image can be a shoe product image or a shoe photo in daily life. However, there are large discrepancies between these two kinds of images. Particularly, the background of product images is relatively clean (like the retrieval result in Figure 1), while the background of daily photos is usually cluttered (like the query image in Figure 1). To decrease the discrepancies, previous studies prove that the object parts are more expressive for objects representation than the whole image [Bourdev et al. 2011; Liu et al. 2012]. By extracting the features from the semantic parts of objects, the influence of the background can be filtered out and the discrepancies caused by view changes can also be reduced [Yang and Ramanan 2011]. Therefore, some semantic parts of shoe are manually defined in our system (see Figure 2(a)), such as toe, heel, and vamp, etc. Additionally, some auxiliary parts are interpolated between every two semantic parts for better representation (see Figure 2(b)). To elaborately model the spatial relation of parts, three tree-structure models are designed for three views of shoes, that is, the frontal, half-profile, and profile view, respectively.

Besides the parts of the object, many recent methods [Chen et al. 2012; Siddiquie et al. 2011; Wang and Mori 2010] propose that the semantic attributes can help to enhance the representation of objects. Usually, the text-based shoe attributes are presented beside the, for example, "high heel", "round toe", etc.. Traditionally, these attributes are used as the complement of visual features by attaching to the entire object. However, we propose that the attributes should be attached to certain *parts* of objects. For example, it is more reasonable that the attribute "toe shape" is combined with the "toe" parts.

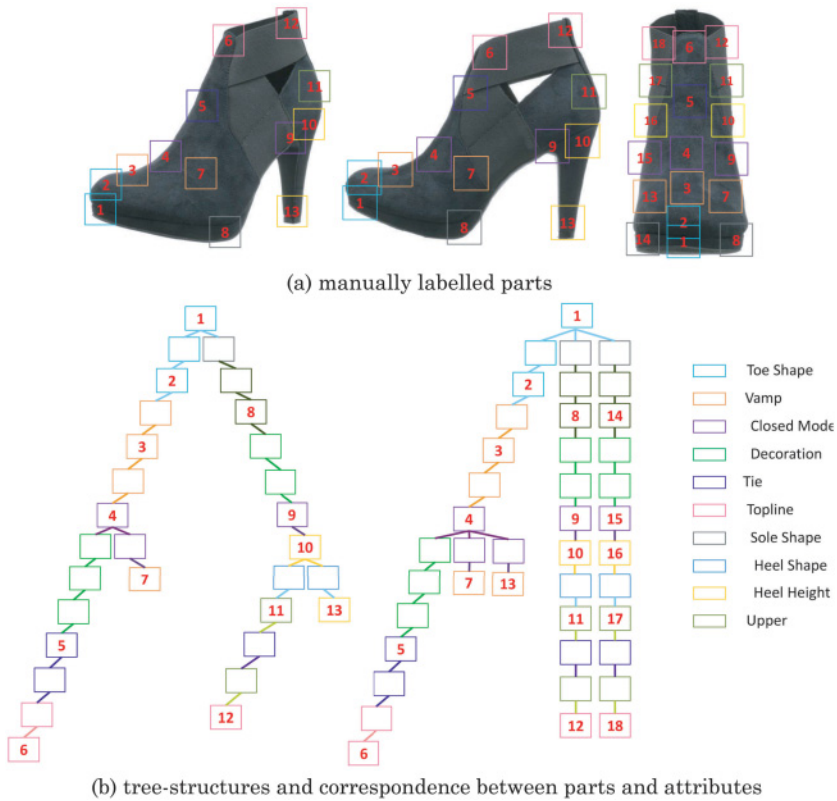


Fig. 2. The semantic parts of shoes in three views are presented in (a). The color of each part indicates its corresponding attribute. The mapping of part color and attributes can be seen in the right part of (b). The semantic meaning of these parts is introduced in Section 3.3. The number within each part indicates its order in the tree-structure model. By applying different combinations strategies of the manually labelled part in the detection model, we present two well-designed tree structures for profile view and frontal view, respectively. Similarly, the color of each part indicates its corresponding attributes. Note that some additional parts (the boxes without number) are interpolated into two manually labelled parts for better representation.

Therefore, the attributes are used as the high-order relation among shoe parts in our system.

Generally, our system can be divided into a detection component and a retrieval component. In the detection component, an attribute-aware part-based shoe detection model is proposed. Specifically, several tree-structure models are designed to model the shoes of different views. The nodes of each tree represent the predefined shoe part and the edges between every two nodes indicate the deformation between two parts. Besides this traditional framework, we claim that the key characteristic of our model is the use of consistency between attributes and relevant parts. Specifically, the appearance and deformation of parts should be influenced by the relevant attribute values, and vice versa. For example, if the attribute “toe shape” is of the value “round”, the visual appearance of the “toe” parts should be reasonably “round”. Meanwhile, the more precise location of parts can also help to improve the prediction of relevant attributes. Due to the consistency between parts and attributes, the detection of parts and the estimation of semantic attributes can be conducted via the Expectation Maximization (EM) approach until the consistency is achieved.

In the retrieval component, the detected shoe parts and the predicted attributes are fed into a query-specific attribute refinement retrieval model. This model aims to refine the attributes of the query and those of retrieval images in the retrieval repository simultaneously, with which the shoes in the repository are ranked and returned. The main contributions of this work can be summarized as follows.

- By defining the correlations between parts and attributes, we propose a novel attribute-aware part-based detection model. Due to the consistency between parts and attributes, the simultaneous shoe detection and attribute prediction can be performed efficiently in an EM manner.
- The predicted attributes and semantic parts can explicitly explain the ranking criterion of our retrieval system. Therefore, the result of our retrieval system is more explicit and expressive. Moreover, our system can handle the nonrigid objects due to its part-based property. This is still lacking in many state-of-the-art retrieval systems.
- To the best of our knowledge, this is the first time to comprehensively explore the shoe retrieval problem in the multimedia area. This problem has great market potential in practice. Meanwhile, the query-specific attribute refinement retrieval model for this problem is totally new in the retrieval area.
- We collect a large-scale dataset of shoes, with 17,151 shoe images annotated with 10 shoe attributes. Each shoe has 3 ~ 4 images of different views.

The article is organized as follows. In Section 2, we briefly present the latest relevant research progress. In Section 3, the collection of the shoes dataset is discussed. In Section 4, we introduce the Circle & Search system, including the shoe detection model and shoe retrieval model. The experiments are demonstrated in Section 5. The concluding remarks are given in Section 6.

## 2. RELATED WORK

### 2.1. Object Retrieval

The study of object retrieval has attracted much attention, both in academic and industrial areas. In Kovashka et al. [2012], a feedback system using relative attributes is proposed for retrieval. Besides attribute, the browse and search behaviours of users are also used to improve the online shopping experience in Lu et al. [2012], where the system is deployed on a tablet pad by taking advantage of the multitouch interfaces and the proposed interactive visual search system. Shen et al. proposed a method to automatically extract the query object for mobile product image search [2012]. By extracting the query object, the influence of cluttered background on visual features is removed and the retrieval performance is significantly improved. Particularly, He et al. [2012] proposed a novel mobile search system based on the “bag of hash bits”, where the image is represented as bag of hash bits. Overall, this system shows good searching performance and efficiency. In Arandjelovic and Zisserman [2011], the authors described a scalable method for smooth object retrieval, within which the real-time system can localize all the occurrences of outlined objects. By using subspace decomposition, Jegou et al. [2011] introduced a product-quantization-based method for approximate nearest-neighbour search.

In practice, many Web sites provide the shoe retrieval function, such as Google Goggles<sup>1</sup>, Baidu Stu<sup>2</sup>, etc., that allow users to upload an image and return similar products. Although no details are available about their techniques, it is likely that

---

<sup>1</sup><http://www.google.com/mobile/goggles>.

<sup>2</sup><http://stu.baidu.com>.



Fig. 3. Some shoe examples of different views in our dataset. the images at the first row are the product images; the images at the second row are daily photos. Totally, our dataset contains 17,151 shoe images of several views.

some visual features are extracted from the entire images or fixed patches of images, and certain distance metrics are designed to calculate the similarities.

## 2.2. Object Detection

In Felzenszwalb et al. [2010], the authors proposed a pictorial structure model using the mixtures of parts. However, these parts are greedily placed to cover the high-energy regions in a specific area. The uncertainty of those parts causes difficulty in associating the attributes with specific parts. In Yang and Ramanan [2011], the authors proposed an effective and flexible extension of the part-based model. By defining different types of mixtures in every part, this model is suitable in many specific situations. For example, if the type of mixtures is defined as the orientations of instances, the parts can precisely model the articulation of objects. Moreover, this model provides a general framework for modelling the co-occurrence relations of parts, as well as the spatial relations between parts, that construct the foundation of our detection model.


## 2.3. Attributes Analysis

The methods of attribute learning have been widely applied in many computer vision and multimedia tasks. In Ferrari and Zisserman [2008], the authors presented a probabilistic generative model to learn the visual attributes. In Farhadi et al. [2009], with the discriminative attributes, the objects are effectively categorized by using the compact attribute representation. Similarly, the authors in Parikh and Grauman [2011] built a set of discriminative attributes by interactively displaying categorized object images to humans. In Kumar et al. [2009], the authors proposed the attributes and simile classifiers that describe the face appearances and demonstrated the competitive results for the application of face verification. Siddiquie et al. [2011] explored the co-occurrence of attributes for image ranking and retrieval with multi-attribute queries. In the fashion-related area, researchers extracted the attributes by mining the images and their descriptive texts from the Internet [Berg et al. 2010], or by manually defining some domain-specific attributes [Chen et al. 2012; Liu et al. 2012].

## 3. THE SHOES DATASET

Recently, Kang et al. [2012] collected a database of 5 million product images that contains 1.2 million objects with multiple views. Shen et al. [2012] collected a real-life dataset of sport product images in 10 categories (hats, shirts, trousers, shoes, socks, gloves, balls, bags, neckerchiefs and bands) with 43,953 images. However, these datasets are collected for general product search. In this article, we construct a new dataset specific for the shoe retrieval task.



Attribute Name	Attribute Values			
Heel Shape				
	High-thin	Thick	Cubic	Wineglass
Toe Shape				
	Round	Square	Pointed	Fish-mouth
Vamp				
	Net	Dot	Stripe	Colorful
Upper				
	Super-high	High	Middle	Low
Closed Mode				
	Velcro	zipper	Shoelace	Elastic
Heel Height				
	Super-high	High	Middle	Short





Attribute Name	Attribute Values			
Heel Shape				
	Cone	Wedge	Flat	
Decoration				
	Frontal	Lateral	Rear	None
Sole Shape				
	Platform	Thick	Flat	
Tie				
	Front-strap	Back-strap	None	
Topline				
	Small	Big		

Fig. 4. The illustration of shoes attributes. The first column of each table is the attributes name and the rest of the columns of each table are the corresponding attribute values.

### 3.1. Shoe Images Collection

Some example images of our dataset are shown in Figure 3. The images are collected from some online shopping Web sites (e.g., Amazon.com) and photo sharing Web sites (e.g., Flickr.com), by using queries such as “shoes”, “footwear”, “boots”, “sandals”, etc. In total, 17,151 images are collected.

### 3.2. Attribute Annotation

In this dataset, 10 shoe attributes are defined. These attributes are learned from the study of several online shopping Web sites. Some students are hired to annotate shoe attributes on the whole dataset, with three annotators assigned for each image. A label is considered as true if more than two annotators agree with it. We double-check all the annotations to guarantee their accuracy. The illustration of shoe attributes, including name and values, is shown in Figure 4.

### 3.3. Shoe Parts Annotation

To handle the out-of-plane rotation, three views are defined according to the angle of the out-of-plane rotation of the shoe. Generally, the angles of frontal view, half-profile view, and profile view are around  $0^\circ$ ,  $45^\circ$ , and  $90^\circ$ , respectively. Then, 13 parts are defined for the shoes of profile view and half-profile view, within which 11 parts are selected for the shoes of the frontal view. The same students are hired to label the views and parts of shoes, and each image is annotated by three students. We filter out those images whose views are not agreed upon by all three students. Finally, the average position of parts is considered as the ground-truth annotation if the views of images are labelled.

To fully capture the appearance of the shoe in each view, we design several tree-structure models and select the optimal structure for each view by applying Yang and Ramanan [2011] on each model. The structures of the model for the profile view and frontal view are shown in Figure 2(b). The nodes of trees represent the shoe parts

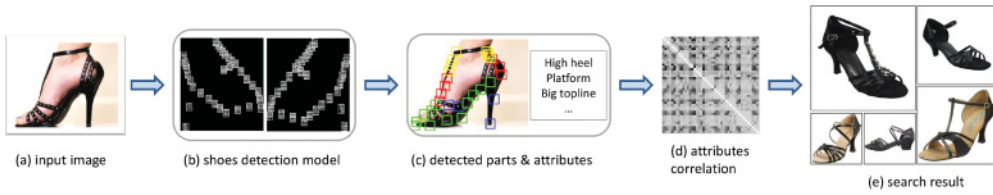


Fig. 5. The framework of our proposed shoe retrieval system: given a shoe image (a), the parts and attributes of the shoe (c) are predicted by our attribute-aware part-based detection model (b). These parts and attributes are then fed into a query-specific attribute refinement retrieval model (d) for refinement, with which as features the search results (e) are returned.

and the edges of trees are the deformation between every two parts. The semantic attributes are attached to several relevant parts. Note that shoes of different views contain differing numbers of parts. The definitions of parts and their relevant attributes are listed as follows.

- Part 1 and Part 2*. These are the two ending parts of the toe, combined with the attribute *toe shape*.
- Part 3*. This is the neck of the big toe. This part represents the attribute *vamp*.
- Part 4*. This is the surface of the foot. This part is located between Part 3 and Part 5 and combined with the attribute *closed mode*.
- Part 5 and Part 12*. Part 12 is close to the Achilles tendon. Usually, there is a concave on the human foot at Part 12. Part 5 is in front of Part 12. The attribute *tie* is associated with these two parts. The distance between Part 12 and Part 13 is the *upper*.
- Part 6 and Part 13*. These are the uppermost parts of the shoe. The distance between them is the length of the *topline*.
- Part 7*. This is the space beside the vamp, representing the attribute *vamp*.
- Part 8*. This position is at the big toe mound. It is at the middle of the toe and the foot arch (Part 9). The attribute *sole shape* is also described by this part.
- Part 9*. This part is at the inner arch of the foot, which represents the attribute *decorations*.
- Part 10 and Part 11*. These are the two ending parts of the heel. The distance between them is the attribute *height of heel*. Their shape represents the attribute *heel shape*.

#### 4. THE CIRCLE & SEARCH SYSTEM

The framework of our system is illustrated in Figure 5. Given a query image, the attribute-aware part-based detection model detects the locations of shoe parts and predicts the corresponding attributes. Those predicted attributes, that can fully capture the properties of the query image are concatenated to form the semantic feature. Based on this feature, the retrieval images are ranked and returned as the retrieval result. However, because different attributes are predicted independently, there should be noise in the predicted result. Thus, we utilize the co-occurrence of attributes in the retrieval repository to refine the predicted attributes by our query-specific attribute refinement model. Finally, the retrieval images are ranked and returned according to the refined attributes.

##### 4.1. Attribute-Aware Shoe Detection

*4.1.1. Hierarchical Mixture.* Inspired by the part-based detection approaches [Felzenszwalb et al. 2010; Yang and Ramanan 2011], the tree-structure models constructed by a set of deformable semantic parts are used to represent the shoes of

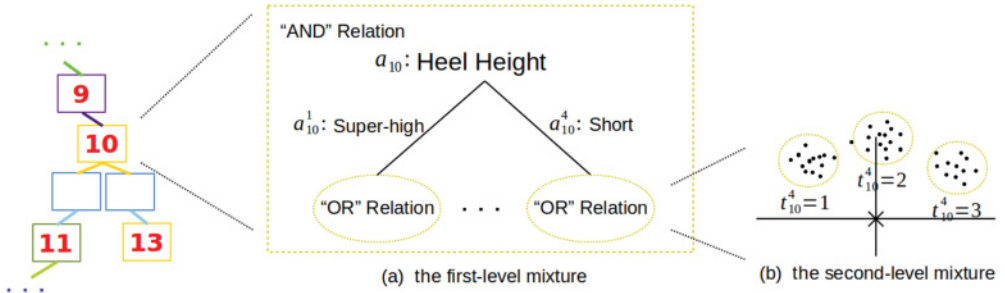


Fig. 6. The subfigure (a) presents the AND-OR hierarchical structure of Part 10. First, the 10-th parts of the training samples are divided into several components with the AND relation by attribute values, which constructs the first-level mixture (a). Each component of the first-level mixture consists of  $K$  components with the OR relation generated by  $K$ -means (b). The cross at the origin of coordinates at subfigure (b) represents the position of Part 9 and the feature of parts for  $K$ -means is the scaled distance along the  $x$ -axis and  $y$ -axis from Part 10 to its parent part, namely Part 9.

different views. Due to the variance of shoes, the visual appearance of shoes cannot be fully captured by the low-level image features, therefore, for better representation, the shoe attributes are introduced into our detection model.

Denote the shoe image as  $I$ ,  $\{l_i = (x_i, y_i)\}_{i=1}^N$  are the positions of  $N$  shoe parts in  $I$ , and  $\{\alpha_i\}_{i=1}^M$  is the  $M$  shoe attributes. Usually, each attribute has several attribute values (see Figure 4) and we write  $\alpha_i = [\alpha_i^1, \dots, \alpha_i^{n_{\alpha_i}}]$ , where  $n_{\alpha_i}$  is the number of values of attribute  $\alpha_i$  and each element in  $\alpha_i$  is the probability of the shoe having the corresponding attribute value. In our model, one attribute is associated with several parts and we simplify the model by assuming that each part only attaches to one attribute. Therefore, the attributes maintain the high-order relations between parts. The mapping between shoe parts and attributes can be obtained by defining two functions, namely  $i_a = f_a(i_p)$  and  $\mathbf{i}_p = f_p(i_a)$ , where  $\mathbf{i}_p$  denotes the indices of shoe parts affected by the  $i_a$ -th attribute and  $i_p \in \mathbf{i}_p$  is the index of the one specific part. Given the index of the specific part  $i_p$ , we can easily find out its corresponding attribute by  $f_a(i_p)$ .

For notational convenience, the attribute of the  $i$ -th part will be simply denoted as  $\alpha_i$  as long as no confusion is caused, and similarly we define  $\alpha_i = [\alpha_i^1, \dots, \alpha_i^{n_{\alpha_i}}]$ . With such notation, we should notice that  $\alpha_i$  and  $\alpha_j$  may be the same attribute, as one attribute is associated with several parts.

Intuitively, the attribute  $\alpha_i$  attached to the  $i$ -th part has an explicit effect on the visual appearance of this part. In other words, the same part in different images may be combined with different attribute values (e.g., the attribute *toe shape* may be *round* or *pointed* in different images), thus resulting in the discrepancy of appearance of this part in different images. Therefore, a first-level mixture, discriminated by the values of attribute  $\alpha_i$ , is constructed for the  $i$ -th part (see Figure 6(a)). Obviously, the number of attribute values  $n_{\alpha_i}$  is the number of components in the first-level mixture of the  $i$ -th part. Note that the relation of components in a first-level mixture is modelled as AND in our model, as we suggest that each value of a specific attribute is assigned to its related parts with certain probability.

Though the first-level mixture can considerably reduce the appearance discrepancy of the same parts between different images, we argue that the displacement between two parts is still different among shoes, even if the same part in different shoes has the same attribute value. To model the deformation of parts, the same parts in different shoes with the same attribute value are clustered into several groups according to their scaled spatial distribution.



Specifically, the deformation type of the  $i$ -th part with attribute value  $a_i^k$ , denoted by  $t_i^k \in \{1, \dots, K\}$ , is integrated into each component of the first-level mixture to construct a second-level mixture (see Figure 6(b)). Note that  $K$  is the number of components in each second-level mixture and we make  $K$  the same in our detection model. To generate the second-level mixture,  $K$ -means is applied on the training samples by using the normalized distance between each part and its parent part along the  $x$ -axis and  $y$ -axis as the feature. Obviously, the centres of  $K$ -means stand for the components of the second-level mixture. Different from the relations of components in the first-level mixture, the relation of deformation types  $t_i^k$ , namely the components of the second-level mixture, is modelled as an OR node. In other words,  $t_i^k$  can only be exclusively selected as one integer within the value range  $\{1, \dots, K\}$ .

Therefore, the  $t_i = [t_i^1, \dots, t_i^{n_{a_i}}]$ , which represents the types of components in the first-level mixture in the  $i$ -th part, can represent the hierarchical AND-OR structure of the  $i$ -th part. This hierarchical mixture can significantly stabilize the performance of our tree model, as it can greatly reduce the discrepancy resulting from the visual appearance and the deformation of parts.

**4.1.2. Model Formulation.** To introduce the model, let's denote  $G = (V, E)$  as a single tree, where  $V$  is the set of tree nodes, with each node corresponding to a shoe part, and  $E$  is the set of tree edges. The score function  $S(I, a, t, l)$  of this tree model can be written as follows with the configuration of attributes  $a$ , part types  $t$ , and positions  $l$ :

$$S(I, a, t, l) = \sum_{i \in V} a_i \odot (w_i^{t_i} \odot \phi(I, l_i) + b_i^{t_i}) + \sum_{ij \in E} a_{ij} \odot (w_{ij}^{t_i, t_j} \odot \theta(l_i, l_j) + b_{ij}^{t_i, t_j}), \quad (1)$$

where  $\phi(I, l_i) \in \mathbb{R}^d$  is the HOG feature of image  $I$  extracted at the  $i$ -th part, and  $\theta(l_i, l_j) = [dx, dy, dx^2, dy^2]^T \in \mathbb{R}^4$  indicates the relative position of the  $i$ -th part to the  $j$ -th part by defining  $dx = x_i - x_j$  and  $dy = y_i - y_j$ . Furthermore  $a_i \in \mathbb{R}^{n_{a_i}}$  is the probability vector of the  $i$ -th part containing values of attribute  $a_i$ . This probability vector is used to model the influence of attribute  $a_i$  on this part. Similarly,  $a_{ij} \in \mathbb{R}^{n_{a_i}} \times \mathbb{R}^{n_{a_j}}$  is the joint attribute value probability matrix of the  $i$ -th part and the  $j$ -th part, those is used to model the effect of pairwise attributes on two adjacent parts.  $\omega_i^{t_i} \in \mathbb{R}^{n_{a_i} \times d}$  and  $\omega_{ij}^{t_i, t_j} \in \mathbb{R}^{n_{a_i} \times n_{a_j} \times 4}$  are the model parameters to be learned. Note that here  $\odot$  is a generalized dot-product operator that can be performed on two tensors of different orders and dimensions. Denoting  $A \in \mathbb{R}^{m_1 \times \dots \times m_p \times n_1 \times \dots \times n_q}$  and  $B \in \mathbb{R}^{n_1 \times \dots \times n_q}$ , each element in the resulting tensor  $C = A \odot B \in \mathbb{R}^{m_1 \times \dots \times m_p}$  is calculated as

$$C(i_1, \dots, i_p) = \sum_{j_1} \dots \sum_{j_q} A(\dots, j_1, \dots, j_q) B(j_1, \dots, j_q). \quad (2)$$

**Appearance Model.** The first term in Eq. (1), called the appearance model, indicates the local response of putting a set of templates  $w_i^{t_i}$  at position  $l_i$  for the  $i$ -th part by tuning the attribute value probability vector  $a_i$  and the types  $t_i$ . It should be emphasized that the types  $t_i \in \mathbb{R}^{n_{a_i}}$  and each element  $t_i^{k_i}$  in the type vector  $t_i$  indicate the index of the component in the  $k_i$ -th second-level mixture of the  $i$ -th part. The bias  $b_i^{t_i}$  is the preference of assigning types  $t_i$  to the  $i$ -th part with different attribute values. Obviously, the formula of the appearance model indicates its AND-OR node structure, as only one response of the template from each second-level mixture is selected and the response of the appearance model in each part is the weighted sum of selected responses from every second-level mixture.

*Deformation Model.* Given a specific combination of joint attribute values ( $a_i^{k_i}, a_j^{k_j}$ ) for the adjacent  $i$ -th part and  $j$ -th part, the different combinations of displacement ( $t_i^{k_i}, t_j^{k_j}$ ) are determined by the normalized distance between these two parts. Each combination presents the particular relative placement of the  $i$ -th and  $j$ -th part under the condition that the  $i$ -th (or  $j$ -th) part is assigned as  $k_i$ -th (or  $k_j$ -th) attribute value with probability  $a_i^{k_i}$  (or  $a_j^{k_j}$ ). By tuning the combinations of types for two adjacent parts and the probability vector of their joint attribute values, the second term in Eq. (1), also known as the deformation cost, controls the co-occurrence of spatial information and joint attributes combination between two parts. Similarly, the bias  $b_{i_j}^{t_i, t_j}$  presents the preference of a particular co-occurrence of types combination ( $t_i, t_j$ ).

*Attribute Integration.* In our detection model, we suggest that the semantic attributes can affect the visual appearance of multiple parts and the deformation between every two parts. Therefore, the attributes are considered as a high-order relation of relevant parts. For example, if the attribute *heel shape* of a certain shoe is of the value *high-thin*, the two ending parts of the heel should be relatively *thin* at the same time. Therefore, the selection of optimal templates for parts will be constrained to fit the global relations preserved by attributes. To integrate the attribute into one part, we rescale the response of templates in each second-level mixture by multiplying the probabilities of corresponding attribute values and sum up the rescaled response of the optimal template from each second-level mixture as the response of this part.

Specifically, attribute classifiers are pretrained by using the concatenated low-level image features of detected parts. During the detection, the probability vector of attribute  $a_i$  can be obtained by inputting the concatenated features of current detected parts into a corresponding classifier. Each element of probability vector  $a_i^k$  indicates the probability of relevant parts having the  $k$ -th value of attribute  $a_i$ , namely the  $k$ -th component of the first-level mixture in the  $i$ -th part. For each component in the first-level mixture, its response is the highest score of the template in the second-level mixture multiplied its attribute value probability. This again indicates that the displacement type in each second-level mixture is selected as an OR relation and the component in every first-level mixture is selected as an AND relation. Note that one specific attribute may affect several parts at the same time, indicating that the placement of parts must be affected by the high-order relation maintained by attributes. Therefore, the hierarchical the AND-OR structure and high-order relations can enhance the performance of part detection and attribute prediction by unifying the appearance of parts and the semantic meaning of attributes.

*4.1.3. Model Inference.* With the learned model parameters ( $\omega_i, \omega_{ij}, b_i, b_{ij}$ ), we can detect the parts and attributes of the shoe by maximizing the score function  $S(I, a, t, l)$  over  $a, t$ , and  $l$ . In practice, the feature pyramid is extracted to decide the optimal scale. However, as with the integration of attributes, the score function  $S(I, a, t, l)$  is hard to solve due to its nonconvex property. Fortunately, an EM-based approach can be applied in the inference to iteratively achieve the solution. Generally, when fixing  $a$ , Eq. (1) becomes convex over  $t$  and  $l$  and can be effectively solved by dynamic programming due to its tree structure [Felzenszwalb et al. 2010; Yang and Ramanan 2011]. After getting the current optimal  $t$  and  $l$ , we can calculate the expected attributes  $a$  according to the current predicted parts. These two steps iterate until convergence is achieved.

*Initialization.* For notational convenience, we define  $z_i = (l_i, t_i)$  as the location and types of the  $i$ -th part. In the initialization, we aim to detect the initial parts of the shoe. To eliminate the influence of attributes, we pretrained a normal detection model

without attributes, to get an initial estimation of the locations and types of parts via this detection model.

*E-Step.* The E-Step aims to estimate the expected attributes  $a$  based on the positions  $l$  and types  $t$ . Specifically, the relations between parts and attributes are predefined by training a multiclass linear SVM for each attribute. The input of SVM is the concatenation of low-level image features extracted from relevant parts. Based on the detected parts in the previous step (denoted as  $z$ ), we concatenate the features of corresponding parts and estimate the expectation values of the attribute from the normalized score estimated by the multiclass linear SVM. Formally, this procedure can be written as

$$\hat{a}_i = f_i(\varphi(I, z_{f_p(f_a(i))})), \quad (3)$$

$$\hat{a}_{ij} = \mathbf{E}(a_{ij}) = f_{ij}(\theta(z_i, z_j)), \quad (4)$$

where  $f_i(\cdot)$  and  $f_{ij}(\cdot)$  denote the attribute classifiers. The  $\varphi(I, z)$  denotes the concatenated feature  $\phi(I, l_i)$ , where  $l_i \in z$ , and  $\theta(z_i, z_j)$  is the spatial feature similar to  $\theta(l_i, l_j)$  in Eq. (1).

*M-Step.* The M-Step aims to estimate the position  $l$  and type  $t$  in  $z$  of every part based on the updated attribute  $a$ . This procedure can be conducted by fixing attribute  $a$  and evaluating the following objective function using dynamic programming,

$$\text{score}_i(z_i) = \hat{a}_i \odot (w_i^{t_i} \odot \phi(I, l_i) + b_i^{t_i}) + \sum_{k \in \text{kids}(i)} m_k(z_i), \quad (5)$$

$$m_k(z_i) = \max_{z_k} [\text{score}_k(z_k) + \hat{a}_{ki} \cdot (w_{ki}^{t_k, t_i} \cdot \theta(l_k - l_i) + b_{ki}^{t_k, t_i})], \quad (6)$$

where  $\text{kids}(i)$  is the set of children nodes of the  $i$ -th part and empty for the leaf parts. During detection, the tree model starts from the leaves and moves upwards until arriving at the root node.

Given a fixed attribute  $\hat{a}_i$ , Eq. (5) calculates the current local score of the  $i$ -th part over every pixel position  $l_i$  and types  $t_i$ , then collects the score message from its children. Particularly, the first term in Eq. (5) can be computed by traversing the whole feature pyramid with different types of mixtures. Eq. (6) adds the local score of the  $k$ -th part with relative deformation cost and passes the best score as the message to its parent node. Note that the fixed pairwise attribute  $\hat{a}_{ki}$  is computed in the previous expectation step. Once the message arrives at the root part, the configuration of the root part with the best score becomes the optimal configuration of the current detection over position  $l_1$  and type  $t_1$ . By keeping the trace of message passing, one can backtrack the direction from root to leaves to get the optimal configuration of each part.

*Time Complexity.* Due to the linear-time complexity of the E-Step, the time complexity of our model is mainly decided by the complexity of the M-Step. The M-Step concentrates on the dynamic programming in Eq. (6) over every position  $l$  and type  $t$ . In practice, the distance transform [Felzenszwalb and Huttenlocher 2004] is used to calculate the message of each part on every candidate position  $l$  with  $O(|l|)$  complexity. By looping over every  $|t| \times |t|$  possible types of parent nodes and children nodes, the complexity of this part becomes  $O(|l| \times |t| \times |t|)$ , which is also the complexity of our detection model.

*4.1.4. Model Parameter Learning.* The model is trained in a supervised learning paradigm. Given the labelled positive example set  $I_{\text{pos}}$  with annotations  $(a_{\text{pos}}, t_{\text{pos}}, l_{\text{pos}})$  and negative example set  $I_{\text{neg}}$ , we aim to solve a structured object function similar to

those proposed in Felzenszwalb et al. [2010] and Yang and Ramanan [2011]. For notational convenience, we denote  $y_n = (a_n, t_n, l_n)$  as the prior information, where  $n \in \text{pos}$ . Since the score function is linear in model parameters  $\beta = (\omega, b)$ , it can be rewritten as  $S(I, y_n) = \beta \cdot \Phi(I_n, y_n)$ , where  $\Phi(I_n, y_n)$  is the concatenation of appearance or deformation features. To maximize the score function, the model is learned in the max-margin form:

$$\begin{aligned} \min_{\omega, \xi_n \geq 0} \quad & \frac{1}{2} \beta \cdot \beta + \lambda \sum_n \xi_n \\ \text{s.t. } \forall n \in \text{pos}, \quad & \beta \cdot \Phi(I_n, y_n) \geq 1 - \xi_n \\ \forall n \in \text{neg}, \forall y, \quad & \beta \cdot \Phi(I_n, y) \leq -1 + \xi_n. \end{aligned} \quad (7)$$

The preceding quadratic program problem is known as structural SVM and can be solved by many optimization solvers. In our experiments, the dual coordinate descent algorithm developed in Yang and Ramanan [2011] is adopted to solve the problem. The learning procedure can be roughly separated into two subprocedures. The first subprocedure includes the construction of hierarchical mixtures, the learning of separate mixtures, and the learning of part deformations with the labelled attribute values. The second subprocedure is the adjustment of parts and attributes, as the consistency of parts and attributes is also required in the learning procedure.

*Learning with Labelled Attributes.* As aforementioned, each part in the tree model consists of a hierarchical mixture, where the first-level mixture is discriminated by the labelled attribute values. In each component of the first-level mixture, the second-level mixture is constructed by  $K$ -means with the relative distance between the a parent part and children part as a feature. At the beginning, the template of each component in the second-level mixture in one part is trained on the image parts that contain the corresponding attribute value and type. This indicates that the attribute value probability vector  $a_i$  of the  $i$ -th part is one if we use the labelled attributes. After learning the templates for each part, the weights of the deformation model are calculated in a second-round training with labelled information. Consequently, an initial tree model is constructed. Note that the attribute classifiers are also learned separately according to the corresponding features and labelled attributes.

*Adjustment of Parts and Attributes.* The detection problem is easy if the labelled attributes are available both in the training and the testing procedures. However, the problem becomes tough when the attributes are unknown in the testing procedure. In the inference section, we introduce an EM-based approach to keep the consistency between parts and attributes by choosing the optimal mixtures for every part. However, such consistency may not be achieved unless the tree model and attribute classifiers are consistent in the training procedure as well. Therefore, the EM-based approach is also conducted in the learning procedure. Note that the consistency in the training procedure is achieved by adjusting the weights of templates for each part, rather than selecting the optimal template.

In this subprocedure, the parts and attributes of the training images are slightly adjusted and thus the attribute classifiers and tree model will be updated accordingly. Specifically, assuming that the tree model  $tree^{(i)}$  is trained by the image parts  $l^{(i)}$ , types  $t^{(i)}$ , and attributes  $a^{(i)}$  at the  $i$ -th step, we can redetect the parts  $l^{(i+1)}$  on every training image and thus get the attributes  $a^{(i+1)}$  according to the hierarchical structure. Based on the detected parts  $l^{(i+1)}$ , we can predict their attributes  $a^{(i+1)'}$  by the pretrained classifiers  $classifier^{(i)}$ . Then, the attribute classifiers are retrained as  $classifier^{(i+1)}$  by the parts  $l^{(i+1)}$  and attributes  $a^{(i+1)}$  from tree model  $tree^{(i)}$ . Meanwhile, the tree model can also be updated as  $tree^{(i+1)}$  by the new attributes  $a^{(i+1)'}$  from attribute classifiers.

These two steps are iterated until convergence. In the experiments, we find the training is converged only in two or three iterations. Similar findings are also reported by Yang and Ramanan [2011].

#### 4.2. Query-Specific Attribute Refinement for Shoe Retrieval

The probability vector of attribute values of the query image, denoted by  $x$ , can be calculated by the detection model introduced in Section 4.1. The task of shoe retrieval is to calculate the similarity between  $x$  and the attribute value probability vector  $y$  of each candidate image in our retrieval repository  $Y$ . Traditionally, the ranking criterion uses the Euclidean distance, namely  $\|x - y\|_2$ .

However, the attribute value probability vectors  $x$  and  $y$  may be noisy. We propose to refine them by considering the correlations between different attribute values. For example, the value *high heel* of attribute *heel height* usually appears with the value *fish mouth* of attribute *toe shape*. To model the pairwise correlations between different attribute values, we obtain the co-occurrence matrix  $C$  of the attributes from the training dataset. Then, we calculate the Laplacian matrix  $L$  based on the co-occurrence matrix  $C$ . In this work, we propose the query-specific attribute refinement method, which to our best knowledge is totally new, by optimizing the following function:

$$\min_{x', y'} \|x' - x\|_2^2 + \|y' - y\|_2^2 + \alpha(x'^T Lx' + y'^T Ly') + \gamma \|x' - y'\|_2^2, \quad (8)$$

where  $x'$  and  $y'$  are the refined attribute values probability vector of  $x$  and  $y$ , respectively. The first two terms require that the refined attribute values should be similar to the original attribute values. The third term requires that the refined attribute values should also follow the correlations of attributes. The fourth term requires that the refined attributes of the query and the refined attributes of candidate shoes in the database should be similar. The underlying intuition is that Eq. (8) aims to align  $x$  and  $y$  by removing the possible noise existing in  $x$  and  $y$ . Though no rigorous theory can guarantee that the query-specific attribute refinement is better than individual refinement, our later experiments validate the effectiveness of this type of attribute refinement method. Obviously, Eq. (8) can be reformalized as

$$\min_{x', y'} \begin{bmatrix} x' \\ y' \end{bmatrix}^T D \begin{bmatrix} x' \\ y' \end{bmatrix} - 2 \begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad (9)$$

where  $D = \begin{bmatrix} (1+\gamma)I + \alpha L & -\gamma I \\ -\gamma I & (1+\gamma)I + \alpha L \end{bmatrix}$  and Eq. (9) can be solved by setting the derivative as zeros, and thus

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = D^{-1} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (10)$$

Based on the refined attribute from Eq. (10), we can get the final ranking according to  $\|x' - y'\|_2$ .

#### 4.3. Possible Expansion of the Retrieval System

Generally, our Circle & Search system contains two submodules, that is, an attribute-aware part-based detection model and query-specific attribute refinement retrieval model. Though we take the shoe as a concrete example to introduce our retrieval system, we suggest that our system can be easily expanded into other domains as long as the semantic attributes of objects in these domains can be considered as high-order relations between parts.

Taking clothes as an example, we can first define a human-shape model to represent the clothes, including upper body and lower body. Meanwhile, the attributes of clothes



can be collected from some professional Web sites, for example, collar, sleeve, shape, etc. (please refer to Liu et al. [2012] for more cloth attributes). With this information, one attribute may affect the appearance of several relevant parts, known as the high-order relation in our detection model. For example, the V-shape collar requires that the part near the chest should be in a “V” shape and the parts near both shoulders should be straight, where the round collar requires that the parts besides the chest should be relatively round. Therefore, we can easily train an attribute-aware part-based clothes detection model if the attributes of clothes indeed affect the appearance of several parts. With the result from our detection model, the query-specific attribute refinement model can calculate the co-occurrence matrix of clothes attributes and apply the attribute refinement method on the predicted attributes of query clothes and the ground-truth attributes of clothes in the repository. Finally, similar clothes in the repository should be found according to the ranking of similar refined attributes.

## 5. EXPERIMENTS

In this section, we evaluate the performance of our Circle & Search system on our collected database in terms of shoe detection and retrieval. Our shoes database contains 17,151 shoes images, including product images and daily-life photos. In the detection experiment, the detection of parts and prediction of attributes are used to evaluate the performance of detection models. In the retrieval experiment, the query-specific attribute refinement model refines the predicted attributes and ranks the retrieval images according to the refined attributes.

Generally, the detection result indicates that our attribute-aware part-based detection model improves the performance of part detection and attribute prediction. In the retrieval experiment, we claim that the query-specific attribute-refinement-based approach and the predicted attributes both contribute to the enhancement of our retrieval system.

### 5.1. Evaluation of Shoe Detection

*5.1.1. Experimental Settings.* In this experiment, we compare our detection model with the flexible mixtures-of-parts model proposed in Yang and Ramanan [2011]. As stated in Section 3.3, three well-designed tree structures are selected from a group of manually predefined tree structures for each view. Specifically, we apply the baseline model on every tree structure and select the tree structure with the best performance for each view. For fair comparison, the configurations of the baseline and our model are almost the same, except that the number of components in each mixture, namely  $K$ , is 6 in baseline while in our model  $K = 3$ . This indicates that we give prior advantages to the baseline. Beside the manually defined parts, auxiliary parts are interpolated between two labelled parts to enrich the representation of shoes in a specific view for both the baseline and our model. The cell size of the HOG feature is  $5 \times 5$  for each part.

In the training procedure, 750 images (250 images for each view), including product images and daily-life photos, are carefully selected from our shoes database as positive training images. Note that the rotation and flip operation are performed on the training data for data augmentation. The INRIA database [Dalal and Triggs 2005] is used as our negative training set. To evaluate the performance, 2,250 images (750 images for each view) are used as the test dataset, which also contains product images and daily-life photos.

We conduct two experiments to comprehensively evaluate our detection model. In the first part of the detection experiment, we assume the views of testing images are given and the detector of the relevant view is applied for detection. In the second part of the detection experiment, the views of testing images are unknown. To obtain the

Table I. The Comparison of Detection Performance between Baseline and Our Method

Detection Model	View	Mean APK	Mean PCK
[Yang and Ramanan 2011]	Frontal View	48.7%	64.8%
	Half Profile View	60.3%	72.6%
	Profile View	59.5%	73.4%
	Unknown	53.5%	66.7%
Our Detection Model	Frontal View	50.3%	66.5%
	Half Profile View	63.4%	75.8%
	Profile View	64.3%	76.8%
	Unknown	57.9%	70.5%

views, the normalized predicted scores are introduced so that the scores among the three models are comparable.

To evaluate the performance of detection, the metrics Average Precision of Keypoints (APK) and Precision of Correct Keypoints (PCK) [Yang and Ramanan 2011] are employed to evaluate the detection of parts. For the APK, the candidate is considered correct (true positive) if it lies beside the ground-truth part. Particularly, this metric can correctly penalize both misdetection and false positives. The PCK evaluation explicitly factors our detection by requiring the testing images to be annotated with a tightly cropped bounding box for each shoe. Note that we directly consider the images with wrong predicted views as the incorrect prediction in the second part of the detection experiment.

In the part of attribute prediction, because the baseline cannot predict the attributes, we use the multiclass linear SVM to predict the attributes by extracting the SIFT features from circled images and parts detected by Yang and Ramanan [2011] and the ground-truth parts, respectively. The precision is used to evaluate the performance of attribute prediction.

*5.1.2. Performance Comparison.* The results of part detection are demonstrated in Table I. Generally, our attribute-aware part-based detection model outperforms the baseline in both experiments. Specifically, the mean APK and mean PCK of our model are about 3.17% and 2.77% higher than the baseline if the prior knowledge of the view is known. This indicates that the integration of attributes can improve the detection performance. However, compared with the results in profile and half-profile views, the improvement of performance of the frontal-view model is slightly lower. A possible explanation is that some distinctive parts of the shoe in the frontal view are self-occluded.

In the second part of the detection experiment, because the views of testing images are unknown, the detection model has to predict the views of the testing images first. To make the scores of the three models comparable, we normalize the scores of the three models according to the score distributions of the training images predicted by their corresponding model. After normalization, the view with the highest score is considered as the candidate view of each testing image. Using this normalization strategy, the prediction accuracy of the view can reach up to 97.1% and 95.4% in our model and the baseline [Yang and Ramanan 2011], respectively. After obtaining the predicted view, the detected parts and attributes on this view are regarded as the detection result. Generally, compared with the first part of the detection experiment, the performance of both detection models in the second part decreases slightly. Obviously, this is mainly due to the scarcity of view information. However, while the mean APK and mean PCK of our detection model are still about 4.4% and 3.8% higher than baseline. Compared

Table II. Attribute Classification Accuracy of Baselines and Our Method

View	Bounding box	Parts of [Yang and Ramanan 2011]	Our model	Upper bound
Frontal View	64.30%	71.77%	79.90%	82.69%
Half Profile View	67.65%	73.77%	80.73%	85.46%
Profile View	68.12%	74.62%	81.80%	85.89%
Unknown	62.22%	70.89%	75.78%	80.04%

with the improvement in the first detection experiment, this indicates that our model is more stable than the baseline if the views of testing images are unknown.

For the computational time, our model costs about twice the amount of time as the baseline. Specifically, our model spends about 2.5s on processing a typical shoe image with  $500 \times 500$  pixels resolution, the model of Yang and Ramanan [2011] costs about 1.5s for each shoe image.

In the experiment of attributes prediction, by extracting the SIFT feature from circled images, detected parts of Yang and Ramanan [2011], and the ground-truth parts, we implement three baselines with multiclass linear SVM. Particularly, as huge efforts are needed to circle every testing image, we directly use the ground-truth parts to generate the bounding boxes as the circled images. Also, the third baseline with ground-truth parts reasonably indicates the upper-bound performance of attribute prediction. To compare with our result, the inputs of three baselines are twofold: the parts from the image of the specific view and the parts from the image with unknown view.

Table II presents the precision of attributes predicted by three baselines and our model. On average, if the views of the testing images are available, the precision of our model is about 14.12% and 7.42% higher than the first two baselines, and about 3.87% lower than the upper bound. If the views of parts are unknown, our prediction precision is about 13.56% and 4.89% higher than the baselines and about 4.26% lower than the upper bound. Overall, the low precision of the baseline using the bounding box may be caused by the cluttered background of testing images. Compared with the baseline using parts of Yang and Ramanan [2011], we conclude that the attributes and visual presentation of parts can enhance each other in our model. Moreover, the gap between the baseline using ground-truth parts and our model is relatively smaller than the gaps between the other baselines and our model. We suppose that the readjustment strategy that uses the consistency of parts and attributes in our detection model contributes to the reduction of the gap.

**5.1.3. Examples of Shoe Detection.** To illustrate the result of our shoe detection model, we present some testing examples of product images and daily photos returned by our detection in Figure 7. The results illustrate that our model achieves good performance, especially in the vamp part, sole part, and heel part. However, the detection performance of upper parts still needs to be improved. The imprecision of these parts is due to the significant discrepancy of the shoe's upper between high-upper shoes and low-upper shoes, such as *boots* and *sports shoes*.

According to the experiment, we claim that using the parts to represent a shoe can greatly reduce the noise caused by a cluttered background. Meanwhile, by using the constraint between attributes and parts, we can get appreciable improvement both in part detection and attribute prediction.

## 5.2. Evaluation of Shoe Retrieval

In this section, we comprehensively evaluate the performance of our retrieval system, that is, the query-specific attribute-refinement-based shoe retrieval system, by comparing with variant baselines and state-of-the-art retrieval systems. The experi-



Fig. 7. Some examples of the detected bounding boxes. In the result, we can observe that our detection model can effectively localize the different shoe parts, even when the scale and view are quite diverse or the background is cluttered.

ment result presents that our predicted attributes and our retrieval system can both contribute to the improvement of its retrieval performance.

**5.2.1. Experimental Settings.** In the retrieval experiment, 200 product images and daily photos are used as query images and the rest of the product images are used as a retrieval repository. To make the experiment convincing, four searching strategies are used. The first is implemented by using the similarities of the SIFT feature between query images and images in the retrieval repository. The second strategy is implemented by using the similarities of predicted attributes between query images and images in the retrieval repository. The third baseline, called independent attribute refinement, is implemented by setting the  $\gamma$  in Eq. (8) as zero, so that we ignore the effect of similarity between refined attributes. The last searching strategy is our proposed retrieval method, namely the query-specific attribute refinement retrieval method, that uses the jointly refined attributes for retrieval.

To evaluate the effectiveness of our detection model, we use the parts and attributes of three methods mentioned in detection experiment as the input of these four retrieval strategies. Specifically, these three data sources are the result of multiclass linear SVM using a bounding box, the result of Yang and Ramanan [2011], and the result of our model. By using the parts and attributes from one of the three aforementioned methods, we can compare the performance of the four retrieval models. By using a specific retrieval method, we can evaluate the effectiveness of our detection model. Note that, to guarantee the fairness of comparison, fivefold cross-validation is used in our experiment.

To further evaluate our retrieval model, two state-of-the-art object retrieval methods, namely the BoB with segmentation method in Arandjelovic and Zisserman [2011] and BoHB with PCA hashing strategy ( $r = 2$ ) and boundary reranking in He et al. [2012], are also conducted in this experiment. The configuration of training data for these two baselines is similar to our detection experiment. Specifically, 750 images are used as training images for the superpixel classifier in BoB with the segmentation method. The parameters of BoB with segmentation method and BoHB with PCA hashing strategy are made strictly according to the configuration in Arandjelovic and Zisserman [2011] and He et al. [2012], respectively.

**5.2.2. Evaluation Metric.** The performance of retrieval methods is evaluated by normalized Discounted Cumulative Gain (nDCG) [Siddiquie et al. 2011]. The definition of

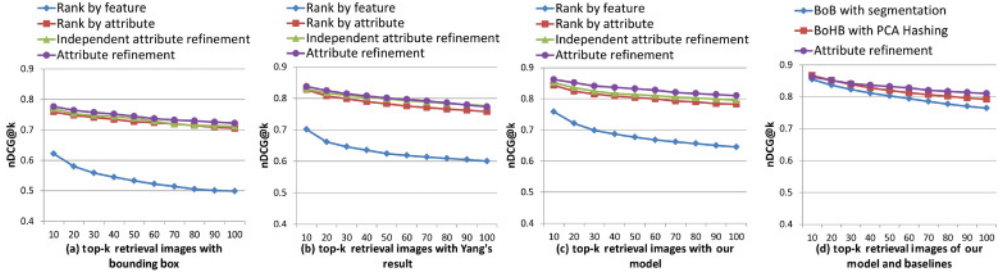


Fig. 8. The nDCG of baselines and our proposed retrieval system, namely the query-specific attribute refinement retrieval method (attribute refinement method in subfigure (c) and subfigure (d)). Ten groups of retrieval experiments are conducted by using different top-k retrieval images for evaluation. The experiment result presents that the nDCG of a specific retrieval method gradually decreases as the number of retrieval images increases, which matches our expectation as the later retrieval images usually are more irrelevant to the query image. When fixing the number of retrieval images, our retrieval method outperforms the baselines under most of the experiment configurations. Specifically, two aspects can be observed. If we use the same input data as shown in each subfigure, our query-specific attribute refinement method is superior to the other three searching strategies. If we use the same retrieval strategy, that is, the lines of the same color across subfigure (a) to subfigure (c), the retrieval method with our detected result (subfigure (c)) outperforms the methods combined with detection baselines (subfigure (a) and subfigure (b)). This observation implicitly indicates the better performance of our detection method. In subfigure (d), we can observe that our retrieval result is also comparable to state-of-the-art retrieval systems.

nDCG is

$$\text{nDCG}@k = \frac{1}{Z} \sum_{j=1}^k \frac{2^{\text{rel}(j)} - 1}{\log(1 + j)}, \quad (11)$$

where  $Z$  is used to normalize the calculated score and  $\text{rel}(\cdot)$  evaluates the similarity between the query and retrieval image in the repository.

**5.2.3. Performance Comparison.** To tune the values for  $\alpha$  and  $\gamma$  in our attribute refinement model, we try several groups of parameters by using some evaluation samples as queries and calculate the performance of our retrieval system. Generally, the performance of our retrieval system comes to maxima when the value of  $\alpha$  and  $\gamma$  is around 1.0. Therefore, we simply set  $\alpha$  and  $\gamma$  as 1 in our experiment. Figure 8 illustrates the result of the baselines and our retrieval model with different input data sources. Generally, our retrieval method outperforms the baselines under most of the configurations. Specifically, two aspects can be observed. If we fix the data source of the input as shown in Figure 8(a), Figure 8(b), and Figure 8(c), our query-specific attribute refinement retrieval system is superior to the other three searching strategies, especially the retrieval strategy with the similarity of SIFT feature. Particularly, the nDCG slightly decreases if we ignore the effect of similarity between refined attributes, indicating that the fourth term in Eq. (8) can improve the performance of attribute refinement. On the other hand, if we fix the searching strategy, that is, the lines of the same color across Figure 8(a), Figure 8(b), and Figure 8(c), the method with our detected result can achieve higher nDCG than the methods with the result of detection baselines. This observation implicitly indicates that our detection method is more accurate than the detection baselines, both in terms of parts and attributes. In practice, by using the short length of the attribute probability vector as a feature, the retrieval time is



Table III. The Average Score and Standard Deviation of User Study on Retrieval Systems

Retrieval Models		Mean Score (Standard Deviation)			
Data Source	Retrieval Strategy	Question 1	Question 2	Question 3	Question 4
Bounding Box	Rank by feat.	5.9 ± 0.54	4.3 ± 0.78	4.1 ± 1.13	7.4 ± 0.43
	Rank by attribute	7.1 ± 0.38	5.5 ± 0.69	4.2 ± 0.88	7.6 ± 0.32
	Independent attribute refinement	7.2 ± 0.35	5.4 ± 0.57	4.2 ± 1.00	7.8 ± 0.46
	Attribute refinement	7.5 ± 0.26	5.6 ± 0.59	4.4 ± 0.88	7.9 ± 0.41
Yang and Ramanan [2011]	Rank by feat.	6.5 ± 0.48	6.2 ± 0.75	5.9 ± 0.96	6.7 ± 0.44
	Rank by attribute	7.7 ± 0.32	7.3 ± 0.61	6.8 ± 1.09	6.9 ± 0.36
	Independent attribute refinement	7.9 ± 0.27	7.4 ± 0.58	7.1 ± 1.13	6.8 ± 0.36
	Attribute refinement	8.2 ± 0.31	7.7 ± 0.58	7.5 ± 0.55	6.5 ± 0.27
Our Retrieval Model	Rank by feat.	7.5 ± 0.41	7.6 ± 0.70	7.2 ± 1.04	6.5 ± 0.48
	Rank by attribute	8.2 ± 0.28	8.3 ± 0.62	7.5 ± 0.95	6.7 ± 0.37
	Independent attribute refinement	8.3 ± 0.25	8.8 ± 0.61	7.6 ± 0.81	6.9 ± 0.54
	Attribute refinement	8.6 ± 0.27	9.1 ± 0.56	7.9 ± 0.82	6.8 ± 0.39
BoB with Segmentation		8.2 ± 0.48	8.2 ± 0.39	7.3 ± 0.76	6.9 ± 0.37
BoHB with PCA Hashing		8.1 ± 0.41	8.4 ± 0.55	7.3 ± 0.60	7.2 ± 0.44

ignorable when comparing with the time for detection. Generally, our retrieval system only spends about 2.5 ~ 3s to retrieve a query image in 500 × 500 pixel resolution.

In Figure 8(d), we compare our query-specific attribute refinement method with an additional two baselines: BoB with the segmentation method, and BoHB with PCA hashing ( $r = 2$ ) and bounding reranking. Generally, the performance of our retrieval system is comparable to the two baselines. However, the performance of BoHB slightly outperforms our attribute refinement model when the number of retrieval images is less than 20. When the number of retrieval images is larger than 20, our attribute refinement model achieves a better result. Basically, these two baselines use the low-level features, namely SURF or HoG, combined with boundary information by using different strategies. This fusion strategy may greatly contribute to the improvement of retrieval performance.

However, we claim that our retrieval system has some advantages compared with BoHB and BoB. By using attribute-related features, our retrieval result is more explicit and expressive. Users can clearly observe the ranking criterion of our retrieval system. Moreover, our model can handle the nonrigid object retrieval problem, that cannot be properly solved in BoHB and BoB. Meanwhile, our system is more tolerant of the discrepancy caused by view and rotation, which is not fully solved in BoHB and BoB either.

**5.2.4. User Study.** To qualitatively evaluate our retrieval system, we conduct a user study on the demo systems to compare the retrieval result of our model and the baselines. Generally, 24 users of different careers are hired to score the results of different retrieval methods. Every query image and top-10 retrieval images returned by retrieval methods are presented as one group. Each user is required to view 50 groups of retrieval results and answer the following questions by scoring each group from 0 to 10. Then, the average scores are calculated on the 50 groups of samples, and the mean score of 24 users with standard deviation is used to evaluate the performance of retrieval systems. The specific questions for this user study include the following.

—*Question 1.* Is the retrieval result relevant to the query image in terms of attributes? Please score them according to heel shape, heel height, decoration, sole shape, tie style, topline, toe shape, vamp style, upper style, and closed mode (1 score for each attribute, 10 scores in total).

- Question 2.* Does the retrieval result preserve the general style of query image? (10–8 scores: fully preserve; 7–5 scores: partially preserve; 4–3 scores: slightly preserve; 2–0 scores: does not preserve.)
- Question 3.* Does the retrieval result meet your performance expectation? (10–9 scores: exceed; 8–7 scores: meet; 6–5 scores: partially meet; 4–3 scores: moderate; 2–0 scores: does not meet.)
- Question 4.* What do you think of the response time of the retrieval system? (10–9 scores: very quick; 8–7 scores: quick; 6–5 scores: acceptable; 4–3 scores: needs improvement; 2–0 scores: unbearable.)

The mean score and standard deviation of retrieval systems are presented in Table III. Obviously, Question 1 and Question 4 are more objective, while Question 2 and Question 3 are more subjective, which can be observed from the standard deviation of user scores.

Generally, our retrieval system achieves a higher score than most of the baselines in terms of quality. The lower standard deviation of our system may also indicate the stability of our retrieval result. Specifically, Question 1 implicitly represents the accuracy of attribute prediction. It is interesting to point out that the score of those retrieval systems coincides with the accuracy of attribute prediction in Table II. This observation further reveals the efficacy of our detection model. Compared with BoB and BoHB, we claim that our detector can extract more distinctive features of query images. In Question 2, most users consider that our retrieval result can better preserve the general style of query images than most of the baselines. This may represent the operability of the attribute-based retrieval strategy. Moreover, it should be noticed that our retrieval system achieves higher scores than BoB and BoHB in this question. However, the overall score of Question 3 is relatively low, which may be due to the high response time of our system. This is also represented in Question 4.

## 6. CONCLUSIONS AND FUTURE WORK

In this article, we propose a Circle & Search shoe retrieval system that searches the most similar shoes in a repository by circling the shoe in daily photos as a query. Our system contains two phases, namely, the shoe detection phase and the shoe retrieval phase. In the phase of shoe detection, by using the consistency between the shoe parts and semantic attributes, the detector can simultaneously estimate the positions of shoe parts and the values of shoe attributes. During the retrieval phase, the correlations between attributes are analysed and used to refine the predicted attribute values. Then, the refined attribute values are used to rank all the shoes in a query-specific way by computing the attribute distances. In the experiment, a large-scale shoe dataset is collected and the experiment result on this dataset well demonstrated the effectiveness of our Circle & Search system. Compared with other retrieval systems, the retrieval result of our system is more expressive due to the semantic meaning of the retrieval feature. Moreover, the retrieval problem of nonrigid objects can also be solved by our system due to the part-based property. Last but not least, our system is more tolerant of the discrepancy caused by view changing and rotation.

By defining the tree structures and part-attribute relations, we claim that our system can be extended to other domains if the attributes of the domain object have local influence on the parts. Besides the domain extension, the automatic learning algorithms for the structure of detectors and the part-attribute relations will be another major task in our future work.

## REFERENCES

Relja Arandjelovic and Andrew Zisserman. 2011. Smooth object retrieval using a bag of boundaries. In *Proceedings of the International Conference on Computer Vision (ICCV'11)*. IEEE, 375–382.

- Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. 2010. Automatic attribute discovery and characterization from noisy web data. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*. Springer, 663–676.
- Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. 2011. Describing people: A poselet-based approach to attribute classification. In *Proceedings of the International Conference on Computer Vision (ICCV'11)*. IEEE, 1543–1550.
- Huizhong Chen, Andrew Gallagher, and Bernd Girod. 2012. Describing clothing by semantic attributes. In *Proceedings of the European Conference on Computer Vision (ECCV'12)*. Springer, 609–623.
- Navneet Dalal and Bill Triggs. 2005. INRIA person dataset. <http://pascal.inrialpes.fr/data/human>.
- Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. 1778–1785.
- Pedro Felzenszwalb and Daniel Huttenlocher. 2004. Distance transforms of sampled functions. Tech. rep., Department of Computing and Information Science, Cornell. <http://www.cs.cornell.edu/~dph/papers/dt.pdf>.
- Pedro Felzenszwalb, Ross B. Girshick, David Mcallester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intel.* 32, 9, 1627–1645.
- Vittorio Ferrari and Andrew Zisserman. 2008. Learning visual attributes. In *Proceedings of the Neural Information Processing Systems Conference (NIPS'08)*.
- Junfeng He, Jinyuan Feng, Xianglong Liu, Tao Cheng, Tai-Hsu Lin, Hyunjin Chung, and Shih-Fu Chang. 2012. Mobile product search with bag of hash bits and boundary reranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. 3005–3012.
- Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2011. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intel.* 33, 1, 117–128.
- Hongwen Kang, Martial Hebert, Alexei A. Efros, and Takeo Kanade. 2012. Connecting missing links: Object discovery from sparse observations using 5 million product images. In *Proceedings of the European Conference on Computer Vision (ECCV'12)*. Springer, 794–807.
- Adriana Kovashka, Devi Parikh, and Kristen Grauman. 2012. WhittleSearch: Image search with relative attribute feedback. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. 2973–2980.
- Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. 2009. Attribute and simile classifiers for face verification. In *Proceedings of the International Conference on Computer Vision (ICCV'09)*. IEEE, 365–372.
- Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. 2012. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. 3330–3337.
- Shiyang Lu, Tao Mei, Jingdong Wang, Jian Zhang, Zhiyong Wang, David Dagan Feng, Jian-Tao Sun, and Shipeng Li. 2012. Browse-to-search. In *Proceedings of the 20<sup>th</sup> ACM International Conference on Multimedia (ACM-MM'12)*. ACM Press, New York, 1323–1324.
- Devi Parikh and Kristen Grauman. 2011. Interactively building a discriminative vocabulary of nameable attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 1681–1688.
- Xiaohui Shen, Zhe Lin, Jonathan Brandt, and Ying Wu. 2012. Mobile product image search by automatic query object extraction. In *Proceedings of the European Conference on Computer Vision (ECCV'12)*. Springer, 114–127.
- Behjat Siddiquie, Rogerio Schmidt Feris, and Larry S. Davis. 2011. Image ranking and retrieval based on multi-attribute queries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 801–808.
- Yang Wang and Greg Mori. 2010. A discriminative latent model of object classes and attributes. In *Proceedings of the European Conference on Computer Vision (ECCV'10)*. Springer, 155–168.
- Yi Yang and Deva Ramanan. 2011. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. 1385–1392.

Received August 2013; revised April 2014; accepted April 2014