Parsing Images of Architectural Scenes

Alexander C. Berg[0] Floraine Grabler[0] Jitendra Malik
Yahoo! Research ETH Zurich EECS Department
Santa Clara, USA Zurich, Switzerland U.C. Berkeley, USA
{aberg,floraine,malik}@eecs.berkeley.edu

Abstract

We address image parsing in the setting of architectural scenes. Our goal is to parse an image into regions of various types such as sky, foliage, buildings, and street. Furthermore we parse the building regions at a finer level of detail, identifying the positions of windows, doors, and rooflines, the colors of walls, and the spatial extent of particular buildings. Recognizing these individual elements is often impossible without the context provided by the initial parsing of the image, for instance a roofline is only defined in relation to the building below and the sky above. Our approach is driven by recognition of generic classes of visual appearance, e.g. *for foliage. The generic recognition results boot-strap an image specific model that provides refined estimates to use for matting, segmentation, and more detailed parsing.*

1. Introduction

> I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have ”327”? No. I have sky, house, and trees.
>
> – Max Wertheimer 1923[1]

Max Wertheimer, the founding father of the Gestalt movement, began his classic paper on visual grouping by referring to a common everyday occurrence, which has nevertheless proved quite challenging for computer vision research. How is it that we interpret images such as Figures 1 and 6 segmenting them into sky, foliage and building, and further parsing the buildings into roof, wall, windows, doors etc. A generic approach to this specific problem is the primary contribution of this paper.

First, some motivation. Visual recognition, specifically, object recognition typically aims at developing techniques independent of the particular classes of objects. Surely, we do not wish to spend years writing papers on motorcycle detection, tiger detection, armadillo detection et cetera ad nauseam! However, some object classes are particularly important because they have additional structure and utility that merit special consideration. Faces, and human figures in general, are one example; handwriting is another. We submit that architectural scenes are another such category. Buildings serve as visual landmarks within an environment. As a result directions are typically given in terms of buildings (*i.e.* the house just past the red building) and icons of distinctive buildings appear in tourist maps. Similarly when buying real estate buyers may shop for houses with particu-

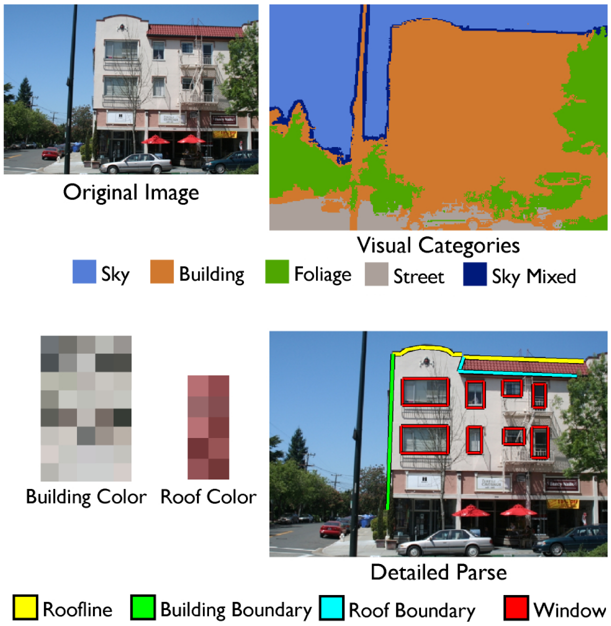

Figure 1. **Top:** *We begin by parsing the original image into five visual categories (sky, building, foliage, street and sky-mixed).* **Bottom:** *We then perform a detailed parse to compute the roofline, building and roof boundaries, and windows. In addition we estimate color models for the walls of the building and the roof of the building.*

[0] Visiting U.C. Berkeley
[1] Translated from German.

lar features (i.e. certain types of windows, specific types of colors, etc.).

Parsing the structure of buildings such as those depicted in figure 1 is essential for these applications. The parsing performance, while not necessarily perfect, has to be sufficiently good, and the technique has to be applicable to a reasonable range of scenes without significant "hand-holding". We are not aware of any current work which attains this goal. Note that while there has been a fair amount of research on interpreting aerial views of urban scenes [12], those techniques don't naturally carry over to the street-level views that we are concerned with in this paper.

We situate our work as following in the broad tradition of context based scene analysis, pioneered in recent years by Tribal *et al.* [17], and more recently also by Hoiem *et al.* [11, 10, 9][2]. We distinguish our work from that in the following respect. We aim to obtain higher performance by restricting the domain to buildings (this is quantified with respect to Hoiem *et al.* in Table 1) and in addition we want to parse out detailed structures on buildings. The 3D interpretation aspects of that work, using vanishing points, horizon etc, are complementary to our analysis, which looks more carefully at the pictorial structure of views of buildings. To emphasize a distinction with another line of work, it is neither our aim to build 3D reconstructions in the style of FACADE [6] or Criminisi *et al.* [5], nor to require 3D information for finding detailed structures as in [15]. Our work could be a pre-processing step for such systems.

2. Approach and Modeling

We approach parsing as a recognition problem both at the coarse level of street, foliage, building, sky, and at the detailed level of window, door, etc. Parsing proceeds in stages and is loosely arranged around an underlying conditional random field model [14, 13].

The first theme in our approach is that most features are computed in a fixed window around each pixel, avoiding the use of segmentation as a preprocessing step. Spatial smoothness in the labels is enforced in two ways, first by explicitly computing a color model for the particular sky and buildings present in an image, and second with a local spatial smoothness term in the model.

Finding detailed structures is intimately tied to coarse classification of the image pixels into visual categories. To this end, after an initial estimate of the spatial support of the building is determined, an initial pass of roofline and window detection is used to refine the estimate. This is in turn used to refine an estimate of roofline and window location. This particular heuristic for inference again emphasizes the importance of some global parameters – it is initially more useful to know that a building is salmon colored (enforcing long range constraints) than to know that nearby pixels probably share the same label (short range constraints).

Next we present the structure of the probabilistic model and an outline of training. A detailed description of the features used in modeling is found in Section 3 and includes more information about training. The parsing procedure is described later in Section 4 in order to permit discussion of some relevant detail.

2.1. Probabilistic Model

Our system is built around a conditional random field model:

$$p(L|I) \propto \prod_i g_i \qquad (1)$$

With labels, $L = \{l_i\}$, an image I, and potential functions g_i. There are two types of labels, l_i^p a "coarse" label for each pixel in the image, and l_k^d a "detail" label for each potential detail feature such as window, door, etcetera in the image. The per pixel, coarse labels, l_i^p are modeled using multinomial distributions over the labels: street, foliage, building, sky, and mixed sky[3]. The detail labels l_k^d are instantiated for each potential detail detected in the image, and have parameters such as location (like the per pixel labels) but also size, type (window, door, roof, etc), color if appropriate, and a probability. The two types of labels have different, adaptive, neighborhood structures as discussed in Sections 3 and 4.1. The potential functions g_i come in four flavors, for coarse or detailed labels, and either image features or surrounding labels in the appropriate neighborhood.

Instead of modeling probabilities as dependent on the image, I, directly, they are modeled in terms of features $f_m(I)$ computed from the image. Descriptions for each feature are found in Section 3.

2.2. Training

The models are trained using a set of hand labeled examples. The basic subroutine in the learning stage is to count the number of times a certain label co-occurs with a certain feature value combination as an empirical estimate of $p(l_i|f_m)$. The specific sets of features considered are discussed in Section 3. We use two types of models for $p(l_i|f_m)$, either a k nearest neighbor density estimate, or support vector regression, depending on the particular set of features. The prior distribution over visual categories is taken to be uniform, and features are modeled as independent. Given the large neighborhoods exact training and inference would be very expensive, and using simply learned

[2] Also see more sampling heavy work on reconstruction from a single image by Han and Zhu [8, 7]. Our work could be seen as providing a good proposal strategy for that style of approach or the work in [18].

[3] *mixed sky* "catches" parts of image that have a bit of sky and of building or foliage, but do not look like either one alone. This is especially important with patch based features.

models is sufficient to illustrate the technique, while providing potential room for improvement.

3. Features for Parsing

Features are computed throughout the image and are used to determine:

$$p(l_i|I,\Theta) \propto \prod_m p(l_i|f_m)$$

with the two simplifying assumptions in the previous section.

In the following subsections we present the basic features and list the combinations of basic features used to make up the f_m as well as the model used for $p(l_i|f_m)$. The coarse label for a pixel depends directly on features computed in a square window directly around that pixel (as well as, of course, on the labels of surrounding pixels). The size of the window is 21×21 pixels for a 400×600 image and scales with the image diagonal.

Additional features are relevant to detecting the detailed structures such as windows and doors on buildings. These are combinations of the contour features discussed below into right angles and t-junctions. Section 4 includes additional description of the detailed parsing stage.

3.1. Color Histogram Feature

The colors in a patch are vector quantized in L.a.b. color space with the Euclidean norm using k-means with 10 centers. Histograms are compared using Mallow's distance (aka earth mover's distance [16]) where the distance between points is the euclidean distance in L.a.b. space.

3.2. Contour Feature

Generally the structure of edges is a useful feature for parsing images of buildings, (cf the Manhattan world assumption [4]). Contours are detected by local non-max suppression on the output of the Gaussian quadrature based elongated odd edge filters at 8 orientations. These are then represented by line segments constrained to be within 2 pixels of the contour. The features are a histogram of how many edge segments are present at each of 8 orientations, and how many edge segments have lengths falling into each of five log spaced bins by length.

3.3. Texturedness Feature

We model texture as simply as possible by just aggregating the total amount of edge energy in a region using the same filters as for contour extraction.

3.4. Position Feature

We use a rough position feature based on height as a percentage of image height. As can be seen in the figures in supplemental material [3], the actual height in the image where streets or buildings or trees are found varies significantly, but generally the street is down and the sky is up.

3.5. Feature combinations and models

The basic features above are combined into five different feature sets and used to predict coarse labels as follows:

1. *Color Histograms* are used alone and modeled with k nearest neighbors on a quantized version of the training data. A set of around 200 (varies by experiment) representative histograms are found that cover the entire set of training data with balls of fixed radius with respect to the Mallows distance. The class distribution of training data within the same fixed distance of a representative histogram is computed on the training data. For a new histogram the estimated class distribution is just an average of the class distributions for its 10 nearest neighbors. This is a variant on edited nearest neighbors.

2. *Texturedness and position:* Texturedness and the percentage height in the image for a 2 dimensional feature and the class distribution can be modeled directly with a histogram.

3. *Central pixel Color and Texturedness:* For the foliage and sky categories we want to provide near pixel accuracy in order to potentially use the result for image matting. To this end, for these categories, we include features based on support vector regression on the category (foliage or sky) given the color of the central pixel concatenated with the amount of texturedness in the window surrounding the pixel to form a vector in R^4. The dimensions are scaled to have nearly equal range. By themselves these features are not sufficient, but they do help in localizing boundaries.

4. *Contour Features and Average Color:* The two contour feature histograms and the average color in Lab space are concatenated and compared using the Euclidean norm. The class distribution is again modeled with edited nearest neighbor density estimates.

5. *Position:* Since position is measured by percentage height in the image the class distribution can be modeled empirically by a count for each percentile of height.

4. Parsing

We next describe the stages of parsing for a novel image and provide a more in depth discussion of the detailed parsing procedure.

The stages of processing for a novel image are as follows:

1. Compute image features – The basic features presented in Section 3 are computed.

2. Per pixel coarse level parsing/recognition using only unary potentials. The features are combined into feature sets and compared to the training data using the models described in Section 3 to produce a distribution over coarse classification classes.

3. Estimate a per image color model for sky and building – Using pixels with high confidence for sky or building labels, a color model for the image is fit using k-means. This allows the algorithm to "realize" that the sky is blue in a particular image.

4. The per pixel coarse level parsing is repeated now averaging together the generic model for coarse label given color with the color model learned in the previous step. (The combination parameter was learned by cross validation.)

5. Spatial smoothing of the per pixel label probabilities. The potential for label co-occurrence in a square around each pixel is learned from labeled training data and iterated five times to smooth label probabilities.

6. Detecting detailed structures.

7. (Optionally iterate steps 3-6.)

8. Spatial smoothing of detailed structure probabilities.

4.1. Detailed Parsing

The detailed labels l^d_j are somewhat different than the per pixel coarse labels. There are potentially a huge set of detailed labels based on the various parameters for each label – each type of feature, door, window, roof, roofline, building boundary, power-line, and bare-tree – and for each, location, scale, and if applicable aspect ratio and color model. In order to manage this potentially huge number of labels we use labeled data to train detectors for the various detailed features, and only instantiate label variables l^d_j when the detector is above a threshold set to keep the number of labels on the order of tens of thousands and below millions.

The detectors for the linear structures such as rooflines, bare trees, and building boundaries are based on the connected contours detected by the contour features computation and the coarse labels immediately surrounding the contour. The detectors for the window and door features are based on detecting nearly right angles and t-junctions (of nearly right angles) in the contour map and the underlying coarse labels.

Once labels are instantiated they are assigned a probability using a data driven model over the parameters mentioned above for each type of detailed feature and the immediately surrounding coarse pixel labels (the neighborhood for that label). This model is again an edited k nearest neighbor density estimate.

For the window labels an additional round of data driven spatial smoothing is applied with the neighborhood for each window label consisting of any window labels overlapping in row or column, and any roofline labels. An example of this is shown in the supplemental material [3].

For coarse pixel labels, the neighborhood is a square twice the side-length of the square used to compute image features relevant to that pixel and centered around the pixel. For detail labels the neighborhood depends on the size and type of structure. As an example, for windows the neighborhood contains all coarse labels under the extent of the window as well as in a buffer zone 30% again as large around that region. Also any other window labels with overlapping vertical or horizontal extent in the image are included, as is any roofline label above the window's extent.

5. Experiments

We take three approaches in evaluating the effectiveness of our parsing system: 5.2 measures the information gain due to some of our individual features and their combination against the output of another system, 5.3 visualizes the detailed parsing produced in the final stage of our system, 5.4 demonstrates how this detailed information makes it possible to search or browse a collection of roadside images by meaningful features such as building color (as opposed to image color) facade layout, and window design.

5.1. Scene Collection

Our images come from a variety of sources (see acknowledgments) including our own collection and photos from [2] and [1] and are taken around the world. Training images were labeled by hand for sky, buildings, trees, street, windows, and roof lines. These images are more limited in variation than the image data set as a whole.

5.2. Information Gain Measurements

We measure the relative information gain, $R(X;Y)$, between a feature X and a label Y, defined as:

$$\begin{aligned} R(X;Y) &= \frac{I(X;Y)}{H(Y)} \quad (2) \\ &= \frac{H(X) + H(Y) - H(X,Y)}{H(Y)} \quad (3) \end{aligned}$$

Here $I(X;Y)$ is the mutual information between X and Y and $H(X)$ is the entropy of X. The relative information gain tells how much a feature X reduces the entropy in a guess of Y as a ratio to the amount of entropy in Y by itself. Values for R fall between 0 and 1, with small values indicating that little information is added by the feature and large values the opposite.

Table 1 shows the relative information gain provided by the color, texture and position, and position features on our dataset. For each feature we measure how much it tells about each label. As a comparison the relative information gain provided by the output of Hoiem *et al.*'s code on our data is shown in the same table.

For our training and test we use cross validation with four splits of 30 training and 11 testing images. Variance was less than 0.02 in all cases.

Table 1. Relative Information Gain (Our dataset)

	Sky	Foliage	Building	Street
Hoiem *et al.*				
por	0.20	0.50	0.16	0.05
sol	0.43	0.10	0.13	0.06
h90	0.54	0.24	0.50	0.12
000	0.30	0.12	0.16	0.34
sky	0.85	0.15	0.20	0.18
This Work				
texture & pos	0.65	0.15	0.21	0.35
color hist	0.77	0.52	0.27	0.20
pos	0.41	0.02	0.15	0.33
combination	0.88	0.56	0.37	0.47

Table 2. Relative Information Gain: Hoiem *et al.* Training & Test

	sky	por	h090	000
texture & pos	0.62	0.19	0.06	0.45
color hist	0.67	0.13	0.13	0.15
pos	0.36	0.04	0.05	0.40

We note first that the Hoiem *et al.* result is quite impressive. Despite being designed for a slightly different task and trained on a separate dataset, their sky probability map (sky) provides a great deal of information about the presence of sky. Also their probability map for porous provides a good information about trees. Side by side comparisons are available at `www.cs.berkeley.edu/~aberg/iccv07`.

As might be expected the individual features shown provide some information about visual categories, but their combination (generic) is better than any individual feature, already providing significantly more information about the location of street, buildings, and sky.

The image specific model provides a small boost in mutual information, but the difference is more noticeable in the parses themselves. Figure 6 gives a qualitative idea of our results. Also smoothing does not greatly increase the mutual information, but it can provide better looking parses, especially by removing unsightly holes from buildings.

Table 2 shows the result of training and testing some of our features on the Hoiem *et al.* geometric context dataset of 300 labeled images split into 5 training and test combinations for validation. Here training, testing, and the vocabulary of labels are different than Table 1, but some measure of the relative difficulty of the datasets is shown. Position alone tells more about the Hoiem *et al.* ground label (000), than about street in out dataset. Foliage seems much easier to identify using color in our dataset, then the more general category porous in theirs.

5.3. Example Labellings

Next we present parses of images into visual categories and detailed structures.

Figure 6 illustrates the processing pipeline for our algorithm. Starting with the initial image in the left column, the second column shows the parse using generic category models. This represents the combination of all the features. The color indicates the most likely visual category at each pixel. (The legend in Figure 1 applies here as well.) Note that sometimes sky or street is the most likely color in some building regions, this indicates a relatively soft position model. The third column shows the result of our image specific parsing where the building color and sky color in the particular image are estimated and used. This usually reduces the number of mistaken pixels in the building region. Local spatial smoothing of the labeling produces is in the fourth column. Building color is estimated, but not shown in this figure, see [3] for full parses. Finally the detailed parse is in the last column. One striking feature is the range of architectural scenes that can be parsed with this simple model.

5.4. Similarity and Search

We show examples of how this additional structure might be used to drive novel similarity based search or browsing through collections of roadside images. Because we can obtain a good estimate of building color it is possible to search for buildings of similar color as shown in Figure 4. Note that though the building color itself is sometimes a small portion of the image, the search automatically ignores other parts of the image that would confuse a color similarity search based on the whole image. Also similarity in the layout, aspect ratio, and number of windows can provide clues to building similarity as shown in Figure 2. Finally by extracting windows from all of the images we can browse roadside images by their windows and group similar windows Figure 3. All of these detailed features would be nearly impossible to extract without the contextual information provided by the initial parsing into visual categories.

6. Conclusion

We have presented a system for parsing architectural scenes – first segmenting them into the basic visual cate-

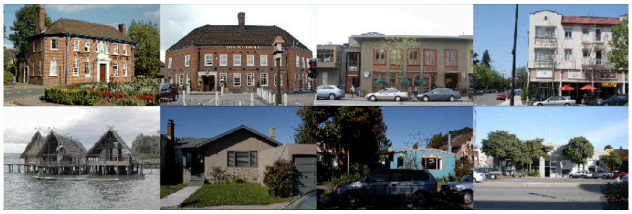

Figure 2. *Window layout similarity based search (# stories and # windows/story): Using the image at upper left as a query, the rest of the database is ranked by similarity of window layout. The remainder of the top row shows the most similar matches. The bottom row shows the least similar buildings.*

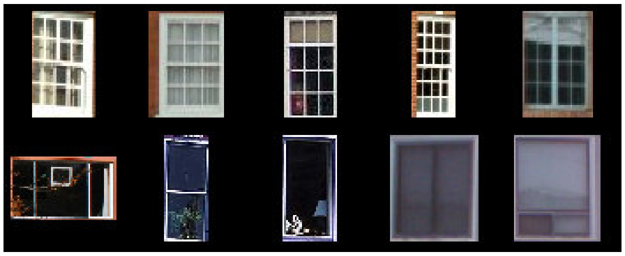

Figure 3. *Window appearance similarity search: Using the image at upper left as a query, the rest of the windows in the database are ranked by appearance similarity. The remainder of the top row shows the most similar matches. The bottom row shows the least similar. The actual database contains 709 windows.*

gories sky, foliage, building and street, and further parsing out detailed structure including rooflines, walls, windows, doors etc. The system is driven by fixed-size patch-based features, but produces results comparable or better than the state of the art for parsing into basic categories and identifying detailed structures. A key idea in our system is first applying a generic appearance model, and using this to fit an image specific appearance model. We demonstrate how our parsing results can be used to drive similarity search based on semantically relevant structures on buildings.

7. Acknowledgments

We would like to thank the reviewers for especially helpful comments, Maneesh Agrawala for pointing us toward streetside images, Jason Sanders for photos around Berkeley, and Tamara Berg for labeling. This work was partially supported by ONR Grant N00014-06-1-0734.

References

[1] http://arglist.com.

[2] http://bigphoto.com.

[3] http://www.cs.berkeley.edu/∼aberg/iccv07.

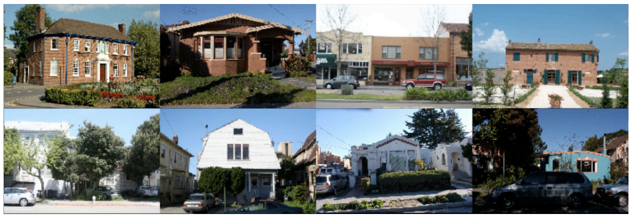

Figure 4. *Building color similarity based search: Using the image at upper left as a query, the rest of the database is ranked by similarity of building color. The remainder of the top row shows the most similar matches. The bottom row shows the least similarly colored buildings. Note the difference between building color and overall image color that can be exploited given a reasonable estimate of the spatial support of the building in the image.*

[4] J. M. Coughlan and A. L. Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NIPS*, pages 845–851, 2000.

[5] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. In *IJCV 40*, 2000.

[6] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics*, 30, 1996.

[7] F. Han and S. Zhu. Bayesian reconstruction of 3d shapes and scenes from a single image. In *Workshop on Higher Level Knowl. in 3D, Nice*, 2003.

[8] F. Han and S. Zhu. Bottom-up/top-down image parsing by attribute graph grammar. In *ICCV*, 2005.

[9] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. In *SIGGRAPH*, 2005.

[10] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005.

[11] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. In *cvpr*, 2006.

[12] Z. Kim and R. Nevatia. Automatic description of complex buildings from multuiple images. *Computer Vision and Image Understanding*, 96, no. 1, 2004.

[13] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV*, 2003.

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[15] S. C. Lee and R. Nevatia. Extraction and integration of window in a 3d building model from ground view images. In *Computer Vision and Pattern Recognition*, 2004.

[16] E. Levina and B. P. The earth mover's distance is the mallows distance: some insights from statistics. In *NIPS*, November 2001.

[17] A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, November 2005.

[18] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: unifying segmentation, detection, and recognition. In *IJCV 63*, 2005.

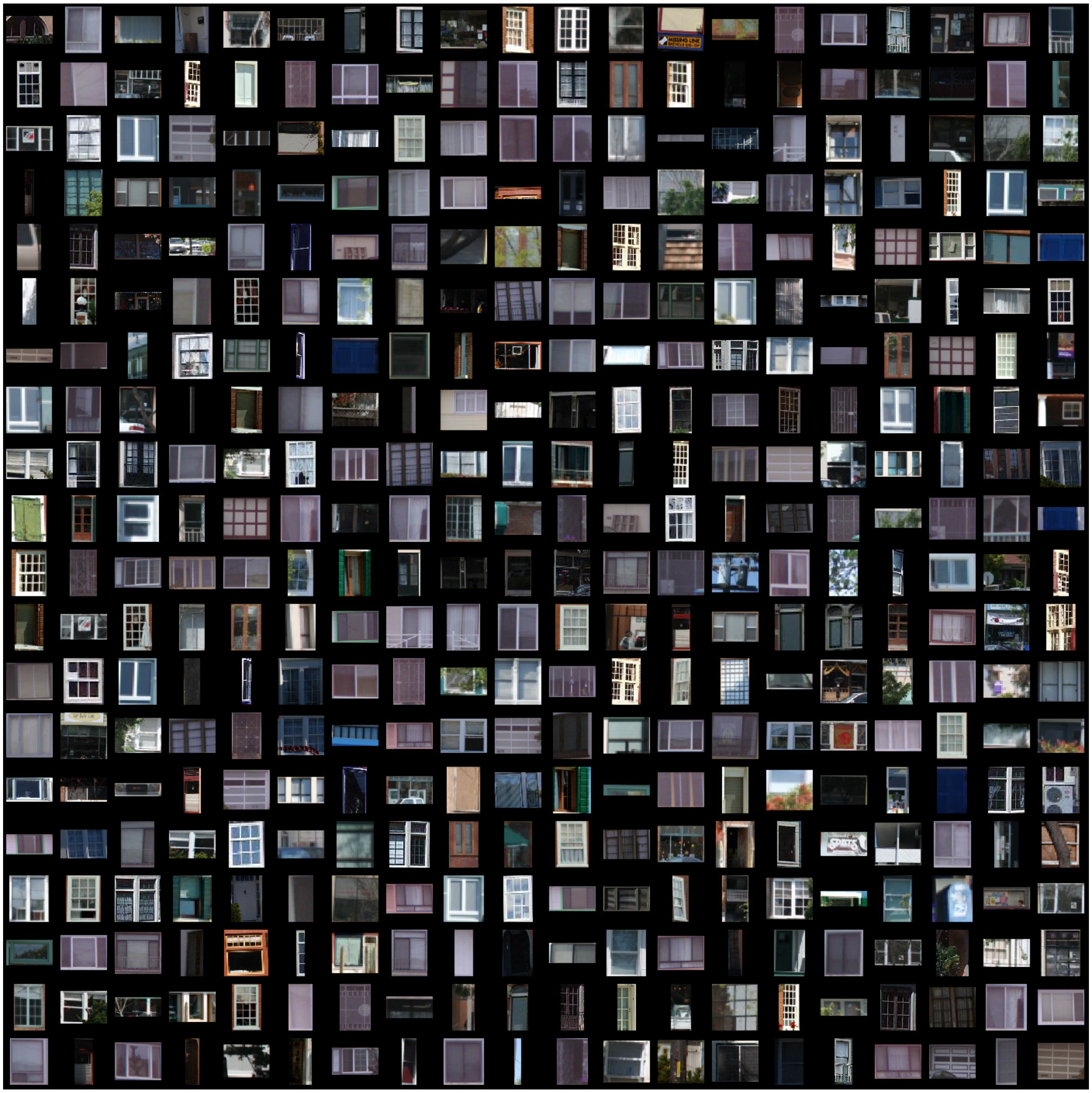

Figure 5. *A sampling of automatic window detections from our dataset, illustrating the wide variety of windows present. Some window-like patterns are detected as windows. Can you spot the garage doors or car windows?*

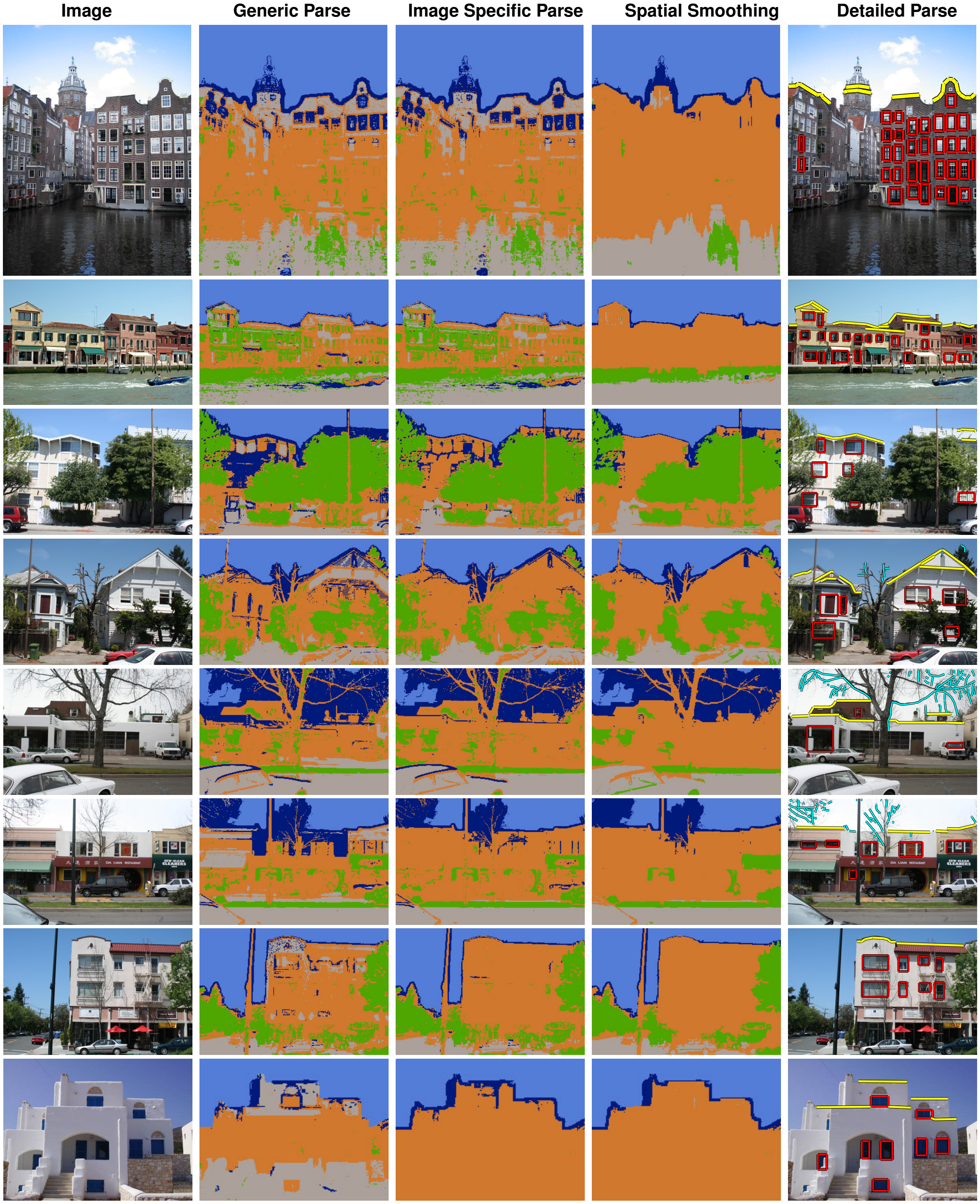

Figure 6. *Examples of images passing through the processing pipeline, see Figure 1 and Section 5.3. The legend for parsing can be found in Figure 1. Many more examples are available [3].*