What Makes Paris Look like Paris?

Carl Doersch
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
cdoersch@cs.cmu.edu

Saurabh Singh
University of Illinois,
Urbana-Champaign
Champaign, IL 61820
saurabh.me@gmail.com

Abhinav Gupta
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15217
abhinavg@cs.cmu.edu

Josef Sivic
Inria
Ecole Normale Supérieure
Paris, France
Josef.Sivic@ens.fr

Alexei A. Efros
University of California,
Berkeley
Berkeley, CA 94720
efros@eecs.berkeley.edu

ABSTRACT

Given a large repository of geotagged imagery, we seek to automatically find visual elements, e.g. windows, balconies, and street signs, that are most distinctive for a certain geo-spatial area, for example the city of Paris. This is a tremendously difficult task as the visual features distinguishing architectural elements of different places can be very subtle. In addition, we face a hard search problem: given all possible patches in all images, which of them are both frequently occurring *and* geographically informative? To address these issues, we propose to use a discriminative clustering approach able to take into account the weak geographic supervision. We show that geographically representative image elements can be discovered automatically from Google Street View imagery in a discriminative manner. We demonstrate that these elements are visually interpretable and perceptually geo-informative. The discovered visual elements can also support a variety of *computational geography* tasks, such as mapping architectural correspondences and influences within and across cities, finding representative elements at different geo-spatial scales, and geographically-informed image retrieval.

1. INTRODUCTION

Consider the two photographs in Figure 1, both downloaded from Google Street View. One comes from Paris, the other one from London. Can you tell which is which? Surprisingly, even for these nondescript street scenes, people who have been to Europe tend to do quite well on this task. In an informal survey, we presented 11 subjects with 100 random Street View images of which 50% were from Paris, and the rest from eleven other cities. We instructed the subjects (who have all been to Paris) to try and ignore any text in the photos, and collected their binary forced-choice responses (Paris / Not Paris). On average, subjects were correct 79% of the time ($std = 6.3$), with chance at 50% (when allowed to scrutinize the text, performance for some subjects went up as high as 90%). What this suggests is that people are remarkably sensitive to the geographically-informative features within the visual environment. But what are those features? In informal debriefings, our subjects suggested that for most images, a few localized, distinctive elements "immediately gave it away". E.g. for Paris, things like windows with railings, the particular style of balconies, the distinctive doorways, the traditional blue/green/white street signs, etc. were particularly helpful. Finding those features can be difficult though, since every image can contain more than $25,000$ candidate patches, and only a tiny fraction will be truly distinctive.

In this work, we want to find such local geo-informative features *automatically*, directly from a large database of photographs from a particular place, such as a city. Specifically, given tens of thousands of geo-localized images of some geographic region R, we aim to find a few hundred visual elements that are both: 1) repeating, i.e. they occur often in R, and 2) geographically discriminative, i.e. they occur much more often in R than in R^C. Figure 1 shows sample output of our algorithm: for each photograph we show three of the most geo-informative visual elements that were automatically discovered. For the Paris scene (left), the street sign, the window with railings, and the balcony support are all flagged as informative.

But why is this topic important for modern computer graphics? 1) Scientifically, the goal of understanding which visual elements are fundamental to our perception of a complex visual concept, such as a place, is an interesting and useful one. Our paper shares this motivation with a number of other recent works that don't actually synthesize new visual imagery, but rather propose ways of finding and visualizing existing image data in better ways, be it selecting candid portraits from a video stream [5], summarizing a scene from photo collections [19], finding iconic images of an object [1], etc. 2) More practically, one possible future application of the ideas presented here might be to help CG modelers by generating so-called "reference art" for a city. For instance, when modeling Paris for PIXAR's *Ratatouille*, the co-director Jan Pinkava faced exactly this problem: "The basic question for us was: 'what would Paris look like as a model of Paris?', that is, what are the main things that give the city

The original version of this paper was published in SIGGRAPH, 2012.

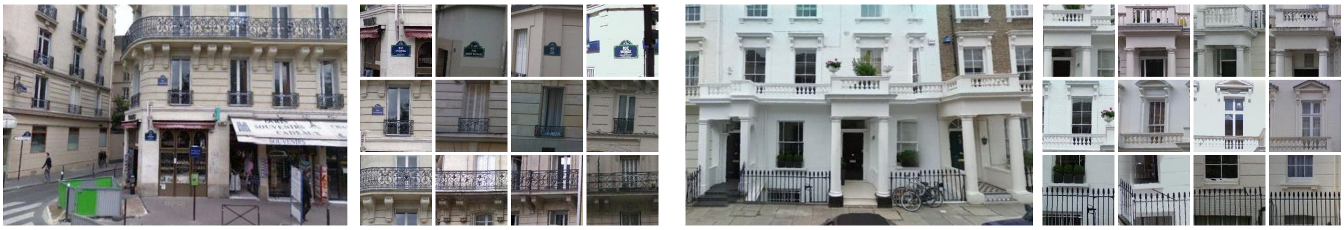


Figure 1: These two photos might seem nondescript, but each contains hints about which city it might belong to. Given a large image database of a given city, our algorithm is able to automatically discover the geographically-informative elements (patch clusters to the right of each photo) that help in capturing its "look and feel". On the left, the emblematic street sign, a balustrade window, and the balcony support are all very indicative of Paris, while on the right, the neoclassical columned entryway sporting a balcony, a Victorian window, and, of course, the cast iron railing are very much features of London.

its unique look?" [14]. Their solution was to "run around Paris for a week like mad tourists, just looking at things, talking about them, and taking lots of pictures" not just of the Eiffel Tower but of the many stylistic Paris details, such as signs, doors etc. [14](see photos on pp.120–121). But if going "on location" is not feasible, our approach could serve as basis for a detail-centric reference art retriever, which would let artists focus their attention on the most statistically significant stylistic elements of the city. 3) And finally, more philosophically, our ultimate goal is to provide a *stylistic narrative* for a visual experience of a place. Such narrative, once established, can be related to others in a kind of geo-cultural visual reference graph, highlighting similarities and differences between regions. E.g. one could imagine finding a visual appearance "trail" from Greece, through Italy and Spain and into Latin America. In this work, we only take the first steps in this direction – connecting visual appearance across cities, finding similarities within a continent, and differences between neighborhoods. But we hope that our work might act as a catalyst for research in this new area, which might be called *computational geo-cultural modeling.*

2. PRIOR WORK

In the field of architectural history, descriptions of urban and regional architectural styles and their elements are well established. Such local elements and rules for combining them have been used in computer systems for procedural modeling of architecture to generate 3D models of entire cities in an astonishing level of detail, e.g. [12], or to parse images of facades, e.g. [22]. However, such systems require significant manual effort from an expert to specify the appropriate elements and rules for each architectural style.

At the other end of the spectrum, data-driven approaches have been leveraging the huge datasets of geotagged images that have recently become available online. For example, Crandall et al. [2] use the GPS locations of 35 thousand consumer photos from Flickr to plot photographer-defined frequency maps of cities and countries. Geo-tagged datasets have also been used for place recognition [17, 8] including famous landmarks [10, 11]. Our work is particularly related to [17, 8], where geotags are also used as a *supervisory signal* to find sets of image features discriminative for a particular place. While these approaches can work very well, their image features typically cannot generalize beyond matching specific buildings imaged from different viewpoints. Alternatively, global image representations from scene recognition, such as GIST descriptor [13] have been used for geo-localization of generic scenes on the global Earth scale [6, 7]. There, too, reasonable recognition performance has been achieved, but the use of global descriptors makes it hard for a human to interpret *why* a given image gets assigned to a certain location.

Finally, our paper is related to a line of work on unsupervised object discovery [16, 20] (and especially [15], who also deal with mining geo-tagged image data). Such methods attempt to explicitly discover features or objects which occur frequently in many images and are also useful as human-interpretable elements of visual representation. But being unsupervised, these methods are limited to only discovering things that are both very common and highly visually consistent.

In contrast, here we propose a discovery method that is weakly constrained by location labels derived from GPS tags, and which is able to mine representative visual elements automatically from a large online image dataset. Not only are the resulting visual elements geographically discriminative (i.e. they occur only in a given locale), but they also typically look meaningful to humans, making them suitable for a variety of geo-data visualization applications. The next section describes the data used in this work, followed by the full description of our algorithm.

3. THE DATA

Flickr has emerged as the data-source of choice for most recently developed data-driven applications in computer vision and graphics, including visual geo-location [6, 2, 11]. However, the difficulty with Flickr and other consumer photo-sharing websites for geographical tasks is that there is a strong data bias towards famous landmarks. To correct for this bias and provide a more uniform sampling of the geographical space, we turn to Google Street View – a huge database of street-level imagery, captured as panoramas using specially-designed vehicles. This enables extraction of roughly fronto-parallel views of building facades and, to some extent, avoids dealing with large variations of camera viewpoint.

Given a geographical area on a map, we automatically scrape a dense sampling of panoramas of that area from Google Street View. From each panorama, we extract two perspective images (936x537 pixels), one on each side of the capturing vehicle, so that the image plane is roughly parallel to the vehicle's direction of motion. This results in approximately 10,000 images per city. For this project, we downloaded 12 cities: Paris, London, Prague, Barcelona, Milan, New York, Boston, Philadelphia, San Francisco, San Paulo, Mexico City, and Tokyo.

4. DISCOVERING GEO-INFORMATIVE ELEMENTS

Our goal is to discover visual elements which are characteristic of a given geographical locale (e.g. the city of Paris). That is, we seek patterns that are both *frequently occurring* within the given locale, **and** *geographically discriminative*, i.e. they appear in that locale and do not appear elsewhere. Note that neither of these two requirements by itself is enough: sidewalks and cars occur frequently in Paris but are hardly discriminative, whereas the Eiffel Tower is very discriminative, but too rare to be useful ($< 0.0001\%$ in our data). In this work, we will represent visual elements by square image patches at various resolutions, and mine them from our large image database. The database will be divided into two parts: (i) the positive set containing images from the location whose visual elements we wish to discover (e.g. Paris); and (ii) the negative set containing images from the rest of the world (in our case, the other 11 cities in the dataset). We assume that many frequently occurring but uninteresting visual patterns (trees, cars, sky, etc.) will occur in both the positive and negative sets, and should be filtered out. Our biggest challenge is that the overwhelming majority of our data is uninteresting, so matching the occurrences of the rare interesting elements is like finding a few needles in a haystack.

One possible way to attack this problem would be to first discover repeated elements and then simply pick the ones which are the most geographically discriminative. A standard technique for finding repeated patterns in data is clustering. For example, in computer vision, "visual word" approaches [21] use k-means clustering on image patches represented by SIFT descriptors. Unfortunately, standard visual words tend to be dominated by low-level features, like edges and corners (Figure 2a), not the larger visual structures we are hoping to find. While we can try clustering using larger image patches (with a higher-dimensional feature descriptor, such as HOG [3]), k-means behaves poorly in very high dimensions, producing visually inhomogeneous clusters (Figure 2b). We believe this happens because k-means (and similar approaches) partition the *entire* feature space. This tends to lose the needles in our haystack: the rare discriminative elements get mixed with, and overwhelmed by, less interesting patches, making it unlikely that a distinctive element could ever emerge as its own cluster.

In this paper, we propose an approach that avoids partitioning the entire feature space into clusters. Instead, we start with a large number of randomly sampled candidate patches, and then give each candidate a chance to see if it can converge to a cluster that is both frequent and discriminative. We first compute the nearest neighbors of each candidate, and reject candidates with too many neighbors in the negative set. Then we gradually build clusters by applying iterative discriminative learning to each surviving candidate. The following section presents the details of this algorithm.

4.1 Our Approach

From the tens of millions of patches in our full positive set, we randomly sample a subset of 25,000 high-contrast patches to serve as candidates for seeding the clusters. Throughout the algorithm, we represent such patches using a HOG+color descriptor. First, the initial geo-informativeness of each patch is estimated by finding the top 20 nearest neighbor (NN) patches in the full dataset (both positive and negative), measured by normalized correlation, and counting how many of them come from Paris. Figure 3 shows nearest neighbors for a few randomly-selected patches and for the patches whose neighbors all come from Paris. Note that the latter patches are not only more Parisian, but also considerably more coherent. This is because generating a coherent cluster is a prerequisite to retrieving matches exclusively from Paris: any patch whose matches are incoherent will likely draw those matches randomly from inside and outside Paris. We keep the candidate patches that have the highest proportion of their nearest neighbors in the positive set, while also rejecting near-duplicate patches (measured by spatial overlap of more than 30% between any 5 of their top 50 nearest neighbors). This reduces the number of candidates to about 1000.

Some good elements, however, get matched incorrectly during the nearest-neighbors phase. Figure 4 shows a patch that contains both a street sign and a vertical bar on the right (the end of the facade). The naïve distance metric doesn't know what's important, and so it tries to match both. Yet too few such patches exist in the dataset; for the remainder, the algorithm matches the vertical bar simply because it's more frequent. To fix this problem, we aim to learn a distance metric that gives higher weight to the features that make the patch geo-discriminative.

Recently, [18] showed how one can improve visual retrieval by adapting the distance metric to the given query using discriminative learning. We adopt similar machinery, training a linear SVM detector for each visual element in an iterative manner as in [20]. Unlike these previous works, however, we emphasize that the weak labels are the workhorse of the distance learning. In the case of Figure 4, for example, we know that the street sign is more important because it occurs only in Paris, whereas the vertical bar occurs everywhere. We train an SVM detector for each visual element, using the top k nearest neigbors from the positive set as positive examples, and all negative-set patches as negative examples. While this produces a small improvement (Figure 5, row 2), it is not enough, since the top k matches might not have been very good to begin with. So, we iterate the SVM learning, using the top k detections from previous round as positives (we set $k = 5$ for all experiments). The idea is that with each round, the top detections will become better and better, resulting in a continuously improving detector. However, doing this directly would not produce much improvement because the SVM tends to overfit to the initial positive examples [20], and will prefer them in each next round over new (and better) ones. Therefore, we apply cross-validation by dividing both the positive and the negative parts of the dataset into l equally-sized subsets (we set $l = 3$ for all experiments). At each iteration of the training, we apply the detectors trained on the previous round to a new, *unseen* subset of data to select the top k detections for retraining. In our experiments, we used three iterations, as most good clusters didn't need more to converge (i.e. stop changing). After the final iteration, we rank the resulting detectors based on their accuracy: per-

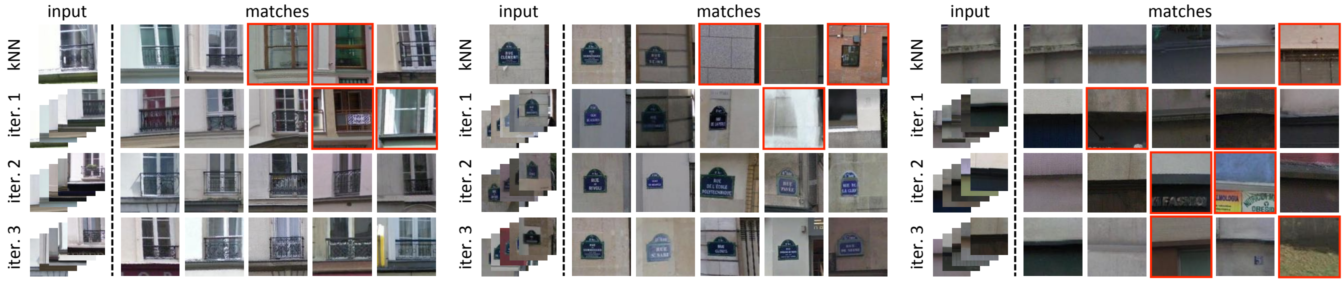
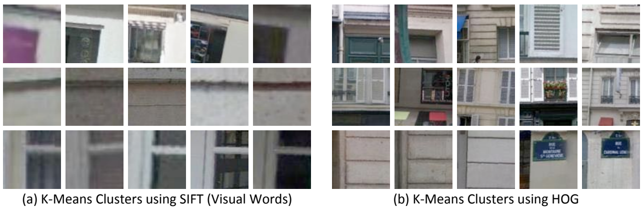


Figure 5: Steps of our algorithm for three sample candidate patches in Paris. The first row: initial candidate and its NN matches. Rows 2-4: iterations of SVM learning (trained using patches on left). Red boxes indicate matches outside Paris. Rows show every 7th match for clarity. Notice how the number of not-Paris matches decreases with each iteration, except for rightmost cluster, which is eventually discarded.


(a) K-Means Clusters using SIFT (Visual Words)

(b) K-Means Clusters using HOG

Figure 2: (a) k-means clustering using SIFT (visual words) is dominated by low level features. (b) k-means clustering over higher dimensional HOG features produces visually incoherent clusters.

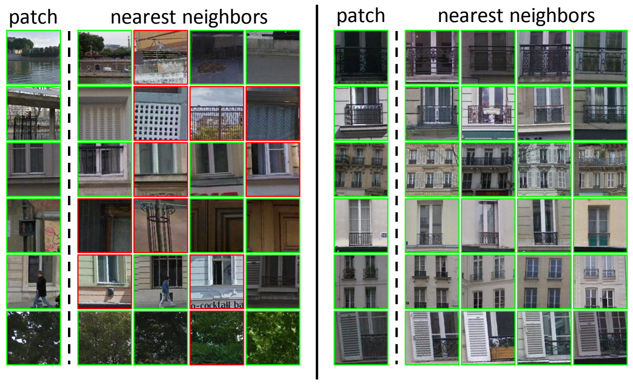


Figure 3: Left: randomly-sampled candidate patches and their nearest neighbors according to a standard distance metric. Right: after sorting the candidates by the number of retrieved neighbors that come from Paris, coherent Parisian elements have risen to the top.

centage of top 50 firings that are in the positive dataset (i.e. in Paris). We return the top few hundred detectors as our geo-informative visual elements.

Figure 5 illustrates the progression of these iterations. For example, in the left column, the initial nearest neighbors contain only a few windows with railings. However, windows with railings differ more from the negative set than the windows without railings; thus the detector quickly becomes more sensitive to them as the algorithm progresses. The right-most example does not appear to improve, either in visual similarity or in geo-discriminativeness. This is because the original candidate patch was intrinsically not very geo-informative and would not make a good visual element. Such patches have a low final accuracy and are discarded.

Implementation Details: Our current implementation considers only square patches (although it would not be difficult to add other aspect ratios), and takes patches at scales ranging from 80-by-80 pixels all the way to height-of-image size. Patches are represented with standard HOG [3] (8x8x31 cells), plus a 8x8 color image in L*a*b colorspace (a and b only). Thus the resulting feature has 8x8x33 = 2112 dimentions. During iterative learning, we use a soft-margin SVM with C fixed to 0.1. The full mining computation is quite expensive; a single city requires approximately $1,800$ CPU-hours. But since the algorithm is highly parallelizable, it can be done overnight on a cluster.

4.2 Results and Validation

Figure 6 shows the results of running our algorithm on several well-known cities. For each city, the left column shows randomly chosen images from Google Street View, while the right column shows some of the top-ranked visual element clusters that were automatically discovered (due to space limitations, a subset of elements was selected manually to show variety; see the project webpage for the full list). Note that for each city, our visual elements convey a better stylistic feel of the city than do the random images. For example, in Paris, the top-scoring elements zero-in on some of the main features that make Paris look like Paris: doors, balconies, windows with railings, street signs and special Parisian lampposts. It is also interesting to note that, on the whole, the algorithm had more trouble with American cities: it was able to discover only a few geo-informative elements, and some of them turned out to be different brands

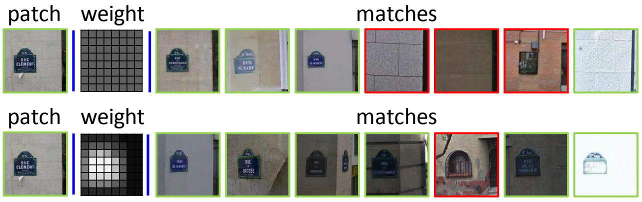
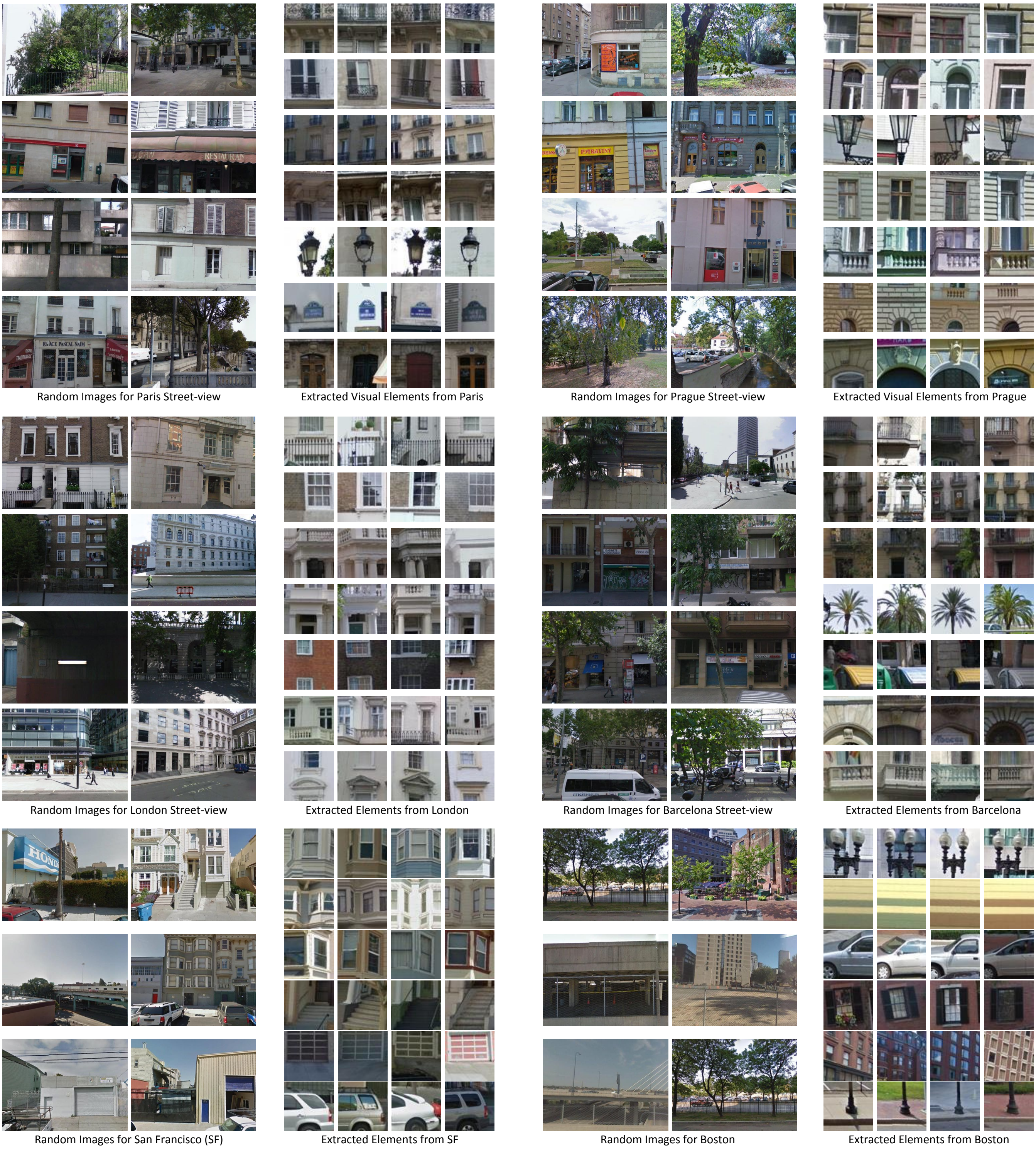


Figure 4: Top: using the naïve distance metric for this patch retrieves some good matches and some poor matches, because the patch contains both a street sign and a vertical bar on the right. Bottom: our algorithm reweights the dimensions of our patch descriptor to separate Paris from non-Paris. The algorithm learns that focusing on the street sign achieves maximum separation from the non-Paris walls.



Random Images for Paris Street-view

Extracted Visual Elements from Paris

Random Images for Prague Street-view

Extracted Visual Elements from Prague

Random Images for London Street-view

Extracted Elements from London

Random Images for Barcelona Street-view

Extracted Elements from Barcelona

Random Images for San Francisco (SF)

Extracted Elements from SF

Random Images for Boston

Extracted Elements from Boston

Figure 6: Google Street View vs. geo-informative elements for six cities. Arguably, the geo-informative elements (right) are able to provide better stylistic representation of a city than randomly sampled Google Street View images (left).

of cars, road tunnels, etc. This might be explained by the relative lack of stylistic coherence and uniqueness in American cities (with its melting pot of styles and influences), as well as the supreme reign of the automobile on American streets.

In addition to the qualitative results, we would also like to provide a more quantitative evaluation of our algorithm. While validating data-mining approaches is difficult in general, it is possible to measure 1) to what extent our elements are specific to particular locations, and 2) do users find them

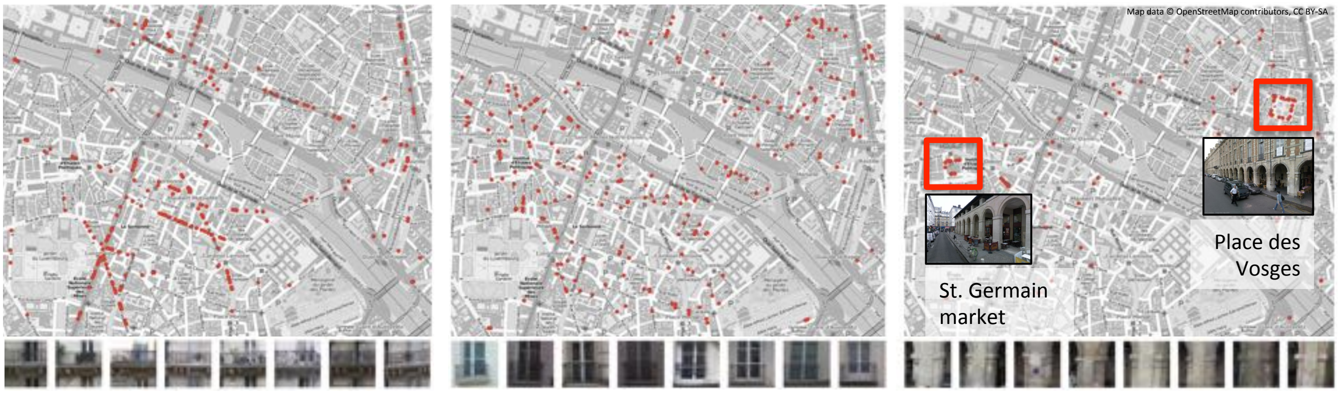


Figure 7: Examples of geographic patterns in Paris (shown as red dots on the maps) for three discovered visual elements (shown below each map). Balconies with cast-iron railings are concentrated on the main boulevards (left). Windows with railings mostly occur on smaller streets (middle). Arch supporting columns are concentrated on Place des Vosges and the St. Germain market (right).

subjectively geo-informative in a visual discrimination task?

To evaluate how geo-informative our visual elements are, we ran the top 100 Paris element detectors over an unseen dataset which was 50% from Paris and 50% from elsewhere. For each element, we found its geo-informativeness by computing the percentage of the time it fired in Paris out of the top 100 firings. The average accuracy of our top detectors was 83% (where chance is 50%). We repeated this for our top 100 Prague detectors, and found the average accuracy on an unseen dataset of Prague to be 92%.

Next, we repeated the above experiment with people rather than computers. To avoid subject fatigue, we reduced the dataset to 100 visual elements, 50 from Paris and 50 from Prague. 50% of the elements were the top-ranked ones returned by our algorithm for Paris and Prague. The other 50% were randomly sampled patches of Paris and Prague (but biased to be high-contrast, as before, to avoid empty sky patches, etc). In a web-based study, subjects (who have all been to Paris but not necessarily Prague) were asked to label each of the 100 patches as belonging to either Paris or Prague (forced choice). The results of our study (22 naive subjects) are as follows: average classification performance for the algorithm-selected patches was 78.5% ($std = 11.8$), while for random patches it was 58.1% ($std = 6.1$); the p-value for a paired-samples t-test was $< 10^{-8}$. While on random patches subjects did not do much better than chance, performance on our geo-informative elements was roughly comparable to the much simpler full-image classification task reported in the beginning of the paper (although since here we only used Prague, the setups are not quite the same).

5. APPLICATIONS

Now that we have a tool for discovering geographically-informative visual elements for a given locale, we can use them to explore ways of building stylistic narratives for cities and of making visual connections between them. Here we discuss just a few such directions.

5.1 Mapping Patterns of Visual Elements

So far, we have shown the discovered visual elements for a given city as an ordered list of patch clusters (Figure 6). Given that we know the GPS coordinates of each patch, however, we could easily display them on a map, and then search for interesting geo-spatial patterns in the occurrences of a given visual element. Figure 7 shows the geographical locations for the top-scoring detections for each of 3 different visual elements (a sampling of detections are shown below each map), revealing interestingly non-uniform distributions. For example, it seems that balconies with cast-iron railings (left) occur predominantly on the large thoroughfares (bd Saint-Michel, bd Saint-Germain, rue de Rivoli), whereas windows with cast-iron railings (middle) appear mostly on smaller streets. The arch-supporting column (right) is a distinguishing feature of the famous Place des Vosges, yet it also appears in other parts of Paris, particularly as part of more recent Marché Saint-Germain (this is a possible example of so-called "architectural citation"). Automatically discovering such architectural patterns may be useful to both architects and urban historians.

5.2 Visual Correspondences Across Cities

Given a set of architectural elements (windows, balconies, etc.) discovered for a particular city, it is natural to ask what these same elements might look like in other cities. As it turns out, a minor modification to our algorithm can often accomplish this task. We have observed that a detector for a location-specific architectural element will often fire on functionally similar elements in other cities, just with a much lower score. That is, a Paris balcony detector will return mostly London balconies if it is forced to run only on London images. Naturally these results will be noisy, but we can clean them up using an iterative learning approach similar to the one in Section 4.1. The only difference is that we require the positive patches from each iteration of training to be taken not just from the source city, but from all the cities where we wish to find correspondences. For example, to find correspondences between Paris, Prague, and London, we initialize with visual elements discovered in Paris and then, at each round of "clean-up" training, we use 9 top positive matches to train each element SVM, 3 from each of the three cities. Figure 8 illustrates the result of this procedure. Note how capturing the correspondence between similar visual elements across cities can often highlight certain stylistic differences, such as the material for the balconies, the style of the street-lamps, or the presence and position of ledges on the facades.

5.3 Visualizing Facade Layout

Another interesting observation is that some discovered

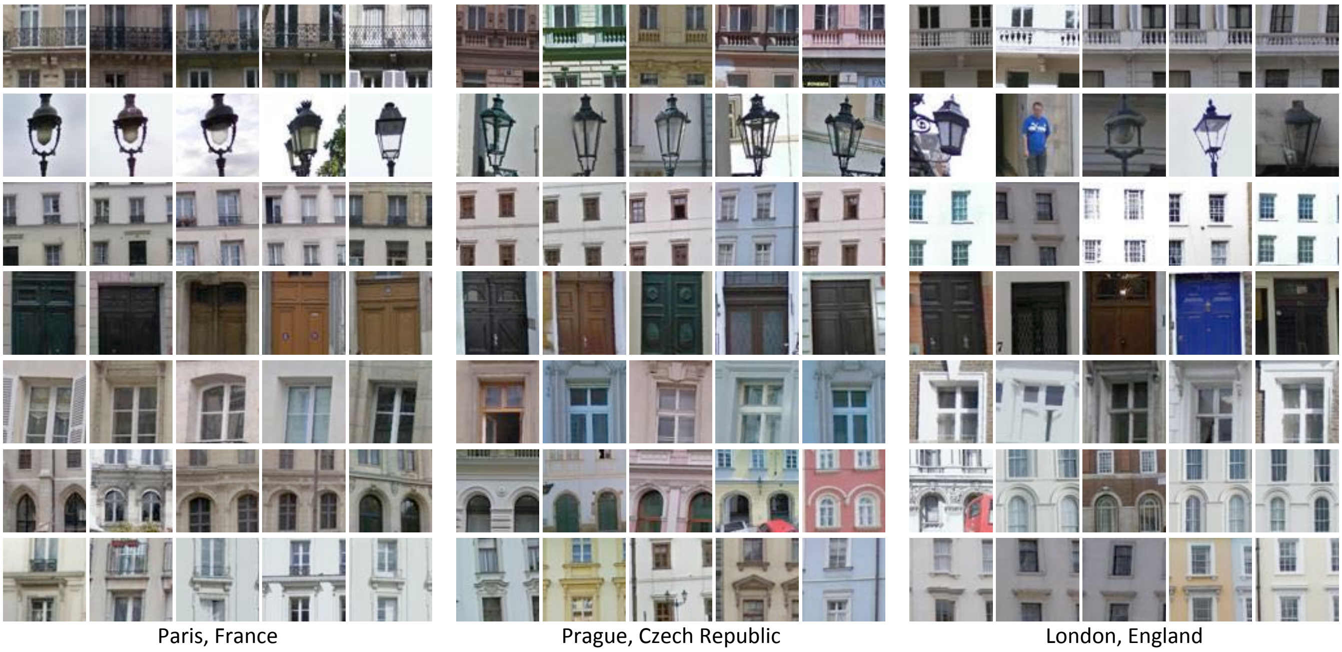


Figure 8: Visual Correspondence. Each row shows corresponding detections of a single visual element detector across three different cities.

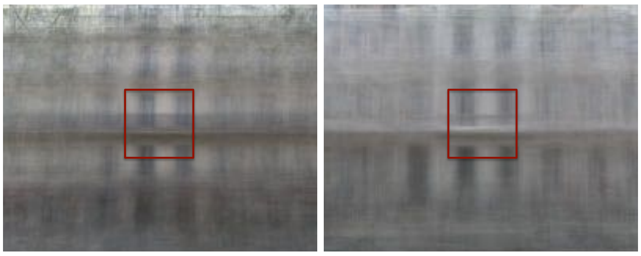


Figure 9: Object-centric image averages for the element detector in the top row of Figure 8. Note how the context captures the differences in facade styles between Paris (left) and London (right).

visual elements, despite having a limited spatial extent, can often encode a much larger architectural context. This becomes particularly apparent when looking at the same visual element detector applied in different cities. Figure 9 shows object-centric averages (in the style of [23]) for the detector in the top row of Figure 8 for Paris and London. That is, for each city, the images with the top 100 detections of the element are first centered on that element and then averaged together in image space. Note that not only do the average detections (red squares) look quite different between the two cities, but the average contexts reveal quite a lot about the differences in the structure and style of facades. In Paris, one can clearly see four equal-height floors, with a balcony row on the third floor. In London, though, floor heights are uneven, with the first floor much taller and more stately.

6. CONCLUSION

So, what makes Paris look like Paris? We argued that the "look and feel" of a city rests not so much on the few famous landmarks (e.g. the Eiffel Tower), but largely on a set of stylistic elements, the visual minutiae of daily urban life. We proposed a method that can automatically find a subset of such visual elements from a large dataset offered by Google Street View, and demonstrated some promising applications. This work is but a first step towards our ultimate goal of providing stylistic narratives to explore the diverse visual geographies of our world. Currently, the method is limited to discovering only local elements (image patches), so a logical next step would be trying to capture larger structures, both urban (e.g. facades), as well as natural (e.g. fields, rivers). Finally, the proposed algorithm is not limited to geographic data. Figure 10 shows promising results for mining discriminative patches on indoor scenes, and cars, suggesting that visual elments can be a useful tool for exploring a wide variety of image data domains.

7. REFERENCES

[1] T. Berg and A. Berg. Finding iconic images. In *The 2nd Internet Vision Workshop at Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[2] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pages 761–770, 2009.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[4] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, pages 494–502, 2013.

[5] J. Fiss, A. Agarwala, and B. Curless. Candid portrait selection from video. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 30(6):128, 2011.

[6] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *IEEE Conference*

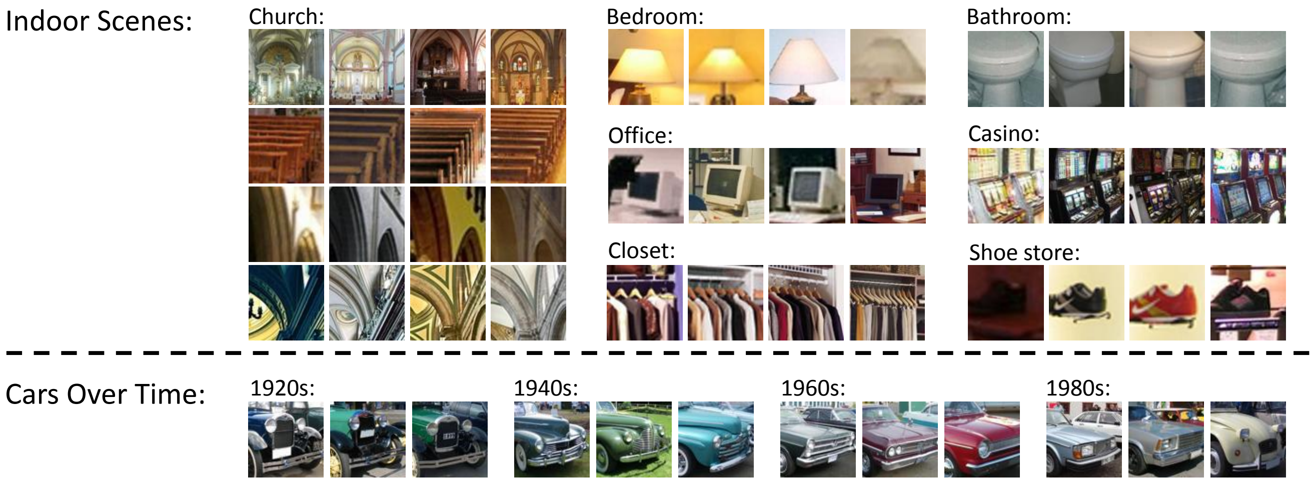


Figure 10: Our algorithm applied to other data sources. Top: elements for indoor scenes, where the weak label is one of 67 indoor scene categories [4, 20]. Bottom: stylistic elements that differentiate cars from different decades [9].

on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008.

[7] E. Kalogerakis, O. Vesselova, J. Hays, A. Efros, and A. Hertzmann. Image sequence geolocation with human travel priors. In *IEEE 12th International Conference on Computer Vision (ICCV)*, pages 253–260, 2009.

[8] J. Knopp, J. Sivic, and T. Pajdla. Avoiding confusing features in place recognition. In *European Conference on Computer Vision (ECCV)*, pages 748–761, 2010.

[9] Y. J. Lee, A. A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *IEEE 14th International Conference on Computer Vision (ICCV)*, pages 1857–1864. IEEE, 2013.

[10] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *European Conference on Computer Vision (ECCV)*, pages 427–440, 2008.

[11] Y. Li, D. Crandall, and D. Huttenlocher. Landmark classification in large-scale image collections. In *IEEE 12th International Conference on Computer Vision (ICCV)*, pages 1957–1964, 2009.

[12] P. Mueller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Transactions on Graphics (SIGGRAPH)*, 25(3):614–623, 2006.

[13] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[14] K. Paik. *The Art of Ratatouille*. Chronicle Books, 2006.

[15] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proceedings of the International Conference on Content-based Image and Video Retrieval (CIVR)*, pages 47–56, 2008.

[16] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1605–1614, 2006.

[17] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007.

[18] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 30(6):154, 2011.

[19] I. Simon, N. Snavely, and S. M. Seitz. Scene summarization for online image collections. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[20] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. pages 73–86, 2012.

[21] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE 9th International Conference on Computer Vision (ICCV)*, pages 1470–1477, 2003.

[22] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3105–3112, 2010.

[23] A. Torralba and A. Oliva. Statistics of natural image categories. *Network: Computation in Neural Systems*, pages 391–412, 2003.