Discovering the Spatial Extent of Relative Attributes

Fanyi Xiao, Yong Jae Lee
Department of Computer Science, University of California Davis

1 Introduction

Visual attributes are human-nameable object properties that serve as an intermediate representation between low-level image features and high-level objects or scenes [3, 4, 5]. They can offer a great gateway for human-object interaction. For example, when we want to interact with an unfamiliar object, it is likely that we first infer its attributes from its appearance (e.g., is it furry or slippery?) and then decide how to interact with it. Thus, modelling visual attributes would be valuable for understanding human-object interactions. Researchers have developed systems that model binary attributes [3, 4, 5]—a property’s presence/absence (e.g., “is furry/not furry”)—and relative attributes [6, 8]—a property’s relative strength (e.g., “furrier than”). In this work, we focus on relative attributes since they often describe object properties better than binary ones [6], especially if the property exhibits large appearance variations (see Fig. 1).

While most existing work use global image representations to model attributes (e.g., [5, 6]), recent work demonstrates the effectiveness of using localized part-based representations [1, 7, 9]. They show that attributes—be it global (“is male”) or local (“smiling”)—can be more accurately learned by first bringing the underlying object-parts into correspondence, and then modeling the attributes conditioned on those object-parts. To compute such correspondences, pre-trained part detectors are used (e.g., faces [7] and people [1, 9]). However, because the part detectors are trained independently of the attribute, the learned parts may not necessarily be useful for modeling the desired attribute. Furthermore, some objects do not naturally have well-defined parts, which means modeling the part-based detector itself becomes a challenge. The approach of [2] addresses these issues by discovering useful and localized attributes. However, it requires a human-in-the-loop, which limits its scalability.

So, how can we develop robust visual representations for relative attributes, without expensive and potentially uninformative pre-trained part detectors or humans-in-the-loop? To do so, we will need to automatically identify the visual patterns in each image whose appearance correlates with attribute strength. In this work, we propose a method that automatically discovers the spatial extent of relative attributes in images across varying attribute strengths. The main idea is to leverage the fact that the visual concept underlying the attribute undergoes a gradual change in appearance across the attribute spectrum. In this way, we propose to discover a set of local, transitive connections ("visual chains") that establish correspondences between the same object-part, even when its appearance changes drastically over long ranges. Given the candidate set of visual chains, we then automatically select those that together best model the changing appearance of the attribute across the attribute spectrum. Importantly, by combining a subset of the most-informative discovered visual chains, our approach aims to discover the full spatial extent of the attribute, whether it be concentrated on a particular object-part or spread across a larger spatial area.

2 Approach

Given an image collection \( S = \{I_1, \ldots, I_N\} \) with pairwise ordered and unordered image-level relative comparisons of an attribute (i.e., in the form of \( \Omega(I_i) > \Omega(I_j) \) and \( \Omega(I_i) = \Omega(I_j) \)), where \( i, j \in \{1, \ldots, N\} \) and \( \Omega(I_i) \) is \( I_i \)'s attribute strength, our goal is to discover the spatial extent of the attribute in each image and learn a ranking function that predicts the attribute strength for any new image.

There are three main steps to our approach: (1) initializing a candidate set of visual chains; (2) iteratively growing each visual chain along the attribute spectrum; and (3) ranking the chains according to their relevance to the target attribute to create an ensemble image representation.

**Initializing candidate visual chains:** A visual attribute can potentially exhibit large appearance variations across the attribute spectrum. Take the "high-at-the-heel" attribute as an example: high-heeled shoes have strong vertical gradients while flat-heeled shoes have strong horizontal gradients. However, the attribute’s appearance will be quite similar in any local region of the attribute spectrum. Therefore, we start with multiple short but visually homogeneous chains of image regions in a local region of the attribute spectrum, and smoothly grow them out to cover the entire spectrum.

We start by first sorting the images in \( S \) in descending order of predicted attribute strength—with \( I_1 \) as the strongest image and \( I_N \) as the weakest—using a linear SVM-ranker trained with global image features. To initialize a single chain, we take the top \( N_{init} \) images and select a set of patches (one from each image) whose appearance varies smoothly with its neighbors in the chain, by minimizing the following objective function:

\[
\min_{P} C(P) = \sum_{i=1}^{N_{init}} ||\phi(P_i) - \phi(P_{i-1})||_2.
\]

where \( \phi(P) \) is the appearance feature of patch \( P \) in \( I \), and \( P = \{P_1, \ldots, P_{N_{init}}\} \) is the set of patches in a chain. Candidate patches for each image are densely sampled at multiple scales. This objective enforces local smoothness: the appearances of the patches in the images with neighboring indices should vary smoothly within a chain. Given the objective’s chain structure, we can efficiently find its global optimum using Dynamic Programming (DP).

In the backtracking stage of DP, we obtain a large number of \( K \)-best solutions. We then perform a chain-level non-maximum-suppression (NMS) to remove redundant chains to retain a set of \( K_{init} \) diverse candidate chains.

**Iteratively growing each visual chain:** The initial set of \( K_{init} \) chains are visually homogeneous but cover only a tiny fraction of the attribute spectrum. We next iteratively grow each chain to cover the entire attribute spectrum by training a model that adapts to the attribute’s smoothly changing appearance. Specifically, for each chain, we iteratively train a detector and in each iteration and use it to grow the chain while simultaneously refining it. To grow the chain, we again minimize Eqn. 1 but now with an additional term:

\[
\min_{P} C(P) = \sum_{i=2}^{N_{iter}} ||\phi(P_i) - \phi(P_{i-1})||_2 - \lambda \sum_{i=1}^{N_{iter}} w_i^T \phi(P_i),
\]

where \( w_i \) is a linear SVM detector learned from the patches in the chain from the \((i-1)\)-th iteration, \( P = \{P_1, \ldots, P_{N_{iter}}\} \) is the set of patches in a chain, and \( N_{iter} \) is the number of images considered in each iteration. As before, the first term enforces local smoothness. The second term is the detection term: since the ordering of the images in the chain is only a rough estimate and thus possibly noisy, \( w_i \) prevents the inference from drifting in the cases where local smoothness does not strictly hold. \( \lambda \) is a constant that trades-off the two terms. We use the same DP inference procedure used to optimize Eqn. 1.

Once \( P \) is found, we train a new detector with all of its patches as positive instances. The negative instances consist of randomly sampled patches...
whose intersection-over-union scores are lower than 0.3 with any of the patches in \( P \). We use this new detector \( w_i \) in the next growing iteration. We repeat the above procedure \( T \) times to cover the entire attribute spectrum. By iteratively growing the chain, we are able to coherently connect the attribute despite large appearance variations across its spectrum.

**Ranking and creating a chain ensemble:** We now have a set of \( K_{init} \) chains, each pertaining to a unique visual concept and covering the entire range of the attribute spectrum. However, some image regions that capture the attribute could have still been missed because they are not easily detectable on their own (e.g., forehead region for “visible forehead”). Since the patches in a chain capture the same visual concept across the attribute spectrum, we can use them as anchors to generate new chains by perturbing the patches locally in each image with the same amount of “perturbation”. Note that we get the alignment for the patches in the newly generated chains for free, as they are anchored on an existing chain. We generate \( K_{pert} \) chains for each of the \( K_{init} \) chains, which results in \( K_{pert} \times K_{init} \) chains in total.

Not all of the visual chains are relevant to the attribute of interest and some are noisy. To select the relevant chains, we compute the validation ranking accuracy for every visual chain and select the top \( K_{ens} \) chains accordingly to form the ensemble describing the attribute.

## 3 Results

We analyze our method’s discovered spatial extent of relative attributes as well as demonstrating a novel application called Attribute Editor.

**Visualization of discovered spatial extent:** We show qualitative results of our approach’s discovered spatial extent for each attribute in two datasets, LFW-10 and UT-Zap50K. For each image, we use a heatmap to display the spatial region for each visual chain for the attribute, and replace the corresponding patch in the query image with a patch from a different image that has a stronger/weaker predicted attribute strength. For color compatibility, we retrieve only those image regions with varying attribute strengths. To do this, we take the highest-ranked visual chain for the attribute, and replace the corresponding patch in the query image with a patch from a different image that has a stronger/weaker predicted attribute strength. For color compatibility, we retrieve only those patches that have similar color along its boundary as that of the query patch. We then blend in the retrieved patch using poisson blending.

**Attribute Editor:** One application of our approach is the Attribute Editor, which could be used by designers. The idea is to synthesize a new image, say of a shoe, by editing an attribute to have stronger/weaker strength. This allows the user to visualize the same shoe but e.g., with a pointier toe or sportier look. Fig. 3 shows four examples in which a user has edited the query image (shown in the middle column) to synthesize new images that have varying attribute strengths. To do this, we take the highest-ranked visual chain for the attribute, and replace the corresponding patch in the query image with a patch from a different image that has a stronger/weaker predicted attribute strength. For color compatibility, we retrieve only those patches that have similar color along its boundary as that of the query patch. We then blend in the retrieved patch using poisson blending.

---

**References**


