Window Detection in Frontal Facades

Viraj Kulkarni
Department of Computer Science
University of California, Berkeley

Rohan Nagesh
Department of Computer Science
University of California, Berkeley

Hong Wu
Department of Computer Science
University of California, Berkeley

ABSTRACT
In this paper, we describe a novel technique to identify and extract windows from a building's frontal facade. Feature detection in buildings has long been an area of research interest for its applications in 3D city modeling and scene visualization. We utilize a combination of projection profiles, mutual information, and a feature extraction technique (Snake Algorithm). We have found this approach to work quite well on regular facade structures, and we conclude with a more detailed discussion of our results.

Keywords
Window detection, feature extraction, building facades, urban reconstruction, Snake Algorithm

1. INTRODUCTION

This paper addresses the research question of computationally extracting windows from rectified, frontal facade structures. With many applications ranging from 3D geometric modeling of facades to urban landscape reconstruction, there has been considerable interest and innovation in this space.

There exist numerous computer vision challenges with the task of extracting windows from such facade structures. First, the method of rectifying input images, which are usually satellite or aerial shots, into ground-view images is pivotal. Second, the facade may contain non-interesting artifacts—occlusion by vegetation and transparency of glass windows. Shadows and lighting issues can significantly impact extraction quality as well. Lastly, due to the vast array of building structures, classification algorithms must be versatile enough to handle a variety of geometries.

Broadly speaking, there have been two main approaches to addressing this research question. First, there are machine learning approaches that operate on a training set of images to generate feature weights for future analysis. Second, there are approaches that focus on a particular input image for the duration of the algorithm, exploiting geometrical properties of the image for extraction. We will focus on the latter approach in this paper.

Additionally, we restrict our domain to purely frontal facades. What we mean by this is that aerial or satellite images have been rectified to provide straight, ground-view shots of the facade in interest (as in Figure 1). In particular, we do not analyze angular shot.

2. RELATED WORK

As discussed earlier, there has been much related work in the field, spanning both machine learning and single-image analysis approaches. Cech and Sara discuss *Windowpane Detection based on Maximum Aposteriori Labelling* (2007)[1], a segmentation technique that offers a stronger structure model than traditional Markov Random Fields. Ali et al[2] devises a machine learning approach that utilizes Haar feature model in conjunction with a Gentle Adaboost-driven cascaded decision tree. While both techniques are quite intriguing, we focus on techniques less heavy on machine learning due to significantly faster completion times and similar accuracy.

In the array of non-machine learning approaches, we focus on two papers in particular. Lee and Nevatia[3] employ a projection profile geometry technique to quickly obtain a fairly accurate grid consisting of a facade's windows. Muller et al.[4] discuss a similarity detection technique known as mutual information to obtain their grid segmentation.

While Lee and Nevatia's approach is quick to implement and quite fast to complete, we observed that the resulting window grid segmentation was not as polished as that of Muller's mutual information approach. High levels of gradient in both the vertical and horizontal directions can exist outside the areas containing windows. Muller's mutual information algorithm conducts an exhaustive search on the facade while comparing adjacent regions of the image for similarity; their approach is significantly slower than that of Lee and Nevatia.

While we draw heavily from both of these papers, we believe our approach produces similar results with significantly faster completion times.

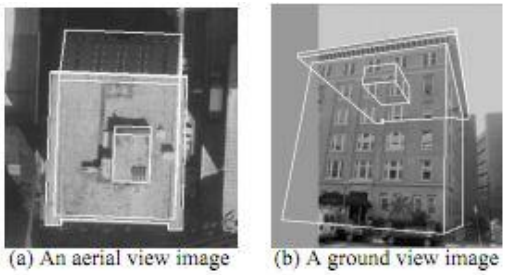

Figure 1: Process of extracting ground view image from aerial view image

3. METHOD

The input images can be obtained from ground-based imagery. Rectified facade textures can be easily extracted from photogrammetric urban models. However, there are also various public tools for rectification for cases where such rectification is still needed.

Our proposed solution consists of using a three staged pipeline to segment out windows from a single frontal image of a facade. Figure 2 shows the pipeline We work by subdividing the facade into a grid of horizontal and vertical lines. Ideally, each rectangle in this grid should have one window. We refer to this rectangle as a tile. The first two stages of our pipeline work towards obtaining this grid. Once we get the grid, we use an object segmentation algorithm to segment out the window inside the tile.

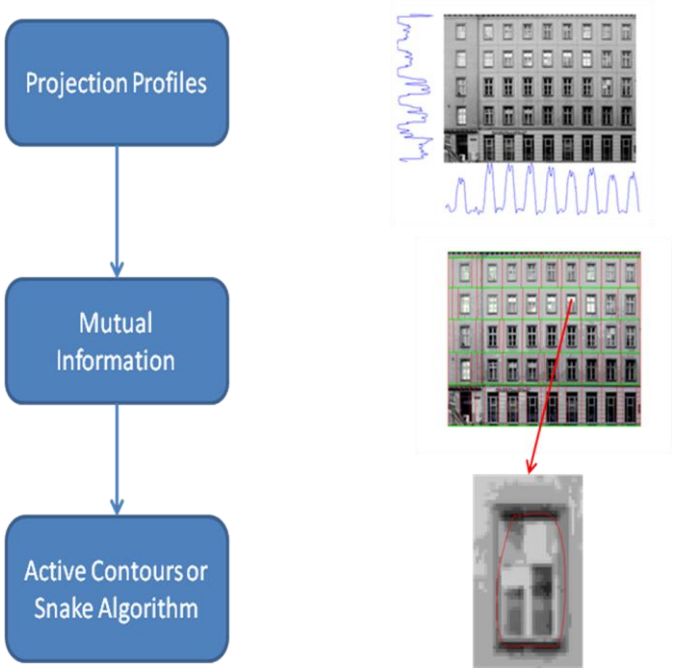

Figure 2: The three stages of our algorithm along with thumbnails showing the outputs at each stage.

In the following subsections, we give a description of each individual stage of our algorithm.

3.1 Projection profile based approach to detect approximate height

A key component of our algorithm is the determination of the dimensions of the grid. In the first stage, we use a projection profile based approach to extract 2D rectangles by exploiting the geometric property of 2D rectangles and alignment of the building windows. This is similar to the method proposed by Lee and Nevatia.[3] The goal of this stage is to get an approximate height of each floor and the width of each window tile in the façade. We use this information in the second stage to formulate a grid which separates out all window tiles.

We project horizontal and vertical image edges to give a total of two projection profiles: a horizontal projection profile of the horizontal edges and a vertical projection profile of the vertical edges as shown in Figure 3. Each projection profile is obtained by summing up the gradients in every row and column.

Because the building windows are horizontally or vertically aligned, the image edges within the windows of the same column or row are accumulated at the same location of the projection profile histograms.

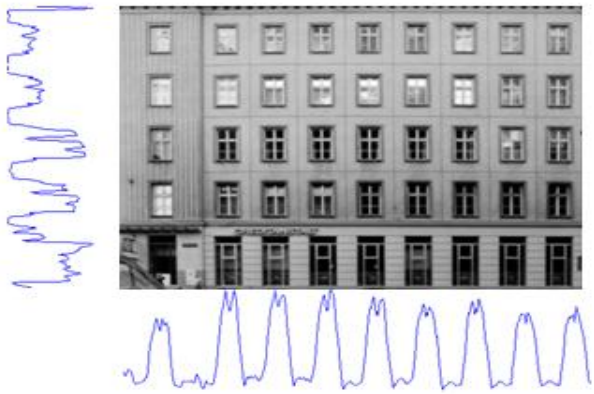

Figure 3: Horizontal and vertical projections of the input image

We select the valleys of these profiles as indicators for approximating the height and width of the window tiles. We select a percentile threshold value of 25 and use it to cut the profile. Basically, we cut the profiles by a straight line such that 25% of the values in the profile would like below this line. The profile wave intersects this line twice for every cycle – once when rising and once when falling. The average of these two points gives us an approximate height (for vertical profiles) and width (for horizontal profiles) of the individual tiles.

Due to noise, these values are only approximations and we do not use them directly for plotting the grid. Instead, we use them to derive a plausible range of values which we use to compute the actual heights and widths in the next stage.

3.2 Determination of Façade Structure

The goal of this stage is to detect the structure in the façade and to subdivide it into floors and tiles as shown in Figure 4. We detect similar regions in the image using mutual information as a measure of similarity as proposed by Muller et al.[4]

Figure 4: Input image split into a grid after the second stage

3.2.1 Mutual Information

In probability theory and information theory, the Mutual Information (M.I.) of two random variables is a quantity that measures the mutual dependence of the two variables. It quantifies the Kullback-Leibler distance (Kullback)[6] between the joint distribution, P(A = a, B = b), and the product of their marginal distributions, P(A = a) and P(B = b), that is

$$MI(A,B) = \sum_{a,b} P(a,b) \log \frac{P(a,b)}{P(a) \cdot P(b)},$$

where A and B are two random variables. MI was proposed as a similarity measure on image intensities for 3D rigid registration in medical imaging by Wells et al.[7] It does not assume any simple or one-to-one relationship between the intensities. The MI-based similarity MI(I(R1), I(R2)) measures the statistical dependence between intensities at corresponding positions in regions R1 and R2. Accordingly, I(R1) and I(R2) are the intensities at corresponding image locations in R1 and R2. Next we describe how MI is used to find similar image regions.

3.2.2 Symmetry Detection

In this step, we use mutual information to find similar floors and tiles in the image. We expect similarity between various floors although the top and the bottom floors often differ. Each floor consists of repeating patterns in the form of windows and each tile is roughly similar to another tile of the same floor. Our algorithm searches first for symmetry in the vertical and then in the horizontal direction. We describe in this section our method for computing the height of the floors in the vertical direction. After this is done, the next step to find the width of each tile is very similar to this.

Let $R_{y,h}$ denote the rectangular image region with a lower left corner of (0, y) and upper right corner of (image width, y+h). For detecting similarity in the vertical direction, we need to analyze regions which can be denoted by $R_{y1,h}$ and $R_{y2,h.}$ for all possible values of y and h. Such an exhaustive search takes a very long time to complete.

We simplify this problem by analyzing only vertically adjacent regions $R_{y,h}$ and $R_{y-h,\,h}$ where h can take a range of values which is obtained from the previous stage.

The similarity between two adjacent regions with height h is computed by:

$$S(y,h) = MI(I(\mathcal{R}_{y,h}), I(\mathcal{R}_{y-h,h})).$$

We use an exhaustive search strategy to compute S(y,h) for all positions y, and a range of parameters for h. We use the approximate height we obtained in the first stage to derive the range of parameters for h.

The same strategy is applied in both vertical and horizontal directions. At the end of this second stage, we get a grid of lines that divide the façade into tiles such as shown in Figure 4.

3.3 Snake Algorithm

In this third and final stage, we segment out individual windows from the grid we obtain in stage two. We use the snake active contours model as proposed by Kass[5] for this purpose.

A snake is an energy minimizing, deformable spline influenced by constraint and image forces that pull it towards object contours. One may visualize the snake as a rubber band of arbitrary shape that is deforming with time trying to get as close as possible to the object contour.

The following formula represents the energy function which we try to minimize. It consists of three energy terms: snake or image energy, constraint energy and internal energy.

$$\begin{aligned} E^*_{\text{snake}} &= \int_0^1 E_{\text{snake}}(\mathbf{v}(s))\, ds \\ &= \int_0^1 E_{\text{int}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) \\ &\quad + E_{\text{con}}(\mathbf{v}(s))\, ds \end{aligned}$$

The snake and constraint energies are together referred to as the external energy. The internal energy is the part that depends on intrinsic properties of the snake, such as its length or curvature. The external energy depends on factors such as image structure, and particular constraints the user has imposed.

We run this algorithm on each tile to segment out the windows. The final output of this stage is a single window for each tile as shown in Figure 5. We operate on each tile independently and this turns out to be a time consuming and more error prone method. This can be optimized by using the notion of an irreducible façade as described by Muller

et al.[4] However, this remains as future work. We describe details of this in the future work section of this paper.

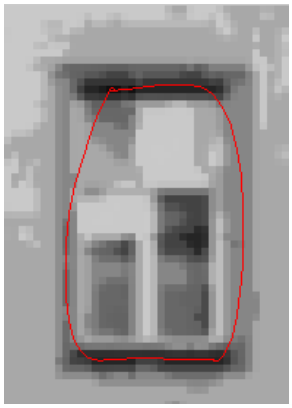

Figure 5: The active contours start from the boundary of the tile and segment the window from the tile

4. RESULTS

Our algorithm is implemented in MATLAB. The results are run on an Intel 2.67GHz Core i5 processor with 2GB RAM.

Figure 3 and 4 show the projection profile and mutual information results. Figure 6 shows the final window detection.

Figure 6: Output shows the individual windows marked out in red.

Figure 7 and 10 are the input images. Figure 8 and 11 are the grid of tiles. In figure 9 and 12, the boundaries of the windows are marked as red contours.

Figure 13 and 14 are the images which the algorithm cannot detect the tiles correctly. The tile detection is quite important to the whole process because Snake algorithm does not work well on bad tiles.

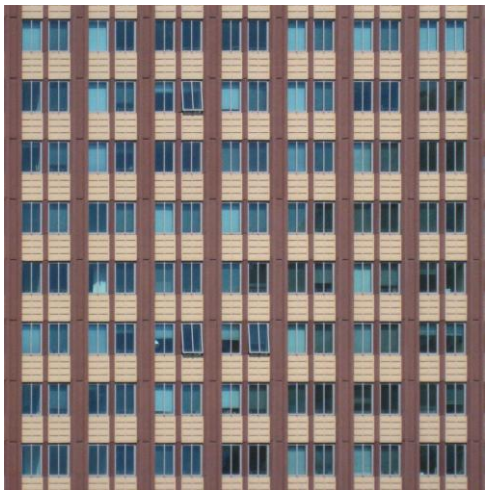

Figure 7: Input image of a facade

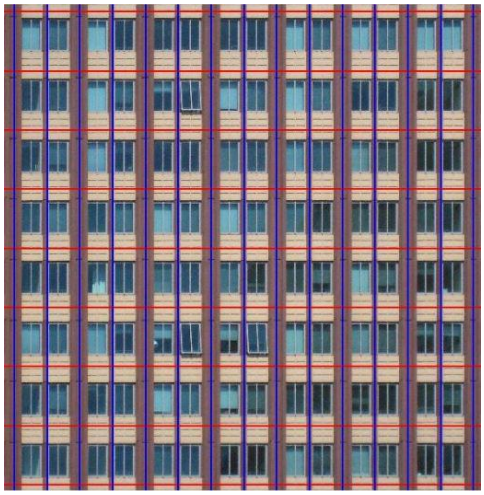

Figure 8: The division of the façade into a grid using mutual information to detect similarities

Figure 9: Output shows the individual windows marked out in red.

Figure 10: Input image of a façade

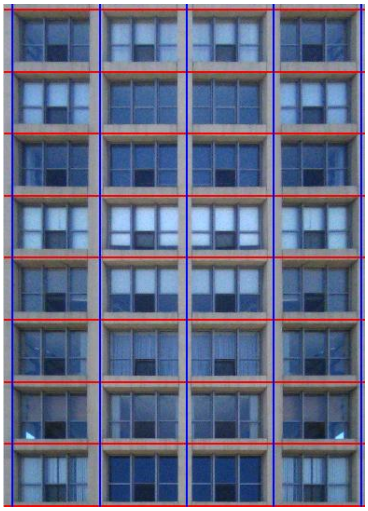

Figure 11: The division of the façade into a grid using mutual information to detect similarities

Figure 12: Output shows the individual windows marked out in red.

Figure 13: The input facade on which our algorithm failed.

Figure 14: The tile segmentation is poor because of the small margin between windows and the glare created

5. DISCUSSION

For each image group, we present the input image, the image after the first two steps of our workflow (projection profile followed by mutual information), and lastly the image after executing our Snake algorithm. To recap, we utilized projection profiles and mutual information to divide the facade into a grid and thereby identify tiles containing windows. From this, we execute our Snake algorithm to extract the windows from the tiles.

In terms of efficiency improvements, our algorithm proved to be significantly faster than Muller et al.'s pure Mutual Information approach. Because we first utilize the gradient projection method to obtain an approximation for the height and width of each floor, we do not need to perform an exhaustive search across all possible values of heights. Although Muller et al. does mention assuming a range of 3 meters to 5.5 meters for the height, we found even this approach to be limiting and slower. To provide some concrete data, Muller et al. states his workflow completes on the order of minutes. However, our first two steps (gradient projection and mutual information) complete in less than five seconds utilizing a Matlab implementation and our Snake algorithm completes in 30 seconds to 1 minute.

In evaluating our Snake algorithm, we found the workflow quite successful in extracting windows from tiles containing one window or now windows at all. However, the algorithm struggled to handle tiles that had not been properly segmented and contained two windows as a result.

As observed from our results, our algorithm performs quite well on regular patterns and geometries as in Figures 7 and 10 but lacks the versatility to handle harder images, those with significant occlusion or internal reflection as in Figure 13.

We suspect this behavior was the result of two main obstacles. First, in input images with no clear windowpanes or dividers between sets of windows, our algorithm struggled to produce a clear line in both directions. Figure 13 is a good example of an input image with unclear and faint demarcations between windows, which was a contributing factor to its high difficulty for the algorithm. Second, input images with significant glare or internal reflection created drastic complications for our algorithm. While some glare with clearly demarcated windows fared reasonably well (as in Figure 10), Figure 13 once again illustrates the difficulty, even for humans, to detect location and quantity of windows. Additionally, reflections of images across the facade in the space of the scene will further exacerbate the challenges. We believe our gradient projection step of the workflow itself failed to handle such images, which in turn produced poor results in each of the last two steps.

Overall, we are quite satisfied with our efficiency improvement with our algorithm but will look to address the aforementioned challenges in future iterations.

6. FUTURE WORK

Within the scope of our approach itself, there is ample room for improvement. In particular, there is one element from Muller et al.'s paper, an irreducible facade that would integrate quite nicely with our methodology.

To provide some context, there is often significant similarity in building facades. Entire columns or rows may exhibit similar geometries to the point where executing an algorithm such as mutual information on each cell would simply be redundant. The notion of an irreducible facade is to compress a frontal facade vertically and horizontally into an indivisible, smallest facade that preserves all the unique geometries in the larger input image. By implementing this notion of an irreducible facade, our algorithm need only operate on this irreducible facade and not have to execute duplicate work on the entire image.

From a technical standpoint, the irreducible facade is a data structure that stores a list of pixels instead of a single pixel at every (x,y) location. This list of pixels corresponds to a "stack" of original similar image fragments, which when unrolled can produce the original image. Once no more similarity can be detected, the irreducible facade is complete.

Figure : (a) Original input image (b) Irreducible Facade

Outside of our approach, there is considerable room for improvement in terms of the varieties of input images our algorithm can accurately segment. More specifically, given our limiting assumption of dealing solely with frontal facades, our approach lacks the versatility to handle angular shots of scenes, oblique shapes, and other irregular geometries in both the facade and windows.

For instance, arch windows can be modeled through parametric equations and a Hough Transform in 3 dimensions. Angular shots can be handled either directly or can be converted to rectified frontal facades similar to those we have used as inputs in this paper.

In summary, the approach we have presented in this paper shows tremendous promise in handling the difficult problem of extracting windows from building facades. There is tremendous scope for innovation, and we aim to

address many of the aforementioned limitations of our approach in future iterations.

ACKNOWLEDGMENTS

We thank Professor Maneesh Agrawala and Graduate Student Instructor Floraine Berthouzoz of UC Berkeley's graduate-level course in Image Manipulation and Computational Photography (CS 294-69) for their valuable input and guidance throughout our project.

REFERENCES

1. Cech et al. Windowpane Detection Based on Maximum Aposteriori Labelling. *IWCI.* (2008).
2. Ali et al. Window Detection in Facades. *ICIAP*. (2007)
3. Lee, S. and Nevatia, R. Extraction and Integration of Windows in 3D Building Models from Ground View Images. *CVPR*. (2004).
4. Muller et al. Image-based Procedural Modeling of Facades. *ACM SIGGRAPH*. (2007).
5. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," Int. J. Comput. Vis., vol. 1, pp. 321–331, 1987.
6. Kullback, S. Information Theory and Statistics. John Wiley and Sons., New York. (1959).
7. Wells et al. Multi-modal volume registration by maximization of mutual information. (1996).