Transient Attributes
for High-Level Understanding and Editing of Outdoor Scenes

Pierre-Yves Laffont Zhile Ren Xiaofeng Tao Chao Qian James Hays
Brown University

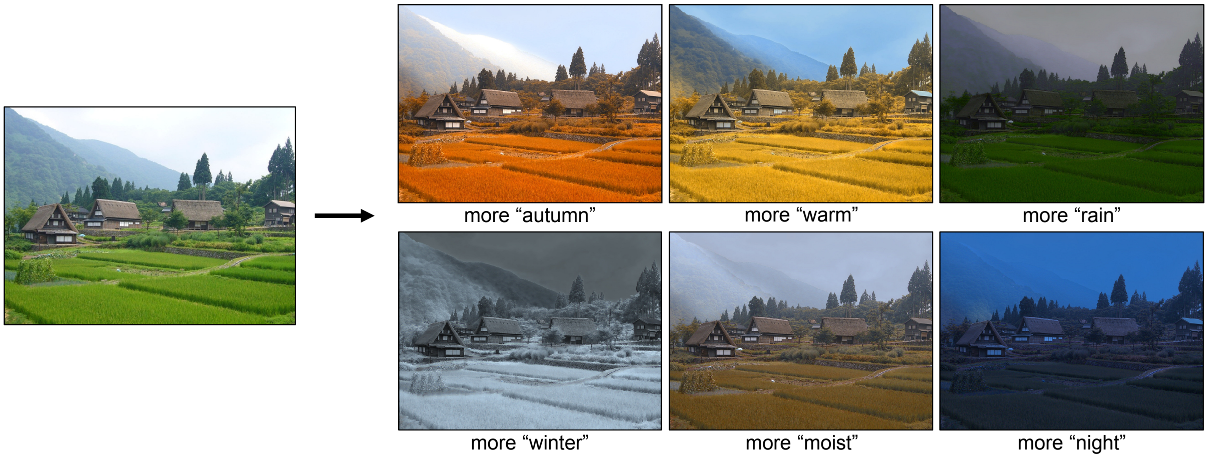


Figure 1: *Our method enables high-level editing of outdoor photographs. In this example, the user provides an input image (left) and six attribute queries corresponding to the desired changes, such as more "autumn". Our method hallucinates six plausible versions of the scene with the desired attributes (right), by learning local color transforms from a large dataset of annotated outdoor webcams.*

Abstract

We live in a dynamic visual world where the appearance of scenes changes dramatically from hour to hour or season to season. In this work we study "transient scene attributes" – high level properties which affect scene appearance, such as "snow", "autumn", "dusk", "fog". We define 40 transient attributes and use crowdsourcing to annotate thousands of images from 101 webcams. We use this "transient attribute database" to train regressors that can predict the presence of attributes in novel images. We demonstrate a photo organization method based on predicted attributes. Finally we propose a high-level image editing method which allows a user to adjust the attributes of a scene, e.g. change a scene to be "snowy" or "sunset". To support attribute manipulation we introduce a novel appearance transfer technique which is simple and fast yet competitive with the state-of-the-art. We show that we can convincingly modify many transient attributes in outdoor scenes.

CR Categories: I.4.8 [Image processing and computer vision]: Scene analysis—Time-varying imagery

Keywords: image database, attribute-based image editing

Links: DL PDF WEB

1 Introduction

The appearance of an outdoor scene changes dramatically with lighting, weather, and season. Fog might roll into a city as night falls, tree branches in a forest can be dusted with light snow, and an overcast day can clear into a gorgeous sunset. The visual world we live in is captivating because of these constant changes. A great deal of photography focuses on capturing images when these transient properties are interesting, unusual, or striking. This paper is the first large scale study of "transient scene attributes" which influence scene appearance. We address questions such as: How accurately can we recognize them? Can we modify scenes to change these properties?

To support these experiments we annotate thousands of images to create the Transient Attribute Database. A key property of the database is that rather than using unrelated images we sample photographs from 101 webcam sequences. This allows us to reason about *intra-scene* attribute changes. We can observe how a particular mountain scene changes with season or how a particular city scene changes with weather. These examples of intra-scene attribute variations will drive our attribute manipulation approach. We focus on outdoor scenes because they vary in more ways than indoor scenes (e.g. weather and seasons).

The Transient Attribute Database uses a taxonomy of 40 attributes related to weather (e.g. "cold" and "snow"), lighting (e.g. "sunny"), time of day (e.g. "dawn / dusk"), season (e.g. "autumn"), and more subjective impressions (e.g. "mysterious" and "soothing"). While some of these scene properties could be derived from ground truth meteorological data (e.g. when the temperature is above the threshold we consider the scene "hot") we instead annotate each attribute manually. Thus our attributes are *perceptual* in nature. We use a carefully cultivated pool of crowdsourced workers to annotate each image with independent attribute labels.

We use our database to train and evaluate regressors which predict the degree to which each attribute is present in a novel scene. We demonstrate that these predicted attributes can be used for photo organization and browsing.

The most ambitious goal of our study is to edit images with a simple user interface based on transient attributes ("make this image look *more sunny and snowy*"). Given a particular image to edit and a particular set of desired attributes, we learn how to change scene appearance from the transformations observed in our database.

We make the following contributions:

- We construct a dataset of 101 outdoor webcams containing images captured over long time spans which exhibit drastic changes in appearance. We align all images and use crowdsourcing to label them with attribute values (Section 3).
- We train regressors to recognize transient attributes in new images of outdoor scenes (Section 4).
- We enable high-level image editing through a simple interface in which users specify which attributes they want to change in a photograph (Section 5).
- We develop an example-based method to transfer drastic appearance changes observed in our database to a new input image (Section 6).

We make our *Transient Attribute Database* and our attribute predictors publicly available on the project website.

2 Related work

Attributes-based representations. Attributes are human-nameable concepts for high-level descriptions of visual phenomena. They stand in contrast to the more common categorical representations used to describe objects, scenes, activities, textures, etc. Attributes have recently emerged as a popular representation in the computer vision community, particularly for objects [Ferrari and Zisserman 2007; Farhadi et al. 2009] but more recently for scenes [Parikh and Grauman 2011], as well. Most closely related to our work is the SUN attribute database in which Patterson and Hays [2012] introduce 102 discriminative scene attributes. However, those attributes are designed to distinguish between scene categories (e.g. the functional attribute "eating" distinguishes a conference room from a cafeteria) and thus describe *inter-scene* variations. Instead, we focus on attributes for *intra-scene* variations – the appearance changes within one scene under varied conditions. Our ultimate goal is to manipulate transient attributes, and the attribute taxonomy in the SUN attribute database is inappropriate for this purpose because few of the attributes are dynamic.

We demonstrate the use of transient scene attributes for organizing photo collections which is similar to other uses of attributes for searching faces [Kumar et al. 2011], sky images [Tao et al. 2009], and objects [Kovashka et al. 2012]. Kovashka et al. [2012] also search scenes but only use a handful of static attributes (e.g. "man-made"). High level attributes have even been defined for BRDF appearance models to support navigation and manipulation within the space of valid BRDFs [Matusik et al. 2003].

Some of our attributes are subjective in nature (e.g. "soothing"). Along these lines, there has been prior work on recognizing subjective attributes such as photo quality or interestingness [Dhar et al. 2011]. There has also been work to enhance these subjective qualities in photographs [Kang et al. 2010; Caicedo et al. 2011; Bychkovsky et al. 2011].

Scene appearance variations. Garg et al. [2009] analyze the dimensionality of the space of scene appearance variations. Multiple photographs of the *same scene* captured from different viewpoints or at different times of day enable 3D navigation in the virtual scene [Snavely et al. 2006], and relighting [Yu and Malik 1998; Laffont et al. 2012]. Sunkavalli et al. [2007] model illumination changes from a single viewpoint. However, none of these methods relate appearance changes to high-level, nameable attributes.

Example-based appearance transfer A common method to guide changes to scene appearance is by providing a second example image with the desired properties. Global color transfer methods apply a global transformation on an input image to match color statistics of the example [Reinhard et al. 2001; Pitié et al. 2005]. They work well when the input and example images depict similar scenes or for hallucinating night images, but do not account for spatial layout of the scene. Other methods incorporate user input [An and Pellacini 2010; Pouli and Reinhard 2011] or a large database [Dale et al. 2009] to guide the color transfer. More recently, Cusano et al. [2012] transfer local color between regions annotated with the same class, and Wu et al. [2013] separately transfer the color style of each semantic region. Bonneel et al. [2013] use color transforms to transfer visual styles across videos.

Instead of using a single example image, image transformations can be learned by analogy from a pair of example images [Hertzmann et al. 2001]. Like our work, the closely related method by Shih et al. [2013] learns to manipulate scenes from a database of webcam sequences in an analogy-based framework. However, it focuses on a particular scene attribute – time of day. We propose a new appearance transfer technique and compare with Shih et al. in Section 6.

High-level image editing. Our work belongs to a growing body of research which focuses on allowing users to direct image manipulation with high-level, semantic guidance rather than low-level control. This is appealing because it allows *non-expert* users to perform image synthesis or manipulation operations that are historically only available to experts and artists. For example, Photo clip Art [Lalonde et al. 2007] allow users to insert common object types (e.g. 'car') into existing scenes by finding objects which are compatible with the scene from a large database.

Berthouzoz et al. [2011] enable high-level image editing by adapting existing macros (e.g. sequences of recorded Photoshop operations) to novel scenes. Many of their manipulations are object- or human-centric but they also demonstrate scene manipulations similar to ours such as adding snow. However, their method adds snow by intelligently replaying a sequence of artistic operations whereas we are learning what it means to become snowy from annotated example scenes which undergo a snowy transformation.

CG2real [Johnson et al. 2011] adds realism to computer generated images by transferring color and texture from real photographs. This could be thought of as scene attribute manipulation in the case of one attribute – realism. Likewise there are other methods built to change one particular high level attribute of a scene, e.g. the amount of haze in the atmosphere [Fattal 2008]. The key distinguishing factor of our work is that we aim to have a common framework to manipulate many dynamic properties of scenes based on what we learn from our image database.

Like our work, Cheng et al. [2014] recognize and manipulate attributes in images. However, they focus on object-centric attributes in indoor scenes (e.g. make the brown couch red) and do not emphasize learning the attribute manipulations from data.

Image datasets and webcams. A significant contribution of this paper is the crowdsourced annotation of a new database of dynamic scenes. In the past decade graphics research has come to rely more on such image databases. For example, crowdsourcing has enabled object insertion in Photo Clip Art [Lalonde et al. 2007], analysis and recognition of human object sketches [Eitz et al. 2012], and extensive surface appearance reference data in [Bell et al. 2013].

Our database uses sequences from the Webcam clipart [Lalonde et al. 2009] and AMOS [Jacobs et al. 2007] databases, which we augment with crowdsourced annotations of many scene attributes. Shih et al. [2013] construct a database of 450 high quality time-lapses which cover short time spans, exhibiting mostly lighting changes due to time of day. Like in our work, Narasimhan et al. [2002] link scene appearance changes over several months to weather and lighting properties, but their database includes only a single urban scene and the annotations are generated automatically from meteorological data. We describe how we gather and annotate our *Transient Attribute Database* in the next section.

3 The *Transient Attribute Database*

The first difficulty in studying outdoor appearance changes is the lack of high-quality images from static viewpoints captured over long periods of time and indexed by the transient properties they exhibit. In order to construct such a dataset, we select a subset of outdoor webcams from two existing sources: the Archive of Many Outdoor Scenes [Jacobs et al. 2007] and the Webcam Clip Art Dataset [Lalonde et al. 2009]. Images in our dataset exhibit the following properties:

Large variations within each scene: for each webcam, we select 60-120 high quality frames that are representative of the appearance variations of the scene; all images are manually reviewed to ensure they do not exhibit serious artifacts such as excessive noise, quantization artifacts, or a dirty lens

Diversity across scenes: our webcams cover different types of outdoor scenes, ranging from mountainous landscapes to urban scenes at different scales

High resolution: image size varies between 640×360 and 4000×3000, with an average resolution of 1.8 megapixels

Accurate alignment: we align all images in each webcam to a reference frame by manually specifying correspondences and applying a homography warp.

In total, our dataset contains 8571 images from 101 webcams. In the rest of this section, we determine which attributes are of interest and describe how we annotate each image of our dataset.

3.1 Discovering transient attributes

There are a litany of high level properties that affect scene appearance, but not all of them are common or easily perceptible (e.g. “eclipse” would be extremely rare and “Tuesday” is probably hard to recognize). We manually define an initial list of 92 scene attributes by collecting a list of adjectives and nouns frequently recurring in written descriptions of outdoor scenes. While some appeared in prior work [Patterson and Hays 2012], such as *spatial envelope* properties (“natural”, “enclosed area”), we add attributes related to lighting (“daylight”, “sunrise”), weather (“fog”, “rain”) and season. We then reduce this list to 40 transient attributes by conducting a crowdsourced experiment on Amazon Mechanical Turk.

Crowdsourced task. In each crowdsourcing task, we show workers all images from a single webcam. For five attributes in our initial list, along with their definition, we ask which ones appear in “all / some / none” of the images. Each task is repeated multiple times to establish consensus.

Results suggest that most of the scene attributes described in prior work do not vary much across images of one scene. In particular, spatial envelope attributes such as “open area” are constant within each scene. However, properties related to weather, lighting, or emotions when viewing the image, can vary drastically across images of one scene.

Transient attributes. After discarding attributes that are rarely present or do not vary within each scene, and grouping pairs of attributes that are correlated or difficult to distinguish (e.g., sunrise/sunset), we obtain **40 transient attributes** in five categories:

- **lighting**: sunrise/sunset, bright, daylight, etc.
- **weather**: sunny, warm, moist, foggy, cloudy, etc.
- **seasons**: spring, summer, autumn, winter
- **subjective impressions**: gloomy, soothing, beautiful, etc.
- **additional attributes**: active/busy, cluttered, dirty/polluted, lush vegetation, etc.

The complete list of our 40 transient attributes, along with their definitions and examples of positive/negative images, is shown in the supplementary material.

3.2 Labeling images with transient attributes

Now that we have established a taxonomy of transient scene attributes we are ready to annotate each database image. Many of our attributes are continuous in nature, but it is difficult to ask annotators to directly assign a real-valued score – what does 0.3 “fog” look like? Another option would be to infer continuous labels from many pairwise rankings, but our attributes often cluster around extremal values which would lead to many uninformative ties (e.g. 78% of images do not have the “night” attribute). Therefore we ask each crowdsourced worker to drag each photograph into one of four bins, according to “how much each image exhibits this attribute”. Possible answers are “totally / a little / not at all”, corresponding respectively to discrete label values 1 / 0.5 / 0. Workers can mark uncertain images by dragging them to the “unsure” bin. Each crowdsourced annotation task corresponds to one webcam and one attribute.

Combining annotations. Although each worker provides a discrete label for each image, we repeat the experiment multiple times with different workers and combine their annotations in order to obtain continuous attribute values between 0 and 1. Because workers have varying reliability, we use control items (i.e., image-attribute pairs with known answer) to evaluate each worker’s performance then aggregate all workers’s annotations by weighting them based on reliability.

We ask an expert to create control items by annotating 15 objective attributes related to time-of-day and weather in 34 webcams. This corresponds to 510 annotations tasks (12.6% of the total number of annotation tasks). In each task, the expert labeled only images for which he was extremely confident (i.e., the attribute is definitely present or absent), leading to 4.5 positive and 7.7 negative annotations on average. In total, the expert marked 6041 control image-attribute pairs (1.8% of the total number of image-attribute pairs).

We model annotation noise and workers’s unreliability with the *bias-variance* Gaussian model described by Liu et al. [2013]. The discrete label x_{iaw} produced for image i and attribute a by worker w can be written as:

$$x_{iaw} = \mu^*_{ia} + b^*_w + \xi_{iaw}, \quad \xi_{iaw} \sim \mathcal{N}(0, \sigma^{*2}_w) \qquad (1)$$

where μ^*_{ia} is the true label of attribute a on image i, b^*_w the bias of worker w, σ^{*2}_w the worker’s variance, and ξ_{iaw} represents annotation noise.

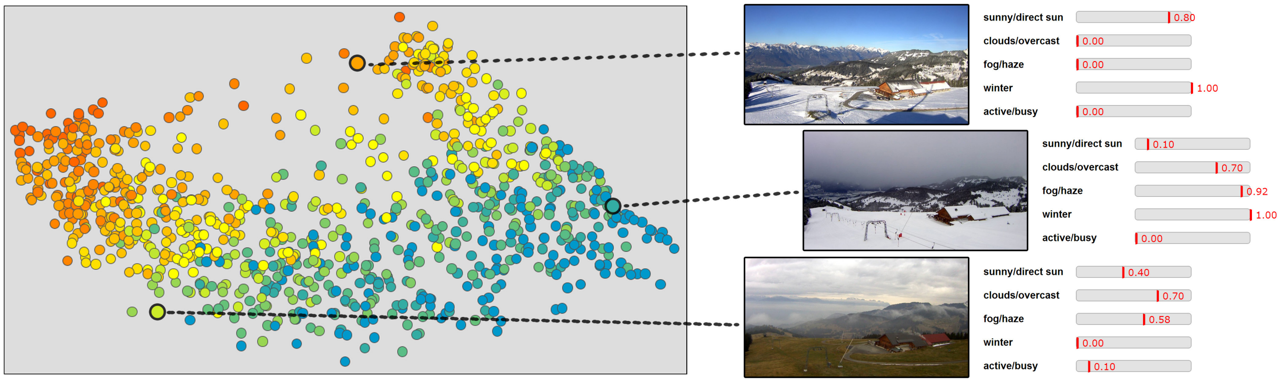


Figure 2: *Transient scene attributes allow us to characterize each image of a scene with a small number of intuitive dimensions. Each image in a photo collection is represented as a point in* ***attribute space****, where each dimension corresponds to a scene property which can vary with time, weather, or lighting conditions. Left: projection of 200 images of our dataset on the dominant plane of attribute space (obtained by PCA); each image is represented as a dot, color-coded according to its value of the "sunny" attribute. Right: values of a few* ***transient attributes*** *for three photographs. The scene appearance and its attributes vary widely between the three images, despite the fixed viewpoint.*

We use the two-stage estimator described by Liu et al. [2013] to estimate workers' parameters and attribute labels. In the *scoring step*, each worker's bias and variance are estimated using the known labels μ^*_{ia} of the control items she answered. In the *prediction step*, the aggregated value of each image-attribute pair is estimated by combining the annotations of all workers, which are bias-corrected and weighted according to their variance.

At the end of this step, we have obtained aggregated attribute labels μ_{ia} which range from 0 (attribute is not present) to 1 (attribute is present), and the estimated bias and variance for each worker.

Identifying reliable workers. The estimated bias and variance parameters also allow us to filter out unreliable workers. We filter out workers whose bias or variance is greater than one standard deviation above the average of all workers ($|b^*_w| > 0.13$ or $\sigma^*_w > 0.52$) or who answered less than 50 control items. We grant custom qualifications to reliable workers, giving them access to more annotation tasks with a better pay (10.5 cents per task on average).

We repeat each task until at least 5 reliable workers submit their annotations. In total, 288 workers completed 37924 annotations tasks. All workers were paid for their work, but we only use annotations from the 37 most reliable workers, who completed 24333 tasks.

3.3 Discussion

Attribute space visualization Transient attributes provide a high-level description of scenes and can be used to organize large collections of photographs such as our dataset. Figure 2 shows 200 images from one scene in our dataset, represented as colored dots and laid out according to their attribute labels. This visualization gives an overview of the attribute distribution in the collection.

Additional analysis We report the correlation between different attributes and we compare our aggregated attribute values for weather-related attributes to actual meteorological measurements in the supplementary document.

Limitations While our dataset is fairly large in terms of number of images it has only 101 unique scenes. By focusing on particular scenes we observe *intra-scene* variations and this enables our attribute manipulation approach (Section 6); however, it also makes our recognition task harder because we have fewer independent examples of attribute-scene combinations.

It is difficult to compare the strength of attributes between different types of scenes. E.g. does 0.6 "hot" mean the same thing in a mountain scene and a beach scene? Our annotation tasks are *intra-scene* so we do not attempt to calibrate for these differences. However, our manipulation method is scene dependent (mountain scenes will be compared with mountain scenes) and does not rely on universally calibrated attributes.

4 Recognizing attributes in new images

The 40 attributes we define and annotate in the previous section are more useful if we can automatically recognize them in new images. In this section, we use the *Transient Attribute Database* to train and evaluate regression and classification methods. We demonstrate how attribute prediction enables the exploration of photo collections in Section 4.3.

4.1 Learning to recognize attributes

For each attribute a, our goal is to predict the attribute label $\tilde{\mu}_{i'a}$ in new images i', given the aggregated labels μ_{ia} in all images of our dataset. Because our aggregated attribute labels are continuous we use regression to predict continuous attribute labels on new scenes. We train different non-linear predictors using the annotated images:

Support Vector Machines (SVM). As a baseline, we train an SVM classifier for each attribute [Patterson and Hays 2012]. To produce a continuous output label $\tilde{\mu}_{i'a}$ instead of a binary label, we linearly scale the SVM confidence value into the range of 0 to 1. To improve the performance of this baseline we use only *strong positive* or *strong negative examples* for training ($\mu_{ia} \geq 0.8$ or $\mu_{ia} \leq 0.2$).

Logistic regression (log reg). We use kernel logistic regression [Murphy 2012] to directly predict $\tilde{\mu}_{i'a}$.

Support Vector Regression (SVR). In Support Vector Regression, the loss function is based on deviation from a real valued training label rather than a binary class label as in standard SVMs. We train a ν-SVR model [Scholkopf et al. 2000] for each attribute to predict $\tilde{\mu}_{i'a}$.

All three methods are trained with the same global image features and individual kernels, which have been shown to work well in

	Random split		Holdout split	
	MSE	AP	MSE	AP
SVM	0.045	0.95	0.070	0.77
log reg	0.063	0.93	0.093	0.75
SVR	**0.018**	**0.97**	**0.043**	**0.80**

Table 1: *Comparison of Support Vector Machines (SVM), logistic regression (log reg), and Support Vector Regression (SVR) for recognizing attributes on two test splits. We use mean squared error (MSE) and average precision (AP) to evaluate their performance.*

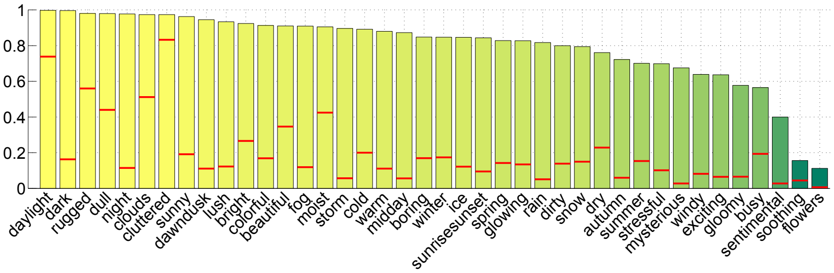


Figure 3: *Average precision for each attribute using SVR on the holdout test split. Our regressors clearly score much higher than chance, indicated by the red lines.*

scene classification [Xiao et al. 2010]. We use histograms of oriented gradients (HOG), self-similarity features (SSIM), GIST, and geometric context color histograms. We normalize individual kernels and average them together in the same way as Patterson and Hays [2012], who used this combination of kernels for scene attribute recognition. We use Fisher Vector encoding [Perronnin et al. 2010] for HOG and SSIM as it performs better than Bag of Visual Words; we compare these encoding methods in the supplementary document.

4.2 Evaluation

We evaluate the performance of these methods on two different training-test splits:

Random split where the test set contains random images selected from all webcams (roughly 20% of the entire dataset),

Holdout split where the training and test sets contain 81 and 20 separate webcams, respectively.

Both splits contain a similar number of test images. However, the holdout split is more difficult because the training set does not contain images from the 20 scenes in the test set – the trained regression methods are being evaluated on completely new scenes. We evaluate the prediction performance with two metrics:

Mean squared error (MSE) is the squared error between $\tilde{\mu}_{ia}$ and μ_{ia} averaged on all test set images for attribute a; it measures how far a method's prediction deviates from the ground truth on average;

Average precision (AP) is the area under the precision-recall curve or equivalently the average of precision over all recall values; because AP is a measure of classification accuracy we only test on the strong positives and negatives in the test sets. We report AP because it is easier to interpret than MSE, but we are most interested in minimizing MSE.

Table 1 reports the mean performance for each method, averaged over all attributes. Unsurprisingly, it is much easier to predict attributes when training data from the same scenes is available in the "Random" test set. When test images are from completely held out scenes in the "Holdout" test set the average precision for all three methods is between 0.75 and 0.80. However, SVR has much lower MSE than SVM because it uses the entire range of attribute labels for training, whereas SVM uses only strong positive and strong negative examples. This suggests that both methods are equally useful for predicting binary attributes (e.g., "ice, daylight"), while SVR is the preferred method for more continuous attributes (e.g., "lush, warm").

Figure 3 shows the per-attribute AP of SVR on the holdout split. It is compared to chance performance (indicated as red bars), which equals the proportion of strong positives in the test set for each attribute. The AP of our predictors is many times higher than chance on all attributes which indicates that we can achieve good performance on entirely new scenes. The average precision is higher for more objective attributes such as daylight or clouds and lower for more subjective attributes such as mysterious and soothing.

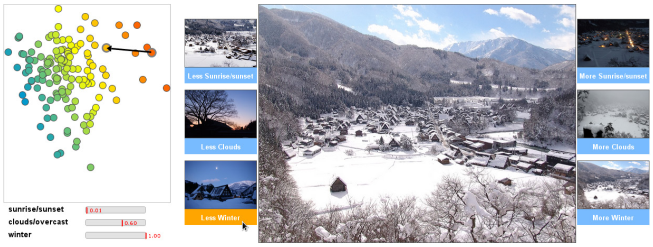


Figure 4: *Screen capture of our browsing interface, also shown in the **supplementary video**. The user can interactively explore the photo collection by retrieving an image (right) with "more" or "less" of a specific attribute, such as "clouds". Each new query corresponds to a walk in a specific direction of the attribute space, represented as an arrow in the color-coded visualization (left).*

The number of strong positive examples is critical for recognition. For example, the bottom three attributes in AP ("sentimental, soothing, and flowers") had among the lowest numbers of strong positive examples. We achieve an overall AP of 0.86 when considering only the 33 attributes with more than 200 strong positive examples.

4.3 Application: browsing photo collections

We apply our attribute recognition approach to browsing and searching online photo collections. We extract the features for each image of a photo collection and estimate its attribute labels using the SVR regressors trained in Section 4.1.

In addition to the visualization shown in Figure 2, we propose new user interfaces that leverage the estimated attribute labels:

An image retrieval interface which returns a subset of images corresponding to the desired attribute query. The query consists of an acceptable range of labels for the desired set of attributes, and can be inferred from a query string ("some snow, no clouds"). The results may be sorted by any attribute.

A browsing interface which enables the user to explore the collection and progressively get closer to the desired image by providing feedback, e.g. "more clouds", then "less winter". Figure 4 shows a screen capture of this interface.

The supplemental video shows how these interfaces are used to browse unlabeled photo collections.

5 Attribute-guided image editing

Our most ambitious goal is to allow casual users to edit images with a simple user interface based on transient attributes. Given a set of attributes that the user wants to modify in the scene ("make it more snowy and cloudy"), we leverage the dataset constructed in Section 3 to find example images that correspond to a similar scene and exhibit the desired attribute change. We then edit the input image using an analogy-based appearance transfer approach.

Finding similar scenes. Appearance transfer methods work better when the input and exemplar images correspond to similar scenes; e.g. transferring the appearance of a lake onto a corn field is undefined. Our first step is to identify images in our database that depict scenes similar to the input using a standard scene matching approach. We compute the distance between the input and each image in our dataset using some of the features described by Xiao et al. [2010]: HOG, GIST, color histograms and tiny images. We keep the 6 images with the lowest distance, with at most one image per webcam. We call these *match images*. We found that this combination of features works well with our database, yielding match images that correspond to similar scenes with similar color distributions and attributes. We experimented with explicitly matching based on the predicted attributes of the input and the ground truth attributes of the database images but this did not improve match quality over the non-parametric scene matching approach.

Selecting target image(s). The next step is to choose *target images* that contain the desired attributes in the relevant webcams. One challenge is that an attribute change such as "more winter" can manifest as many distinct appearance variations: the ground may become covered in snow or left bare, trees may lose their leaves if they are deciduous, a body of water may or may not freeze. Each "winter" image in our database features a different subset of these transformations. We automatically identify a number of *candidate target images* which contain the desired attribute, and let the user decide which target image(s) correspond(s) the most to the desired manipulation.

We now describe how we select the candidate target images shown to the user. We represent the desired attribute changes as a per-attribute offset Δ_a, where Δ_a is a scalar between -1 and 1 for the attribute(s) a to change, and zero for all the other attributes. For each match image M, we aim to find images from the same webcam that are close to the attribute labels μ_{Ma} shifted by Δ_a. We compute a score for each image i based on a weighted L_2-distance:

$$D(i) = \sum_a \omega_a (\mu_{Ma} + \Delta_a - \mu_{ia})^2 \tag{2}$$

where $\omega_a = 1$ for the attributes to modify and 0.01 for other attributes, in order to enforce the desired changes and encourage preserving unconstrained attributes.

We keep the top 3 images for each webcam as *candidate target images*. We show these images to the user on a web interface, and she then selects the actual target image to be used for appearance transfer. Alternatively, the fast method we propose in Section 6 can generate multiple plausible results by using every candidate target image; the user can then pick her favorite result.

Appearance transfer. At this stage, we have: (1) input image I; (2) match image M, which depicts a similar scene in our database; (3) target image T, which exhibits the desired attribute shift and which comes from the same webcam as M. We now aim to modify I in order to make it closer to T. We can use existing color transfer methods [Reinhard et al. 2001; Pitié et al. 2005] to transfer the color statistics of T to I. Alternatively, an analogy-based approach [Shih et al. 2013] can leverage the variations between M and T and apply them to I.

Regardless of the particular appearance transfer method utilized, a key contribution of this work is that the Transient Attribute Database coupled with the scene matching method and target selection criteria in this section allows a non-expert user to modify an image by specifying desired high-level properties – the transient attributes. We spare the user from the task of finding suitable example images manually which would be tedious and difficult with thousands of images. Because we want our system to be interactive speed so that users can explore attribute variations induced by different target images, we propose a novel appearance transfer method in the following section. We show examples of attribute-guided manipulations in Figures 6 and 7.

6 Learning local transforms for appearance transfer

In this section, we propose a fast method to dramatically modify the appearance of input image I by transferring the changes observed between match image M and target image T. This is possible only because images M and T come from the same webcam and have been carefully aligned in our database.

Our approach is based on the observation that groups of similar pixels in M tend to form groups of similar pixels in T. Similarly, local transformations which turn patches in M into patches in T tend to form clusters in the space of transform parameters (Fig. 5a). We precompute transformations for pairs $M{:}T$ in our database, and store them in a *transform library*.

The challenge is to find which transform should be applied at each pixel p_{query} of the input image I. We find pixels similar to p_{query} in M based on local image features, and retrieve the corresponding local transforms observed in the pair $M{:}T$ (Fig. 5b). We then apply these transforms to I, thus transferring the appearance changes learned from our database onto a new scene.

6.1 Precomputing the transform library

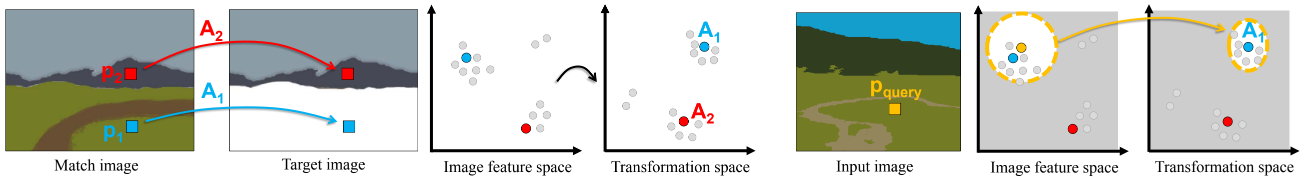
For each image pair $M{:}T$ in our database, our goal is to estimate local transformations which explain the color variations between the two images. We use a locally linear model [Shih et al. 2013] which relates the color of pixels in M to the color of pixels in T. We denote by $\mathbf{v}_k(M)$ the patch centered on pixel p_k in the match image and by $\mathbf{v}_k(T)$ the corresponding patch in the target image. Both are represented as $3 \times N$ matrices in RGB color space; we use patches of $N = 5 \times 5$ pixels. The local linear transform applied on patch k is represented by a 3×3 matrix $\mathbf{A}_k$, and can be estimated with a least-squares minimization:

$$\underset{\mathbf{A}_k}{\operatorname{argmin}} \left\| \mathbf{v}_k(T) - \mathbf{A}_k \mathbf{v}_k(M) \right\|_{\mathsf{F}}^2 + \gamma \left\| \mathbf{A}_k - \mathbf{G} \right\|_{\mathsf{F}}^2 \tag{3}$$

where $\|\cdot\|_{\mathsf{F}}$ denotes the Frobenius norm. The second term regularizes $\mathbf{A}_k$ with a global linear matrix $\mathbf{G}$ estimated on the entire image (we use a small weight $\gamma = 0.01$). We obtain the optimal transform A_k in closed form:

$$\mathbf{A}_k = \left(\mathbf{v}_k(T)\mathbf{v}_k(M)^{\intercal} + \gamma \mathbf{G}\right)\left(\mathbf{v}_k(M)\mathbf{v}_k(M)^{\intercal} + \gamma \mathbf{Id}_3\right)^{-1} \tag{4}$$

where $\mathbf{Id}_3$ depicts the 3×3 identity matrix. Using Equation 4, we precompute the transformation corresponding to each pixel p_k in image pairs $M{:}T$. Note that this differs from the approach by Shih et al. [2013] in which transformations are computed only after M and T have been warped into dense correspondence with the



(a) *Precomputed local transformations*

(b) *Transformations to apply on the input image*

Figure 5: *Schematic overview of our appearance transfer method. (a) We learn local transformations by observing a pair of match-target images. Transforms corresponding to similar regions in the match image tend to form clusters in transformation space. (b) Given a pixel p_{query} in the input image, we retrieve the transformations corresponding to similar pixels in the match image. Those transforms form a transformation pool, from which we draw the transforms to apply on the input image.*

input image. Instead, we have computed the transformations on the unmodified image pairs $M{:}T$ and will now address the correspondence between I and $M{:}T$.

6.2 Transforming the input image

Given all the transforms computed from the pair $M{:}T$, we need to find which transform to apply on each pixel of I. Instead of relying on dense correspondence between the input and match image as Shih et al. [2013], we propose a simple algorithm which is significantly faster yet yields convincing results and prevents color inconsistencies between similar regions in the input image.

Segment matching. We first segment the input image into superpixels and compute the likelihood that each segment is ‘ground’, ‘sky’, ‘vertical’, or ‘porous’ [Hoiem et al. 2007]. We also compute color and texton histograms for each segment. We calculate the distance from each input segment to precomputed segments in the match image using L^2 distance for the surface likelihoods and χ^2 distance for the histograms. Finally, we select the best matched segments with the lowest distance to be in correspondence.

Pool of transforms. For each input segment, we retrieve the precomputed transforms $\mathbf{A}_k$ corresponding to pixels p_k in the match segments. This yields a *pool of transforms*, which model local changes that have been observed in the transformation library on image regions similar to the input segment. These transformations are mostly consistent and tend to form clusters in transformation space (Figure 5b). We found that applying outlier rejection in the space of transformations can improve the manipulation results on a few images. We apply mean-shift clustering [Comaniciu and Meer 2002] and keep only transforms in the largest cluster. We use bandwidth $\sigma_{\text{outliers}} = 0.09$ for all results.

Applying the transforms. Applying a single transform per segment would yield artifacts at the segment boundaries. Instead, we assign a transformation to each pixel and use edge-aware filtering to smooth the transformations spatially. First, at each pixel we randomly select a transform from the corresponding segment’s transformation pool (alternatively, selecting the mean transform yields comparable results). Then, we apply a cross-bilateral filter [Chen et al. 2007] which smooths each component of the transformation matrices according to edges in the input image. At each pixel $\mathbf{p}$, the filtered version of the transform is:

$$\tilde{\mathbf{A}}_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in N(\mathbf{p})} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(|\overline{I_{\mathbf{p}}} - \overline{I_{\mathbf{q}}}|)\, \mathbf{A}_{\mathbf{p}} \qquad (5)$$

which is a simple weighted average over a neighborhood where the weight is the product of a Gaussian on the spatial distance in pixels (G_{σ_s}) and a Gaussian on the pixel value difference (G_{σ_r}) of the grayscale input image. We use $\sigma_s = 14, \sigma_r = 0.1$ for all results. We obtain the output image by applying the filtered transforms on each pixel of the input image. In order to generate a high-definition output image, we can similarly upsample the transforms [Kopf et al. 2007] before applying them to a full-size input image.

Global consistency. Pixels in consistent regions (e.g., grass, sky, or building facade), should be transformed similarly across the entire image. Global color transform methods handle this well, since they operate on the entire image. However, methods based on spatial correspondences are sensitive to this issue, which can yield low-frequency color inconsistencies (see the result from Shih et al. [2013] on Fig. 8d). The edge-aware filtering we used in previous paragraph reduces discrepancies between adjacent segments, but it cannot enforce that similar regions in different parts of the image undergo the same transform.

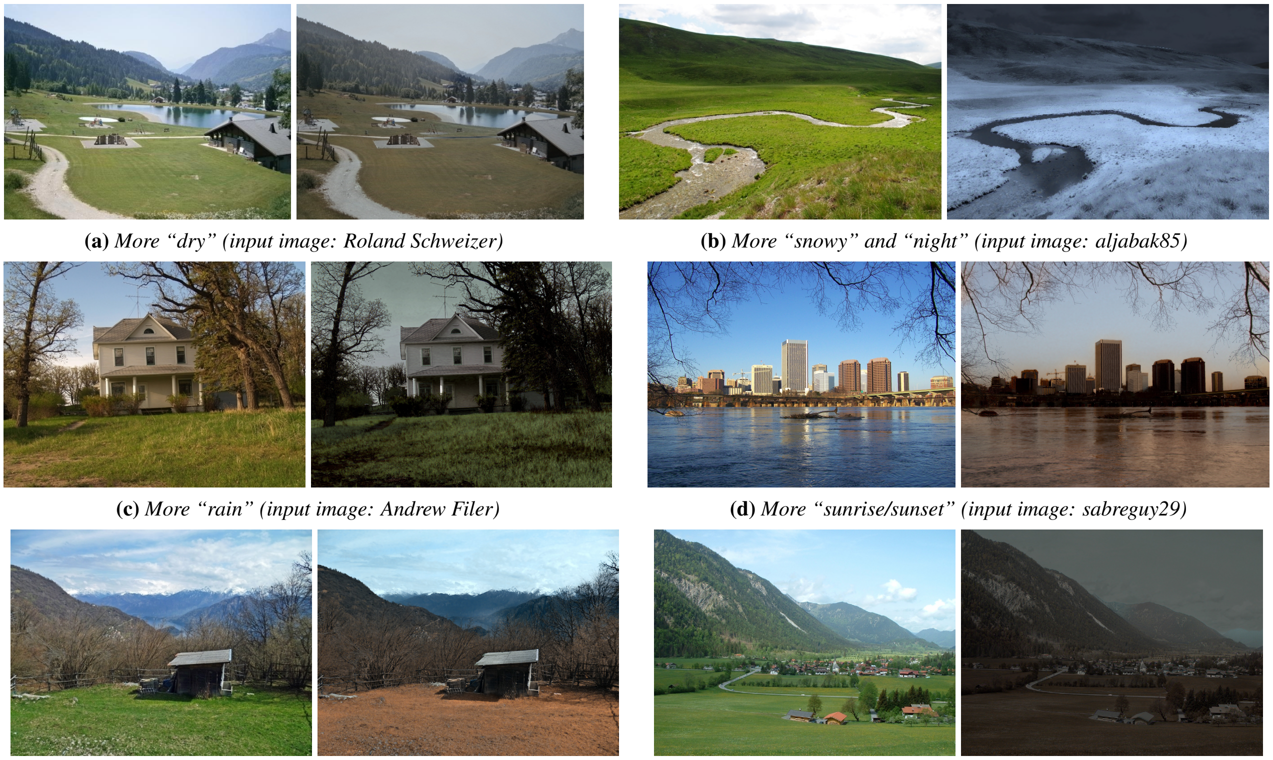
We propose an additional step in order to enforce *global consistency*. After segment matching, we use mean shift clustering to group similar segments according to their mean RGB color; we use bandwidth $\sigma_{\text{global}} = 0.13$ for all results. We then merge the transformation pools from all these segments. This ensures that the transformations applied on similar segments will be drawn from the same pool, which addresses the consistency problem Fig. 8h).

6.3 Results

Attribute-guided image editing. We present results for attribute-guided appearance transfer in Figure 6. For each input image, we choose one or two attributes that we would like to change, then select a target image using the interface described in Section 5. Our method changes the scene appearance dramatically yet produces plausible results. Notice how, in (a), the grass becomes drier but the cabin remains untouched. In the “snowy and night” example (b), our method covers some of the fine texture details of the grass with snow. More subjective attributes such as “gloomy” can also be manipulated (f).

By manipulating a single input image with several different attribute queries, we can synthesize plausible images of the scene at a different time of day, season, or weather condition. We show such *virtual timelapses* in Figure 7 and in the supplementary video.

Comparison of appearance transfer methods. We compare our appearance transfer method with three approaches in Figure 9, using the code released by their authors. Note that none of these appearance transfer methods natively support the high-level attribute manipulations being demonstrated; in each test case, we use the interface described in Section 5 to specify the attribute to modify and select one of the proposed target images. All methods perform



(a) *More "dry" (input image: Roland Schweizer)*

(b) *More "snowy" and "night" (input image: aljabak85)*

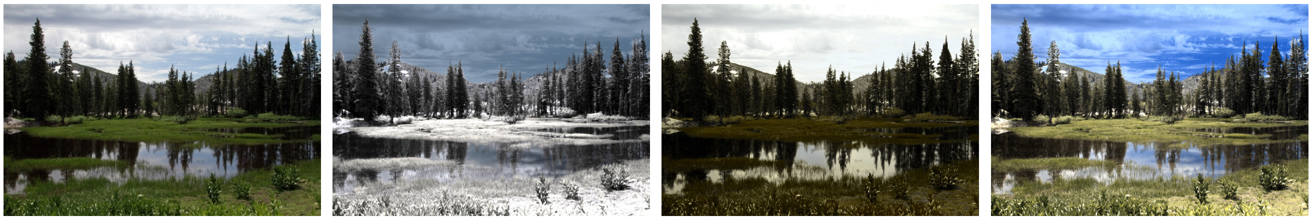
(c) *More "rain" (input image: Andrew Filer)*

(d) *More "sunrise/sunset" (input image: sabreguy29)*

(e) *More "autumn" and "bright" (input image: Charlie Dave)*

(f) *More "gloomy" (input image: Michael Freyermuth)*

Figure 6: *Results of our method for six attribute modifications. In each case, a single input photograph is used (left) and the user selects a target image proposed by our attribute-guided interface (Section 5). Our appearance transfer method (Section 6) then synthesizes an image with the desired attributes (right). Image regions are modified differently according to their semantic content, such as grass or mountains (e).*



(a) *Input image: Vlad Butsky*

(b) *More "storm"*

(c) *More "moist"*

(d) *More "glowing"*

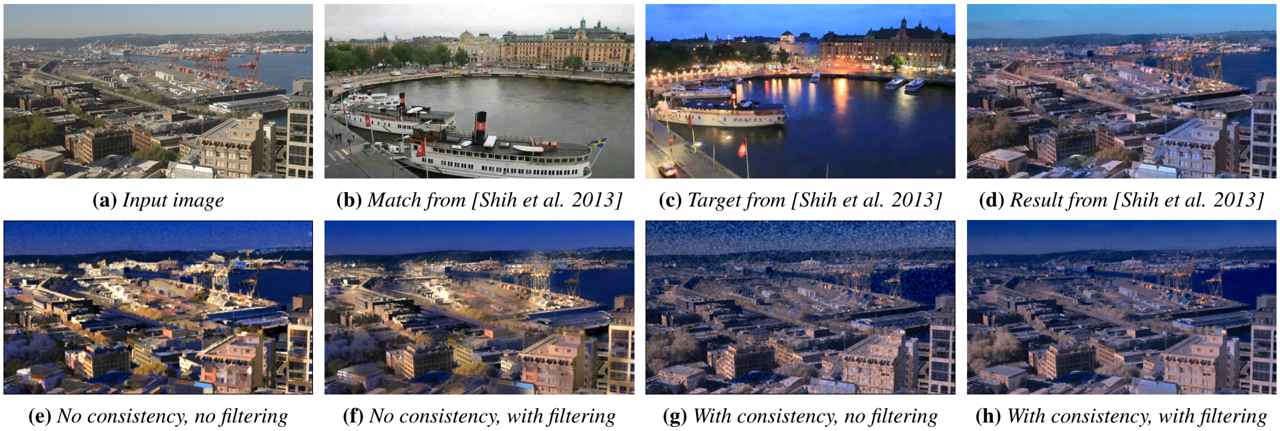
Figure 7: *Our appearance transfer method can generate multiple variations of a single input photograph by varying its attributes. The supplementary video shows* ***virtual timelapses*** *of scenes with different seasons, time of day, and weather conditions.*

relatively well on the "more cloudy" example, which can be well approximated with a global change in color (generally darker, with less contrast). However, global color transfer methods do not perform well on appearance changes that involve modifying the scene materials or texture ("more winter", middle row). In contrast, our method and Shih et al. [2013] plausibly cover the ground with snow because they leverage variations between match and target images. Only our method manages to hallucinate fog in the top example.

We conduct a user study to further compare different appearance transfer techniques. We compare our fast local method to global transfer [Reinhard et al. 2001] and local transforms [Shih et al. 2013] on 60 attribute manipulations, shown on the project website along with the corresponding match/target images. We used 15 participants in a lab environment; average study duration was 37 minutes. We showed one pair of result images at a time and asked the following two questions: "Which of these images looks more like a real photograph?" and "Which image more convincingly seems to have attribute x?" Participants gave 5400 total pairwise evaluations. According to this study:

- Global color transfer results look more like real photographs, compared to local transfer results (87% preference compared to Shih et al. [2013] and 81% compared to ours, respectively; $p < 0.005$ for both). While global changes are less likely to produce artifacts, they are also much less expressive. Our approach, which learns from a pair of match-target images rather than a single target, produces more convincing attribute changes 70% of the time ($p < 0.005$).

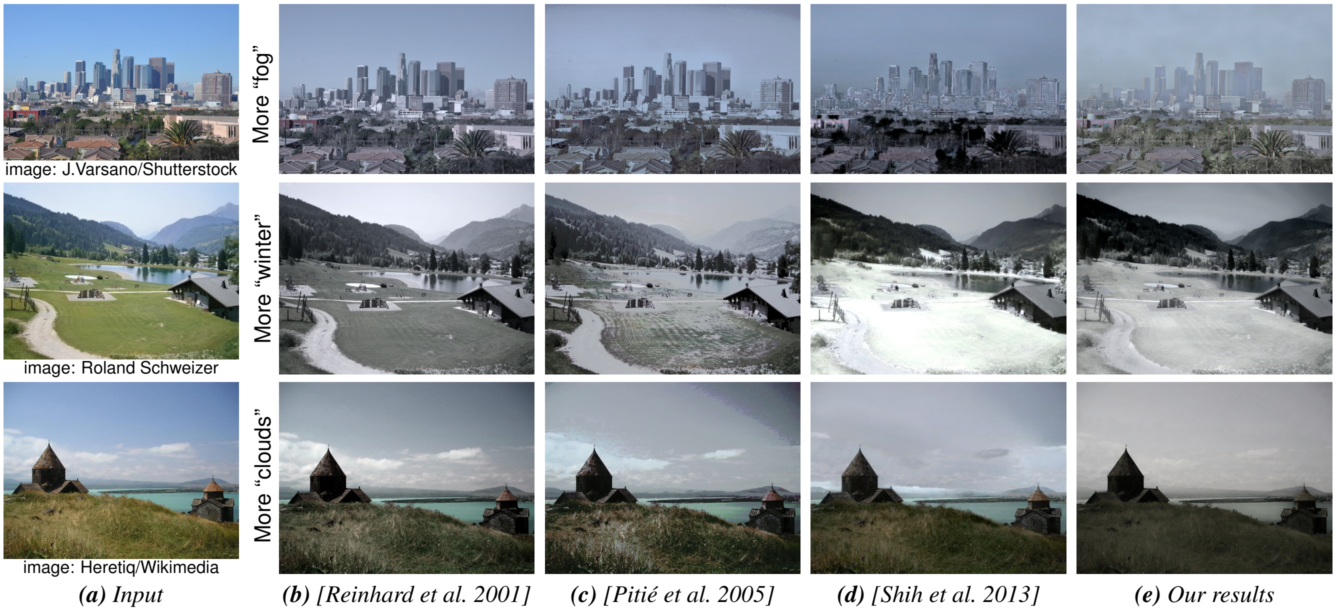
- Compared to Shih et al. [2013], our results more often look like real photographs (71%, $p < 0.005$). Both methods are equally capable of convincing attribute manipulations (52% preference towards ours, not statistically significant).



(a) *Input image* **(b)** *Match from [Shih et al. 2013]* **(c)** *Target from [Shih et al. 2013]* **(d)** *Result from [Shih et al. 2013]*

(e) *No consistency, no filtering* **(f)** *No consistency, with filtering* **(g)** *With consistency, no filtering* **(h)** *With consistency, with filtering*

Figure 8: *Intermediate results of our approach without and with global consistency / bilateral filtering (e-h). Our final result is shown in (h). The input, match, and target images (a-c) are from Shih et al. [2013] and are not obtained with our attribute-guided approach (Section 5). Note that our global consistency step addresses the low-frequency halo artifacts that are visible in their result (d) and in (e-f).*



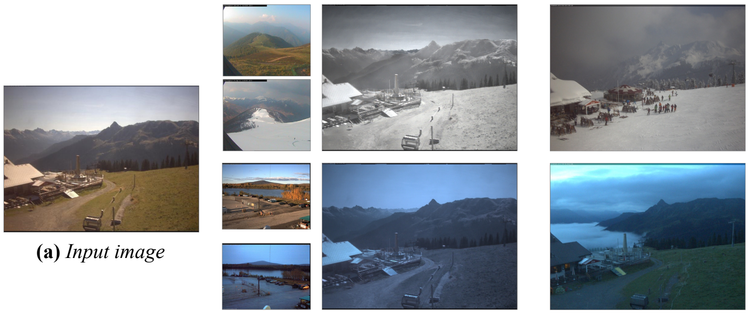
(a) *Input* ***(b)*** *[Reinhard et al. 2001]* ***(c)*** *[Pitié et al. 2005]* ***(d)*** *[Shih et al. 2013]* ***(e)*** *Our results*

Figure 9: *Comparison with appearance transfer methods for three attribute modifications. For each input photograph, all methods use the same target image, retrieved from our annotated dataset using the attribute-guided approach described in Section 5.*

Choice of target image. We show in Figure 10 that our output can be quite sensitive to the choice of target image. While we can produce results completely automatically, we argue that this simple user control is useful for exploring the space of plausible attribute manipulations. In this figure, we use an image from one of our webcams as the input; this allows us to compare to two “ground truth” images of the same webcam that have strong positive “cold” labels. These real photographs with the same attribute exhibit significant variations (b), just like the output of our appearance transfer (c). Our method allows the user to narrow down the set of target images to be used, by specifying that some attributes should not be changed (Section 5).

Performance. Our unoptimized Matlab implementation precomputes local transforms in about 20 seconds for a pair of images 480px high; we store the result for several pairs of each webcam into the transform library. Segmenting and extracting features for a new input image takes 15-20s and is done only once, before all attribute manipulations. At runtime, retrieving candidate transforms for all segments takes 2s, clustering transforms about 2s, and about 1s for filtering the result transforms and producing the output. This is faster than the method by Shih et al. [2013] which processes comparable images in 57s.

Limitations. Like all data-driven techniques, ours is limited by the richness of the training set. Because we focus on outdoor scenes, we do not expect recognition or manipulation to work well on images focused on objects (e.g., a car or a close-up façade of a single building, see Figure 11 top) or unusual scenes (e.g., pictures framed to contain no sky or overly processed photos). Our appearance transfer method based on local color changes cannot reliably add detail that did not exist in the input, e.g., add grass texture to



(a) *Input image* **(b)** *Match, target, result images* **(c)** *Real photos with "cold"*

Figure 10: *Variance in scene appearance among images with similar scene attributes. Input image (a) is chosen from one of our webcams, which allows us to show two "ground truth" images of the same scene with the "cold" attribute (c). Applying our appearance transfer method with two "cold" target images yields results that are different, yet both plausible (b).*

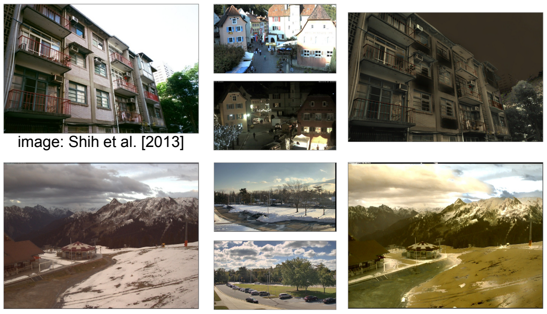


image: Shih et al. [2013]

(a) *Input images* **(b)** *Match, target, results (failure cases)*

Figure 11: *Failure cases. Top, more "night": inaccurate local matching yields artifacts in image manipulation if the input and match images are too different. Bottom, more "summer": our local transforms cannot add detail not present in the input, e.g. grass texture.*

a snow-covered ground for "more summer" (Figure 11, bottom). However, our database contains training examples for these scenarios; we expect future appearance transfer techniques to handle these transformations better. Adding high frequency clouds to a clear sky is also problematic, but might by addressed by techniques designed to transfer skies across images [Tao et al. 2009].

7 Conclusion

We have presented the first dataset containing thousands of images annotated with a number of perceived scene properties that vary with time. We have used those labels to train regressors in order to recognize transient attributes in new images, enabling new possibilities for browsing photo collections. Lastly, we have developed a simple and fast appearance transfer method which can learn from variations of appearance observed in our dataset to modify transient attributes in novel scenes. We are confident that our *Transient Attribute Database* can support future research in attribute recognition and manipulation beyond the first steps we have presented here. We share our annotated database and attribute predictors with the community on our project website:

`http://transattr.cs.brown.edu`

Acknowledgments

We thank Nathan Jacobs and Jean-François Lalonde for sharing their webcam datasets, Yichang Shih for sharing his data and code, Stephanie Mayne for the video voiceover, and the participants to our user study. This research was supported by NSF CAREER Award 1149853 to James Hays and by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory, contract FA8650-12-C-7212.

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

References

AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Comput. Graph. Forum 29*, 2.

BELL, S., UPCHURCH, P., SNAVELY, N., AND BALA, K. 2013. Opensurfaces: A richly annotated catalog of surface appearance. *ACM Trans. Graph. (proc. SIGGRAPH) 32*, 4.

BERTHOUZOZ, F., LI, W., DONTCHEVA, M., AND AGRAWALA, M. 2011. A framework for content-adaptive photo manipulation macros. *ACM Trans. Graph. 30*, 5.

BONNEEL, N., SUNKAVALLI, K., PARIS, S., AND PFISTER, H. 2013. Example-based video color grading. *ACM Trans. Graph. (proc. SIGGRAPH) 32*.

BYCHKOVSKY, V., PARIS, S., CHAN, E., AND DURAND, F. 2011. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*.

CAICEDO, J. C., KAPOOR, A., AND KANG, S. B. 2011. Collaborative personalization of image enhancement. In *CVPR*.

CHEN, J., PARIS, S., AND DURAND, F. 2007. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph. (proc. SIGGRAPH) 26*, 3.

CHENG, M.-M., ZHENG, S., LIN, W.-Y., VINEET, V., STURGESS, P., CROOK, N., MITRA, N., AND TORR, P. 2014. Imagespirit: Verbal guided image parsing. *ACM Trans. Graph.*.

COMANICIU, D., AND MEER, P. 2002. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. PAMI 24*.

CUSANO, C., GASPARINI, F., AND SCHETTINI, R. 2012. Color transfer using semantic image annotation. In *SPIE*, vol. 8299.

DALE, K., JOHNSON, M. K., SUNKAVALLI, K., MATUSIK, W., AND PFISTER, H. 2009. Image restoration using online photo collections. In *ICCV*.

DHAR, S., ORDONEZ, V., AND BERG, T. L. 2011. High level describable attributes for predicting aesthetics and interestingness. In *CVPR*.

EITZ, M., HAYS, J., AND ALEXA, M. 2012. How do humans sketch objects? *ACM Trans. Graph. (proc. SIGGRAPH) 31*, 4.

FARHADI, A., ENDRES, I., HOIEM, D., AND FORSYTH, D. 2009. Describing objects by their attributes. In *CVPR*.

FATTAL, R. 2008. Single image dehazing. *ACM Trans. Graph. (proc. SIGGRAPH) 27*, 3.

FERRARI, V., AND ZISSERMAN, A. 2007. Learning visual attributes. In *NIPS*.

GARG, R., DU, H., SEITZ, S. M., AND SNAVELY, N. 2009. The dimensionality of scene appearance. In *ICCV*.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *SIGGRAPH*.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2007. Recovering surface layout from an image. *Int. J. Comput. Vision 75*, 1.

JACOBS, N., ROMAN, N., AND PLESS, R. 2007. Consistent temporal variations in many outdoor scenes. In *CVPR*.

JOHNSON, M. K., DALE, K., AVIDAN, S., PFISTER, H., FREEMAN, W. T., AND MATUSIK, W. 2011. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Trans. Vis. Comput. Graph. 17*, 9.

KANG, S. B., KAPOOR, A., AND LISCHINSKI, D. 2010. Personalization of image enhancement. In *CVPR*.

KOPF, J., COHEN, M. F., LISCHINSKI, D., AND UYTTENDAELE, M. 2007. Joint bilateral upsampling. *ACM Trans. Graph. (proc. SIGGRAPH) 26*, 3.

KOVASHKA, A., PARIKH, D., AND GRAUMAN, K. 2012. Whittlesearch: Image search with relative attribute feedback. In *CVPR*.

KUMAR, N., BERG, A., BELHUMEUR, P., AND NAYAR, S. 2011. Describable visual attributes for face verification and image search. *IEEE Trans. PAMI 33*, 10.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., AND DRETTAKIS, G. 2012. Coherent intrinsic images from photo collections. *ACM Trans. Graph. (proc. SIGGRAPH Asia) 31*, 6.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. *ACM Trans. Graph. (proc. SIGGRAPH) 26*, 3.

LALONDE, J.-F., EFROS, A., AND NARASIMHAN, S. 2009. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Trans. Graph. (proc. SIGGRAPH Asia) 28*, 5.

LIU, Q., IHLER, A., AND STEYVERS, M. 2013. Scoring workers in crowdsourcing: how many control questions are enough? In *NIPS*.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Trans. Graph. (proc. SIGGRAPH) 22*, 3.

MURPHY, K. P. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

NARASIMHAN, S., WANG, C., AND NAYAR, S. 2002. All the images of an outdoor scene. In *ECCV*.

PARIKH, D., AND GRAUMAN, K. 2011. Relative attributes. In *ICCV*.

PATTERSON, G., AND HAYS, J. 2012. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*.

PERRONNIN, F., SÁNCHEZ, J., AND MENSINK, T. 2010. Improving the fisher kernel for large-scale image classification. In *ECCV*.

PITIÉ, F., KOKARAM, A., AND DAHYOT, R. 2005. N-Dimensional Probability Density Function Transfer and its Application to Colour Transfer. In *ICCV*.

POULI, T., AND REINHARD, E. 2011. Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl. 21*, 5.

SCHOLKOPF, B., SMOLA, A., WILLIAMSON, R., AND BARTLETT, P. 2000. New support vector algorithms. *Neural Computation 12*.

SHIH, Y., PARIS, S., DURAND, F., AND FREEMAN, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph. (proc. SIGGRAPH Asia) 32*, 6.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. (proc. SIGGRAPH) 25*, 3.

SUNKAVALLI, K., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2007. Factored time-lapse video. *ACM Trans. Graph. (proc. SIGGRAPH) 26*, 3.

TAO, L., YUAN, L., AND SUN, J. 2009. Skyfinder: Attribute-based sky image search. *ACM Trans. Graph. (proc. SIGGRAPH) 28*, 3.

WU, F., DONG, W., KONG, Y., MEI, X., PAUL, J.-C., AND ZHANG, X. 2013. Content-Based Colour Transfer. *Comput. Graph. Forum 32*, 1.

XIAO, J., HAYS, J., EHINGER, K. A., OLIVA, A., AND TORRALBA, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*.

YU, Y., AND MALIK, J. 1998. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH*.