WhittleSearch: Image Search with Relative Attribute Feedback

Adriana Kovashka[1]
[1]University of Texas at Austin
[1]{adriana, grauman}@cs.utexas.edu

Devi Parikh[2]
[2]Toyota Technological Institute Chicago (TTIC)
[2]dparikh@ttic.edu

Kristen Grauman[1]

Abstract

We propose a novel mode of feedback for image search, where a user describes which properties of exemplar images should be adjusted in order to more closely match his/her mental model of the image(s) sought. For example, perusing image results for a query "black shoes", the user might state, "Show me shoe images like these, but sportier." Offline, our approach first learns a set of ranking functions, each of which predicts the relative strength of a nameable attribute in an image ('sportiness', 'furriness', etc.). At query time, the system presents an initial set of reference images, and the user selects among them to provide relative attribute feedback. Using the resulting constraints in the multi-dimensional attribute space, our method updates its relevance function and re-ranks the pool of images. This procedure iterates using the accumulated constraints until the top ranked images are acceptably close to the user's envisioned target. In this way, our approach allows a user to efficiently "whittle away" irrelevant portions of the visual feature space, using semantic language to precisely communicate her preferences to the system. We demonstrate the technique for refining image search for people, products, and scenes, and show it outperforms traditional binary relevance feedback in terms of search speed and accuracy.

1. Introduction

Image search entails retrieving those images in a collection that meet a user's needs, whether using a keyword or an image itself as the query. It has great potential for a range of applications where users can envision content of interest, but need the system's help to find it: graphic designers seeking illustrations, computer users searching for favorite personal photos, or shoppers browsing for products online.

In spite of decades of attention, the problem remains challenging. Keywords alone are clearly not enough; even if all existing images were tagged to enable keyword search, it is infeasible to pre-assign tags sufficient to satisfy any future query a user may dream up. Indeed, vision algorithms are necessary to further parse the *content* of images for many search tasks; advances in image descriptors, learning algorithms, and large-scale indexing have all had impact in recent years. Nonetheless, the well-known and frustrating "semantic gap" between low-level visual cues and the high-level intent of a user remains, making it difficult for people to predict the behavior of content-based search systems.

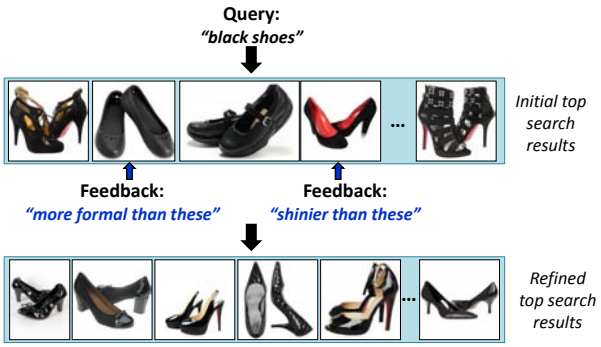


Figure 1. Main idea: Allow users to give relative attribute feedback on reference images to refine their image search.

The key to overcoming the gap appears to be *interactive* search techniques that allow a user to iteratively refine the results retrieved by the system [3, 14, 22, 30, 7, 29]. The basic idea is to show the user candidate results, obtain feedback, and adapt the system's relevance ranking function accordingly. However, existing image search methods provide only a narrow channel of feedback to the system. Typically, a user refines the retrieved images via binary feedback on exemplars deemed "relevant" or "irrelevant" [14, 3, 22, 30, 7], or else attempts to tune system parameters such as weights on a small set of low-level features (e.g., texture, color, edges) [8, 16]. The latter is clearly a burden for a user who likely cannot understand the inner workings of the algorithm. The former feedback is more natural to supply, yet it leaves the system to infer *what about those images* the user found relevant or irrelevant, and therefore can be slow to converge on the user's target in practice.

In light of these shortcomings, we propose a novel mode of feedback where a user directly describes how high-level properties of exemplar images should be adjusted in order to more closely match his/her envisioned target images. For example, when conducting a query on a shopping website,

the user might state: "I want shoes like these, but *more formal*." When browsing images of potential dates on a dating website, he can say: "I am interested in someone who looks like this, but with *longer hair* and *more smiling*." When searching for stock photos to fit an ad, he might say: "I need a scene *similarly bright* as this one and *more urban* than that one." See Fig. 1. In this way, rather than simply state which images are (ir)relevant, the user employs semantic terms to say *how* they are so. We expect such feedback will enable the system to more closely match the user's mental model of the desired content, and with less total interaction effort.

Briefly, the approach works as follows. Offline, we first learn a set of ranking functions, each of which predicts the relative strength of a nameable attribute in an image (e.g., the degree of 'shininess', 'furriness', etc.). At query time, the system presents an initial set of reference images, and the user selects among them to provide relative attribute feedback. Using the resulting constraints in the multi-dimensional attribute space, we update the system's relevance function, re-rank the pool of images, and display the top-ranked set to the user. This procedure iterates using the accumulated constraints until the top ranked images are acceptably close to the user's target. We call the approach *WhittleSearch*, since it allows users to "whittle away" irrelevant portions of the visual feature space via precise, intuitive statements of their attribute preferences.

We demonstrate WhittleSearch for retrieval tasks with people, product, and scene images. We show it refines search results more effectively than traditional binary relevance feedback, and often with less total user interaction. Furthermore, we explore the tradeoffs in learning relative attributes using either top-down category information, absolute attribute judgments, or individual image-level comparisons from human annotators. Our main contribution is to widen human-machine communication for interactive image search in a new and practical way by allowing users to communicate their preferences very precisely.

2. Related Work

We review related work on attributes for image search and recognition, interactive feedback, and using comparative information for visual learning.

Human-nameable *semantic concepts* or *attributes* are often used in the multimedia community to build intermediate representations for image retrieval [24, 21, 17, 29, 5, 27]. The idea is to learn classifiers to predict the presence of various high-level semantic concepts from a lexicon—such as objects, locations, activity types, or properties—and then perform retrieval in the space of those predicted concepts. The same attributes can also be used to pose queries in semantic terms [12, 23]. While it is known that attributes can provide a richer representation than raw low-level image features, no previous work considers attributes as a handle for user feedback, as we propose.

Attributes have also gathered interest in the object recognition community recently [15, 6, 13, 28, 2]. Since attributes are often shared among object categories (e.g., 'made of wood', 'plastic', 'has wheels'), they are amenable to a number of interesting tasks, such as zero-shot learning from category descriptions [15], describing unfamiliar objects [6], or categorizing with a 20-questions game [2]. Attributes are largely assumed to be categorical properties. However, recent work introduces the concept of *relative attributes* as ranking functions, and shows their impact for zero-shot learning and description [20]. We also explore relative attributes, but in the distinct context of feedback for image search; further, we generalize the class-based training procedure used in [20] to exploit human-generated relative comparisons between image exemplars.

Relevance feedback has long been used to improve interactive image search [14, 3, 22, 25, 7]; see the survey of [30] for a broader overview than space permits here. The idea is to tailor the system's ranking function to the current user, based on his (usually iterative) feedback on the relevance of selected exemplar images. This injects subjectivity into the model, implicitly guiding the search engine to pay attention to certain low-level visual cues more than others. To make the most of user feedback, some methods actively select exemplars most in need of feedback (e.g., [3, 7]). Like existing interactive methods, our approach aims to elicit a specific user's target visual concept. However, while prior work restricts input to the form "A is relevant, B is not" or "C is more relevant than D", our approach allows users to comment precisely on what is missing from the current set of results. We show that this richer form of feedback can lead to more effective refinement.

We aggregate relative constraints placed on exemplars to improve image search. Comparative information can also be used to train object recognition systems, by stating similarity between object categories [26], comparing their attributes [20], or explaining attributes that make them different [4]. In this work, we empower the user to better communicate his target visual concept via comparative descriptions to exemplar images, resulting in improved user experience (high quality results and less user effort) during search.

3. Approach

Our approach allows a user to iteratively refine the search using feedback on attributes. The user initializes the search with some keywords—either the name of the general class of interest ("shoes") or some multi-attribute query ("black high-heeled shoes")—and our system's job is to help refine from there. If no such initialization is possible, we simply begin with a random set of top-ranked images for feedback. The top-ranked images are then displayed to the user, and the feedback-refinement loop begins.

Throughout, let $\mathcal{P} = \{I_1, \ldots, I_N\}$ refer to the pool of N database images that are ranked by the system using its current scoring function $S_t : I \rightarrow \mathbb{R}$, where t denotes the iteration of refinement. The scoring function is trained using all accumulated feedback from iterations $1, \ldots, t-1$, and it supplies an ordering (possibly partial) on the images in $\mathcal{P}$. At each iteration, the top $K < N$ ranked images $\mathcal{T}_t = \{I_{t1}, \ldots, I_{tK}\} \subseteq \mathcal{P}$ are displayed to the user for further feedback, where $S_t(I_{t1}) \geq S_t(I_{t2}) \geq \cdots \geq S_t(I_{tK})$. A user then gives feedback of his choosing on any or all of the K refined results in $\mathcal{T}_t$. We refer to $\mathcal{T}_t$ interchangeably as the *reference set* or *top-ranked set*.

In the following, we first describe a traditional binary relevance feedback model (Sec. 3.1), since it will serve as a strong baseline to which to compare our approach. Then we introduce the proposed new mode of relative attribute feedback (Sec. 3.2). Finally, we extend the idea to accommodate both forms of input in a hybrid approach (Sec. 3.3).

3.1. Background: Binary Relevance Feedback

In a binary relevance feedback model, the user identifies a set of relevant images $\mathcal{R}$ and a set of irrelevant images $\bar{\mathcal{R}}$ among the current reference set $\mathcal{T}_t$. In this case, the scoring function S_t^b is a classifier (or some other statistical model), and the binary feedback essentially supplies additional positive and negative training examples to enhance that classifier. That is, the scoring function S_{t+1}^b is trained with the data that trained S_t^b *plus* the images in $\mathcal{R}$ labeled as positive instances and the images in $\bar{\mathcal{R}}$ labeled as negative instances.

We use a binary feedback baseline that is intended to represent traditional approaches such as [3, 7, 22, 25]. While a variety of classifiers have been explored in previous systems, we employ a support vector machine (SVM) classifier for the binary feedback model due to its strong performance in practice. Thus, the scoring function for binary feedback is $S^b(\boldsymbol{x}_j) = \boldsymbol{w}_b \boldsymbol{x}_j^T + b$, where $\boldsymbol{w}_b$, b are the SVM parameters and $\boldsymbol{x}_j$ denotes the visual features extracted from image I_j, to be defined below.

3.2. Relative Attribute Feedback

Suppose we have a vocabulary of M attributes $A = \{a_m\}$, which may be generic or domain-specific for the image search problem of interest. For example, a domain-specific vocabulary for shoe shopping could contain attributes such as "shininess", "heel height", "colorfulness", etc., whereas for scene descriptions it could contain attributes like "openness", "naturalness", "depth". While we assume this vocabulary is given, recent work suggests it may also be discoverable automatically [1, 19].

To leverage the proposed relative attribute feedback, our method requires attribute predictions on all images and a means to aggregate cumulative constraints on individual attributes, as we describe in the following.

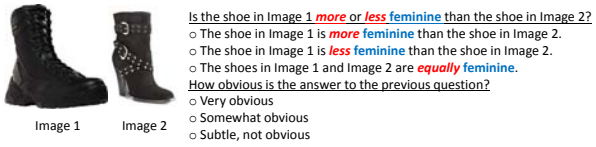


Figure 2. Interface for image-level relative attribute annotations.

3.2.1 Learning to predict relative attributes

Typically semantic visual attributes are learned as categories: a given image either exhibits the concept or it does not, and so a classification approach to predict attribute presence is sufficient [21, 17, 29, 15, 6, 13, 28, 5]. In contrast, to express feedback in the form sketched above, we require *relative* attribute models that can predict the degree to which an attribute is present. Therefore, we first learn a *ranking function* for each attribute in the given vocabulary.

For each attribute a_m, we obtain supervision on a set of image pairs (i, j) in the training set $\mathcal{I}$. We ask human annotators to judge whether that attribute has stronger presence in image i or j, or if it is equally strong in both. Such judgments can be subtle, so on each pair we collect 5 redundant responses from multiple annotators on Mechanical Turk (MTurk); see Fig. 2. To distill reliable relative constraints for training, we use only those for which most labelers agree. This yields a set of ordered image pairs $O_m = \{(i, j)\}$ and a set of un-ordered pairs $E_m = \{(i, j)\}$ such that $(i, j) \in O_m \implies i \succ j$, *i.e.* image i has stronger presence of attribute a_m than j, and $(i, j) \in E_m \implies i \sim j$, *i.e.* i and j have equivalent strengths of a_m.

We stress the design for constraint collection: rather than ask annotators to give an *absolute* score reflecting how much the attribute m is present, we instead ask them to make *comparative* judgements on two exemplars at a time. This is both more natural for an individual annotator, and also permits seamless integration of the supervision from many annotators, each of whom may have a different internal "calibration" for the attribute strengths.

Next, to learn an attribute's ranking function, we employ the large-margin formulation of Joachims [9], which was originally shown for ranking web pages based on click-through data, and recently used for relative attribute learning [20]. Suppose each image I_i is represented in $\mathbb{R}^d$ by a feature vector $\boldsymbol{x}_i$ (we use color and Gist, see below). We aim to learn M ranking functions, one per attribute:

$$r_m(\boldsymbol{x}) = \boldsymbol{w}_m^T \boldsymbol{x}_i, \tag{1}$$

for $m = 1, \ldots, M$, such that the maximum number of the following constraints is satisfied:

$$\forall (i, j) \in O_m : \boldsymbol{w}_m^T \boldsymbol{x}_i > \boldsymbol{w}_m^T \boldsymbol{x}_j \tag{2}$$

$$\forall (i, j) \in E_m : \boldsymbol{w}_m^T \boldsymbol{x}_i = \boldsymbol{w}_m^T \boldsymbol{x}_j. \tag{3}$$

Joachims' algorithm approximates this NP hard problem by introducing (1) a regularization term that prefers a wide

margin between the ranks assigned to the closest pair of training instances, and (2) slack variables ξ_{ij}, γ_{ij} on the constraints, yielding the following objective [9]:

$$\begin{aligned} \text{minimize} \quad & \left(\frac{1}{2}||\boldsymbol{w}_{\boldsymbol{m}}^T||_2^2 + C\left(\sum \xi_{ij}^2 + \sum \gamma_{ij}^2\right)\right) \quad (4) \\ s.t. \quad & \boldsymbol{w}_m^T \boldsymbol{x}_i \geq \boldsymbol{w}_m^T \boldsymbol{x}_j + 1 - \xi_{ij}; \quad \forall (i,j) \in O_m \\ & |\boldsymbol{w}_m^T \boldsymbol{x}_i - \boldsymbol{w}_m^T \boldsymbol{x}_j| \leq \gamma_{ij}; \quad \forall (i,j) \in E_m \\ & \xi_{ij} \geq 0; \gamma_{ij} \geq 0, \end{aligned}$$

where C is a constant penalty. The objective is reminiscent of standard SVM training (and is solvable using similar decomposition algorithms), except the linear constraints enforce relative orderings rather than labels. The method is kernelizable. We use Joachims' SVMRank code [10].

Having trained M such functions, we are then equipped to predict the extent to which each attribute is present in any novel image, by applying the learned functions $r_1, \ldots, r_M$ to its image descriptor $\boldsymbol{x}$. Note that this training is a one-time process done before any search query or feedback is issued, and the data $\mathcal{I}$ used for training attribute rankers is not to be confused with our database pool $\mathcal{P}$.

Whereas Parikh and Grauman [20] propose generating supervision for relative attributes from top-down category comparisons ("person X is (always) more smiley than person Y"), our approach extends the learning process to incorporate *image-level* relative comparisons ("image A exhibits more smiling than image B"). While training from category-level comparisons is clearly more expedient, we find that image-level supervision is important in order to reliably capture those attributes that do not closely follow category boundaries. The 'smiling' attribute is a good example of this contrast, since a given person (the category) need not be smiling to an equal degree in each of his/her photos. In fact, our user studies on MTurk show that category-level relationships violate 23% of the image-level relationships specified by human subjects for the 'smiling' attribute. In the results section, we detail related human studies analyzing the benefits of instance-level comparisons.

3.2.2 Updating the scoring function from feedback

With the ranking functions learned above, we can now map any image from $\mathcal{P}$ into an M-dimensional space, where each dimension corresponds to the relative rank prediction for one attribute. It is in this feature space we propose to handle query refinement from a user's feedback.

A user of the system has a mental model of the target visual content he seeks. To refine the current search results, he surveys the K top-ranked images in $\mathcal{T}_t$, and uses some of them as reference images with which to better express his envisioned optimal result. These constraints are of the form "What I want is more/less/similarly m than image I_{t_r}", where m is an attribute name, and I_{t_r} is an image in $\mathcal{T}_t$

Figure 3. A toy example illustrating the intersection of relative constraints with $M = 2$ attributes. The images are plotted on the axes for both attributes. The space of images that satisfy each constraint are marked in a different color. The region satisfying all constraints is marked with a black dashed line. In this case, there is only one image in it (outlined in black). Best viewed in color.

(the subscript t_r denotes it is a $r$eference image at iteration t). These relative constraints are given for some combination of image(s) and attribute(s) of the user's choosing.

The conjunction of all such user feedback statements gives us a set of constraints for updating the scoring function. For all statements of the form "I want images exhibiting more of attribute m than reference image I_{t_r}", our updated attribute-based scoring function S_{t+1}^a should satisfy:

$$\begin{aligned} & S_{t+1}^a(I_i) > S_{t+1}^a(I_j), \quad \forall I_i, I_j \in \mathcal{P} \quad (5) \\ & s.t. \;\; r_m(\boldsymbol{x}_i) > r_m(\boldsymbol{x}_{t_r}), \;\; r_m(\boldsymbol{x}_j) \leq r_m(\boldsymbol{x}_{t_r}), \end{aligned}$$

where as before $\boldsymbol{x}_i$ denotes the image descriptor for image I_i used to predict its relative attributes. This simply reflects that images having more of the desired property m than the displayed reference image are better than those that do not. We stress that the relative attribute values on all images are *predicted* using the learned function r_m (as opposed to having ground truth on the attribute strengths in each image).

Similarly, for all statements of the form "I want images exhibiting less of attribute m than I_{t_r}", our updated scoring function should satisfy:

$$\begin{aligned} & S_{t+1}^a(I_i) > S_{t+1}^a(I_j), \quad \forall I_i, I_j \in \mathcal{P} \quad (6) \\ & s.t. \;\; r_m(\boldsymbol{x}_i) < r_m(\boldsymbol{x}_{t_r}), \;\; r_m(\boldsymbol{x}_j) \geq r_m(\boldsymbol{x}_{t_r}) \end{aligned}$$

For all statements of the form, "I want images that are similar in terms of attribute m to I_{t_r}", the constraints are:

$$\begin{aligned} & S_{t+1}^a(I_i) > S_{t+1}^a(I_j), \quad \forall I_i, I_j \in \mathcal{P} \quad (7) \\ & s.t. \;\; (r_m(\boldsymbol{x}_{t_r}) - \epsilon) \leq r_m(\boldsymbol{x}_i) \leq (r_m(\boldsymbol{x}_{t_r}) + \epsilon), \\ & r_m(\boldsymbol{x}_j) < r_m(\boldsymbol{x}_{t_r}) - \epsilon \;\text{ or }\; r_m(\boldsymbol{x}_j) > r_m(\boldsymbol{x}_{t_r}) + \epsilon, \end{aligned}$$

where ϵ is a constant specifying the distance in relative attribute space at which instances are considered dissimilar. Note that these similarity constraints differ from binary feedback, in that they single out an individual attribute. Our current implementation focuses on the two relative forms of feedback (more, less).

Each of the above carves out a relevant region of the M-dimensional attribute feature space, whittling away images not meeting the user's requirements. We combine all such constraints to adapt the scoring function from S_t^a to S_{t+1}^a. That is, we take the intersection of all F feedback constraints thus far to identify the set of top ranked images, for which $S_{t+1}^a(I_i) = F$. Those satisfying all but one constraint receive score $F-1$, and so on, until images satisfying no constraints receive the score 0. See Fig. 3. Even if no images satisfy all constraints, we can produce a ranking.

One could alternatively learn a ranking function for S_{t+1}^a using these constraints within the large-margin objective above; however, for the sake of determining the *ordering* on the data—as is needed to refine the top ranked results—its behavior would be equivalent. Thus we take this purely set-logic approach, as it is less costly.

We stress that the proposed form of relative attribute feedback refines the search in ways that a straightforward multi-attribute query cannot. That is, if a user were to simply state the attribute labels of interest ("show me black shoes that are shiny and high-heeled"), one can easily retrieve the images whose attribute predictions meet those criteria. However, since the user's description is in absolute terms, it cannot change based on the retrieved images. In contrast, with access to relative attributes as a mode of communication, for every new set of reference images returned by the system, the user can further refine his description.

Having completed a cycle of feedback and refinement, the method repeats the loop, accepting any additional feedback from the user on the newly top-ranked images. In practice, the system can either iterate until the user's target image is found, or else until his "budget" of interaction effort is expended.

3.3. Hybrid Feedback Approach

Thus far we have considered each form of feedback in isolation. However, they have complementary strengths: when reference images are nearly on target (or completely wrong in all aspects), the user may be best served by providing a simple binary relevance label. Meanwhile, when a reference image is lacking only in certain describable properties, he may be better served by the relative attribute feedback. Thus, it is natural to combine the two modalities, allowing a mix of feedback types at any iteration.

To this end, one can consider a learned hybrid scoring function. The basic idea is to learn a ranking function S_{t+1}^h that unifies both forms of constraints. Recall that $\mathcal{R}$ and $\bar{\mathcal{R}}$ denote the sets of reference images for which the user has given positive and negative binary feedback, respectively. Let $\mathcal{F}_k \subset \mathcal{P}$ denote the subset of images satisfying k of the relative attribute feedback constraints, for $k = 0, \ldots, F$. We define a set of ordered image pairs

$$O_s = \{\{\mathcal{R} \times \bar{\mathcal{R}}\} \cup \{\mathcal{F}_F \times \mathcal{F}_{F-1}\} \cup \cdots \cup \{\mathcal{F}_1 \times \mathcal{F}_0\}\},$$

where $\times$ denotes the Cartesian product. This set O_s reflects all the desired ranking preferences—that relevant images be ranked higher than irrelevant ones, and that images satisfying more relative attribute preferences be ranked higher than those satisfying fewer. As equivalence constraints, we have:

$$E_s = \{\{\mathcal{F}_F \times \mathcal{F}_F\} \cup \cdots \cup \{\mathcal{F}_1 \times \mathcal{F}_1\}\}, \quad (8)$$

reflecting that images satisfying the same amount of relative feedback should be ranked equally high. Note that the subscript s in O_s and E_s distinguishes the sets from those indexed by m above, which were used to train relative attribute ranking functions in Sec. 3.2.1.

Using training constraints O_s and E_s we can learn a function that predicts *relative image relevance* for the current user with the large-margin objective in Eqn. 4. The result is a parameter vector $\boldsymbol{w}_s$ that serves as the hybrid scoring function S_{t+1}^h. We randomly sample from pairs in O_s and E_s to generate representative constraints for training.

To recap the approach section, we now have three forms of scoring functions to be used for refining search results: traditional binary feedback (S^b), relative attribute feedback (S^a), and a hybrid that unifies the two (S^h).

4. Experimental Results

We analyze how the proposed relative attribute feedback can enhance image search compared to classic binary feedback, and study what factors influence their behavior.

4.1. Experimental Design

Datasets We use three datasets: the **Shoes** from the Attribute Discovery Dataset [1], the Public Figures dataset of human faces [13] (**PubFig**), and the Outdoor Scene Recognition dataset of natural scenes [18] (**OSR**). They validate our approach in diverse domains of interest: finding products, people, and scenes. The Shoes data contains 14,658 shoe images from like.com. We augment the data with relative attributes—'pointy-at-the-front', 'open', 'bright-in-color', 'covered-with-ornaments', 'shiny', 'high-at-the-heel', 'long-on-the-leg', 'formal', 'sporty', and 'feminine'. For PubFig we use the subset from [20], which contains 772 images from 8 people and 11 attributes ('young', 'round face', etc.). OSR consists of 2,688 images from 8 categories and 6 attributes ('openness', 'perspective') [20].

For image features $\boldsymbol{x}$, we use Gist [18] and color histograms for Shoes and PubFig, and Gist alone for OSR.

Methodology For each query we select a random *target image* and score how well the search results match that target after feedback. This target stands in for a user's mental model; it allows us to prompt multiple subjects for feedback on a well-defined visual concept, and to precisely judge how accurate results are. This part of our methodology is key to ensure consistent data collection and formal evaluation.

We use two metrics: (1) the ultimate *rank* assigned to the user's target image and (2) the *correlation* between the full ranking computed by S_t and a ground truth ranking that reflects the perceived relevance of all images in $\mathcal{P}$. Lower ranks are better, since that means the target image appears among the top-ranked search results presented to the user. Our method often produces a partial ordering where multiple images satisfy the same number of constraints; thus, we rank all n images that satisfy all constraints as 1, then all images in the next equivalence class as $n+1$, as so on.

The correlation metric captures not only where the target itself ranks, but also how similar to the target the other top-ranked images are. We form the ground truth relevance ranking by sorting all images in $\mathcal{P}$ by their distance to the given target. To ensure this distance reflects *perceived* relevance, we learn a metric based on human judgments. Specifically, we show 750 triplets of images (i, j, k) from each dataset to 7 MTurk human subjects, and ask whether images i and j are more similar, or images i and k. Using their responses, we learn a linear combination of the image and attribute feature spaces that respects these constraints via [9]. Our ground truth rankings thus mimic human perception of image similarity. To score correlation, we use Normalized Discounted Cumulative Gain at top K (NDCG@K) [11], which scores how well the predicted ranking and the ground truth ranking agree, while emphasizing items ranked higher. We use $K = 50$, based on the number of images visible on a page of image search results.

Feedback generation We use MTurk to gather human feedback for our method and the binary feedback baseline. We pair each target image with 16 reference images. For our method we ask, "Is the target image more or less ⟨attribute name⟩ than the reference image?" (for each ⟨attribute name⟩), while for the baseline we ask, "Is the target image similar to or dissimilar from the reference image?" We also request a confidence level for each answer; see Fig. 2. We get each pair labeled by 5 workers and use majority voting to reduce noise. When sampling from these constraints to impose feedback, we take those rated most obvious on average by the workers.

Since the human annotations are costly, for certain studies we generate feedback automatically. For relative constraints, we randomly sample constraints based on the predicted relative attribute values, checking how the target image relates to the reference images. (For example, if the target's predicted 'shininess' is 0.5 and some reference image's 'shininess' is 0.6, then a valid constraint is that the target is "less shiny" than that reference image.) For binary feedback, we analogously sample positive/negative reference examples based on their image feature distance to the true target; we take the top and bottom quartile, respectively. When scoring rank, we add Gaussian noise to the predicted attributes (for our method) and the SVM outputs

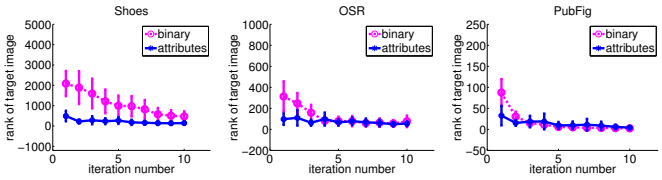


Figure 4. Iteration experiments on the three datasets. Our method often converges on the target image more rapidly.

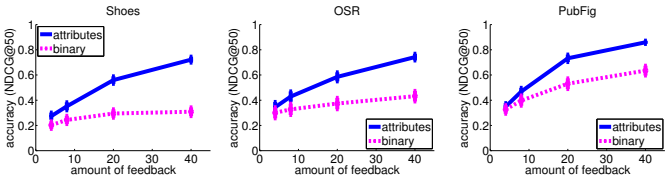


Figure 5. Ranking accuracy as a function of amount of feedback. While more feedback enhances both methods, the proposed attribute feedback yields faster gains per unit of feedback.

(for the baseline), to coarsely mimic human uncertainty in constraint generation. The automatically generated feedback is a good proxy for human feedback since the relative predictions are explicitly trained to represent human judgments. It allows us to test performance on a larger scale.

4.2. Feedback Results

Impact of iterative feedback First we examine how the rank of the target image improves as the methods iterate. Both methods start with the same random set of 16 reference images, and then iteratively obtain 8 automatically generated feedback constraints, each time re-scoring the data to revise the top reference images (using S_t^a and S_t^b for our method and the binary baseline, respectively).[1]

Fig. 4 shows the results, for 50 such queries. Our method outperforms the binary feedback baseline for all datasets, more rapidly converging on a top rank for the target image. On PubFig our advantage is slight, however. We suspect this is due to the strong category-based nature of the PubFig data, which makes it more amenable to binary feedback; adding positive labels on exemplars of the same person as the target image is quite effective. In contrast, on scenes and shoes where images have more fluid category boundaries, our advantage is much stronger. The searches tend to stabilize after 2-10 rounds of feedback. The run-times for our method and the baseline are similar.

Impact of amount of feedback Next we analyze the impact of the amount of feedback, using automatically generated constraints. Fig. 5 shows the rank correlation results for 100 queries. These curves show the quality of all top-ranked results as a function of the amount of feedback given in a single iteration. Recall that a round of feedback consists of a relative attribute constraint or a binary label on one image, for our method or the baseline, respectively. For

[1]To ensure new feedback accumulates per iteration, we do not allow either method to reuse a reference image.

Dataset-Method	Near	Far	Near+Far	Mid
Shoes-Attribute	.39	.29	**.40**	.38
Shoes-Binary	.12	.05	**.27**	.06
PubFig-Attributes	**.60**	.41	.58	.52
PubFig-Binary	.39	.21	**.64**	.15
OSR-Attributes	**.53**	.27	.52	.40
OSR-Binary	.18	.18	**.32**	.11

Figure 6. Ranking accuracy (NDCG@50 scores) as we vary the type of reference images available for feedback.

all datasets, both methods clearly improve with more feedback. However, the precision enabled by our attribute feedback yields a greater "bang for the buck"—higher accuracy for fewer feedback constraints. The result is intuitive, since with our method users can better express *what about* the reference image is (ir)relevant to them, whereas with binary feedback they cannot.

A multi-attribute query baseline that ranks images by how many binary attributes they share with the target image achieves NDCG scores 40% weaker on average than our method when using 40 feedback constraints. This result supports our claim that binary attribute search lacks the expressiveness of iterative relative attribute feedback.

Impact of reference images The results thus far assume that the initial reference images are randomly selected, which is appropriate when the search cannot be initialized with keyword search. We are interested in understanding the impact of the *types* of reference images available for feedback. Thus, we next control the pool of reference images to consist of one of four types: "near", meaning images close to the target image, "far", meaning images far from the target, "near+far", meaning a 50-50 mix of both, and "mid", meaning neither near nor far from the target. Nearness is judged in the Gist/color feature space.

Fig. 6 shows the resulting accuracies, for all types and all datasets using 100 queries and automatic feedback. Both methods generally do well with "near+far" reference images, which makes sense. For attributes, we expect useful feedback to entail statements about images that are similar to the target overall, but lack some attribute. Meanwhile, for binary feedback, we expect useful feedback to contain a mix of good positives and negatives to train the classifier. We further see that attribute feedback also does fairly well with only "near" reference images; intuitively, it may be difficult to meaningfully constrain precise attribute differences on an image much too dissimilar from the target.

Ranking accuracy with human-given feedback Having analyzed in detail the key performance aspects with automatically generated feedback, now we report results using human-generated feedback. Fig. 8 shows the ranking correlation for both methods on 16 queries per dataset after one round of 8 feedback statements. Attribute feedback largely outperforms binary feedback, and does similarly well on OSR. One possible reason for the scenes being less amenable to attribute feedback is that humans seem to have more confusion interpreting the attribute meanings (e.g., "amount of perspective" on a scene is less intuitive than "shininess" on shoes).

Figure 7. Example iterative search result with attribute feedback.

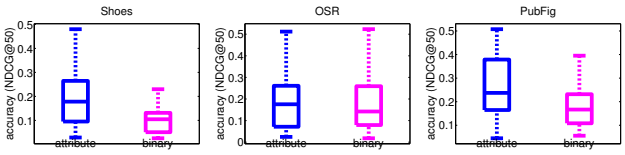


Figure 8. Ranking accuracy with human-generated feedback.

Next, we consider initialization with keyword search. The Shoes dataset provides a good testbed, since an online shopper is likely to kick off his search with descriptive keywords. Fig. 10(a) shows the ranking accuracy results for 16 queries when we restrict the reference images to those matching a keyword query composed of 3 attribute terms. Both methods get 4 feedback statements (we expect less total feedback to be sufficient for this setting, since the keywords already narrow the reference images to good exemplars). Our method maintains its clear advantage over the binary baseline. This result shows (1) there is indeed room for refinement even after keyword search, and (2) the precision of attribute statements is beneficial.

Fig. 7 shows a real example search using this form of our system. Note how the user's mental concept is quickly met by the returned images. Fig. 9 shows a real example using a hybrid of both binary and attribute feedback, as described in Sec. 3.3. This suggests how a user can specify a mix of both forms of input, which are often complementary.

4.3. Consistency of Relative Supervision Types

Finally, we examine the impact of how human judgments about relative attributes are collected.

Class-level vs. instance-level For all results above, we train the relative attribute rankers using image-level judgments. How well could we do if simply training with class-based supervision, i.e., "coasts are more open than forests"? To find out, we use the relative ordering of classes given

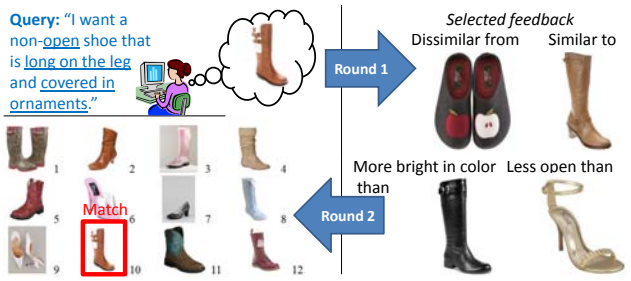
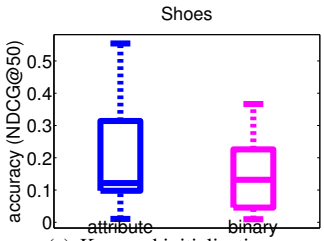


Figure 9. Example search result with hybrid feedback.



(a) Keyword initialization

	Class	Instance
Shoes	26.10%	**22.89%**
OSR	38.92%	**33.41%**
PubFig	**28.38%**	30.16%

(b) Training impact

Figure 10. (a) Accuracy using keyword search initialization. (b) Errors for class- vs. instance-level training.

in [20] for PubFig and OSR, and define them for Shoes (see project website). We train ranking functions for each attribute using both modes of supervision. Figure 10(b) shows the percentage of ∼200 test image pair orderings that are violated by either approach. Intuitively, instance-level supervision outperforms class-level supervision for Shoes and OSR, where categories are more fluid. Further, the 20 MTurkers' inter-subject disagreement on instance-level responses was only 6%, versus 13% on category-level responses. Both results support the proposed design for relative attribute training.

Absolute vs. relative Finally, we analyze the consistency in human responses when asked to make *absolute* judgments about the strength of an attribute in a single image (on a scale of 1 to 3) as opposed to *relative* judgments for pairs of images (more than, less than, or equal). In a similar study as above, for absolute supervision, the majority vote over half the subjects disagreed with the majority vote over the other half 22% of the time. For relative responses, this disagreement was somewhat lower, at 17%.

Conclusion We proposed an effective new form of feedback for image search using relative attributes. In contrast to traditional binary feedback, our approach allows the user to precisely indicate how the results compare with his mental model. In-depth experiments with three diverse datasets show relative attribute feedback's clear promise, and suggest interesting new directions for integrating multiple forms of feedback for image search. Studying other ways to select reference images is interesting future work.

Acknowledgements This research was supported by an ONR YIP grant and ONR ATL N00014-11-1-0105.

References

[1] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.
[2] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.
[3] I. Cox, M. Miller, T. Minka, T. Papathomas, and P. Yianilos. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *Trans on Img Proc*, 9(1), Jan 2000.
[4] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *ICCV*, 2011.
[5] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and Fisher vectors for efficient image retrieval. In *CVPR*, 2011.
[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.
[7] M. Ferecatu and D. Geman. Interactive search for image categories by mental matching. In *ICCV*, 2007.
[8] M. Flickner, H. Sawhney, W. Nilback, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, September 1995.
[9] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.
[10] T. Joachims. Training linear SVMs in linear time. In *KDD*, 2006.
[11] J. Kekalainen and K. Jarvelin. Cumulated gain-based evaluation of IR techniques. *ACM Trans Info Sys*, 20(4):422–446, 2002.
[12] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2008.
[13] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.
[14] T. Kurita and T. Kato. Learning of personal visual impression for image database systems. In *ICDAR*, 1993.
[15] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.
[16] W. Ma and B. Manjunath. NeTra: a toolbox for navigating large image databases. In *ICIP*, 1997.
[17] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3), 2006.
[18] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001.
[19] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011.
[20] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.
[21] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *Trans Multimedia*, 9(5), Aug 2007.
[22] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans on Circuits and Video Technology*, 1998.
[23] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.
[24] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003.
[25] K. Tieu and P. Viola. Boosting image retrieval. In *CVPR*, 2000.
[26] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *CVPR*, 2010.
[27] X. Wang, K. Liu, and X. Tang. Query-specific visual semantic spaces for web image re-ranking. In *CVPR*, 2011.
[28] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.
[29] E. Zavesky and S.-F. Chang. Cuzero: Embracing the frontier of interactive visual search for informed users. In *ACM MIR*, 2008.
[30] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8:536–544, 2003.

Please see `http://vision.cs.utexas.edu/whittlesearch` for the augmented Shoes dataset and additional results.