Detection of Façade Regions in Street View Images from Split-and-Merge of Perspective Patches

Fei Liu [1] and Stefan Seipel [1,2]
[1] Department of Industrial Development, IT and Land Management, University of Gävle, Gävle, Sweden
[2] Centre for Image Analysis, Uppsala University, Uppsala, Sweden
Email: {feiliu, ssl}@hig.se

***Abstract*—Identification of building façades from digital images is one of the central problems in mobile augmented reality (MAR) applications in the built environment. Directly analyzing the whole image can increase the difficulty of façade identification due to the presence of image portions which are not façade. This paper presents an automatic approach to façade region detection given a single street view image as a pre-processing step to subsequent steps of façade identification. We devise a coarse façade region detection method based on the observation that façades are image regions with repetitive patterns containing a large amount of vertical and horizontal line segments. Firstly, scan lines are constructed from vanishing points and center points of image line segments. Hue profiles along these lines are then analyzed and used to decompose the image into rectilinear patches with similar repetitive patterns. Finally, patches are merged into larger coherent regions and the main building façade region is chosen based on the occurrence of horizontal and vertical line segments within each of the merged regions. A validation of our method showed that on average façade regions are detected in conformity with manually segmented images as ground truth.**

***Index Terms*—façade region detection, street view image, vanishing point, mobile augmented reality**

I. Introduction

Recent advances in mobile computing have led to an increased demand for visual landmark identification in mobile augmented reality (MAR). MAR systems rely generally on two separate visual processing components [1]: tracking of the real-time video feed is needed to coherently co-register graphical augmentation with the imagery. Recognition is used to retrieve information relevant to the objects identified in the video. The rapid development of capable smart phones has recently fostered the development of a variety of methods to track features in video images. Some of those systems rely on motion vector tracking [2] and [3], while others, such as the well-known SIFT [4], SURF [5], or CHoG [6] methods, retrieve local feature points based on histograms of gradients. Current feature point tracking algorithms provide stable and coherent tracking and since they are based on recognition and matching of local features, they are unaware of the actual scene content. In an MAR system to be used in the urban environment the recognition component of the system requires identification of the most prominent building or façade in the current view of the camera (see Fig. 1). Building recognition is a challenging task for many reasons. As Chen *et al.* state [7], query images are usually taken under very different conditions from the database images. Buildings also tend to have few discriminative visual features and many repetitive structures and their 3D geometries are not easily captured by simple affine or projective transformations. We also observe that façades are often partly occluded by trees, vehicles or other objects. In this work we contribute with an image pre-processing pipeline for automatic detection and coarse segmentation of building façades in street-view images. The main objective is to delineate the region of the main façade of interest in query images which in a longer perspective will serve to restrict the building recognition task of an MAR system to such pre-selected façade regions in the image.

Figure 1. Concept demonstration of the final AR system in which the proposed method will be used

A. Related Work

Although not aiming at façade detection, the analysis from Korah and Rasmussen [8] about building textures provides an insight into the regularly structured nature of façades. Wendel *et al.* [9] pointed out that a single façade segment was a coherent area in an image, containing repetitive patterns which match in color and texture. The approach they proposed can separate one façade from

Manuscript received February 11, 2014; revised May 9, 2014.

doi: 10.12720/joig.2.1.8-14

another but cannot separate a façade from non-building objects in the scene, which is the goal of this work. To our best knowledge, building detection from the street view has not yet received much attention. David [10] identified façades as planar structures in urban outdoor environments. Intersections of edge lines categorized by scene vanishing points were used as supports for potential planes. Li and Shapiro [11] derived features from outdoor scene images by consecutively clustering edge line segments according to major colors on both sides of each line segment, orientations and locations of the line segments. A simple decision-tree classifier was used for detecting buildings in those images. Similar to the work in [10], Trinh and Jo [12] grouped parallel line segments according to vertical and horizontal vanishing points to identify façades as meshes of basic parallelograms. Delmerico *et al.* tackled the façade detection problem via the assistance of stereo information [13]. They extended Boosting on Multilevel Aggregates (BMA) with the disparity feature to compute a probability map for an image in terms of whether a pixel belongs to a building. A double-layer Markov Random Field (MRF) was constructed using the probability map, estimated plane models and the disparity values to infer whether a pixel is a façade pixel and which façade it belongs to if it is.

Due to human preference, buildings tend to have a lot of horizontal and vertical linear elements on their façades. Under perspective distortion, the extended lines of these elements from a façade converge on two vanishing points, one for horizontal lines and one for vertical. Identifying these vanishing points provides a strong cue for the 3D properties of the façade in the 2D image. Therefore, similar to [10] and [12], our method also starts with scene vanishing points. However, instead of directly using lines incident with the vanishing points to form supports for façade planes, we incorporate the repetitive nature of façade appearance. After detecting the vanishing points, we scan the hue channel of the image along those lines incident with the vanishing points. Given a region with a repetitive pattern, scanning through it along lines with the same orientation should return similar hue profiles. Therefore, the image can be divided into several distinct regions delimited by some scan lines horizontally and vertically. We call these regions homogeneous regions henceforth. The image can then be subdivided into several regions with coherent contents by intersecting these two groups of homogeneous regions. Some non-building objects such as sky, lawns and pavements, due to their uniform nature in appearance, will also yield similar hue profiles when scanned through, thus forming homogeneous regions. In the final step we introduce a criterion to select a coherent image region that is most likely to be the façade. The paper is organized as follows: Section II to V describe each step of the method in order. The experiment for validating the method is covered in Section VI. We discuss our method and draw a conclusion in Section VII.

II. Scanline Generation

In a pre-processing step we blur the original RGB image repeatedly with a Gaussian filter (Fig. 2a) in order to eliminate most disturbing edges that result from, e.g., tree branches or reflections on windows. Also, after conversion of the blurred image into HSV color space later for scanning, the hue channel has more coherent regions, the benefit of which will be elaborated in Section III. Experiments showed that blurring an image around 20 times yielded superior final results in general so we choose to blur images 20 times in this work. A Canny edge detector is then applied to the blurred RGB image and detected edges are fed into edge linking and line segment fitting functions provided by [14]. The end result is a set of line segments derived from edges (Fig. 2b).

A. Two Major Vanishing Points

Repetitive patterns on a façade mainly occur horizontally or vertically. Hence we focus on detecting vanishing points in these two major directions. The detection process consists of two steps, initialization and refinement, which is similar to [15] and [16].

The initialization employs RANdom SAmple Consensus (RANSAC) [17] to produce initial positions of vanishing points. The first two runs of RANSAC will return the two major vanishing points and their respective incident lines.

We use the result from RANSAC as the starting point for the Expectation-Maximization (EM) algorithm to refine the positions of vanishing points. The E step computes the membership of line l_i in terms of vanishing point v_k, which can be expressed as the probability of v_k, given l_i, $P(v_k|l_i)$. Using Bayes' theorem, we have

$$P(v_k|l_i) = \frac{P(l_i|v_k)P(v_k)}{P(l_i)} \quad (1)$$

Assuming a normal distribution for the distance between v_k and l_i, the conditional probability $P(l_i|v_k)$ can be modeled accordingly,

$$P(l_i|v_k) = \exp\left(\frac{-0.5*d^2(v_k,l_i)}{\sigma_k^2}\right) \quad (2)$$

where $d(*)$ denotes distance. The term $P(v_k)$ is initialized according to an equal prior probability and the total probability $P(l_i)$ can be given by

$$P(l_i) = \sum_k P(l_i|v_k) * P(v_k) \quad (3)$$

The M step uses the membership to adjust the position of each vanishing point according to

$$v_k^{next} = \text{argmin}_{\bar{v}_k} \sum_i P(v_k|l_i) * d^2(\bar{v}_k, l_i) \quad (4)$$

where $\overline{v}_k$ starts with v_k. We also update the prior probability in this step using

$$P(v_k) = \frac{1}{N}\sum_i P(v_k|l_i) \quad (5)$$

where N is the total number of inlier lines returned by RANSAC. Fig. 2c and Fig. 2d show the line segments from Fig. 2b whose extended lines are incident with two vanishing points found in this section.

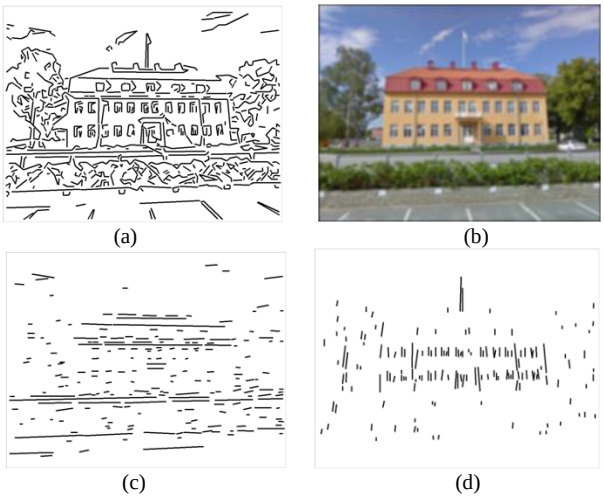

Figure 2. (a) Blurred color image; (b) Detected line segments from edges; (c) Horizontal line segments; (d) Vertical line segments

B. Scan Lines

After establishing the vanishing points, we construct two sets of scan lines across the entire image. For each horizontal or vertical line segment in Fig. 2c and Fig. 2d, we form a line using its center point and the corresponding vanishing point. Not all line segments are used for this purpose. Firstly, because some line segments are collinear. In this case, we choose the longest one among them since it is more reliable. Secondly, some regions end up with very dense scan lines. Since the scanning results are very similar from these lines, we reduce the density by imposing such a constraint as neighboring scan lines should have at least 10-pixel interval between their starting points. Fig. 3 shows constructed scan lines overlaid on the hue channel of Fig. 2a.

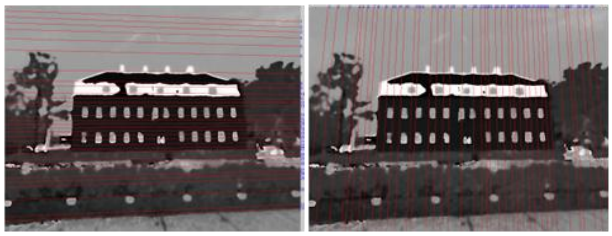

Figure 3. Horizontal and vertical scan lines

III. Homogeneous Region Detection

The newly constructed scan lines form the basis for the subsequent detection of homogenous regions. We scan the hue channel along those lines because hue values are often consistent on objects of the same type but vary between different types. This characteristic makes for more reliable scanning results. As introduced in Section I, homogeneous regions are delimited by scan lines which produce dissimilar profiles. We convert these profiles (1D signals) into the frequency domain by Fast Fourier Transform (FFT) and use the first 20 frequency components as a descriptor to characterize profile shapes. All profiles are re-sampled to have the same length to ensure the same fundamental frequency. The similarity is defined element-wise as

$$similarity_{p1,p2} = \sum_i [(real(D_{p1}^i) - real(D_{p2}^i))^2 + (imag(D_{p1}^i) - imag(D_{p2}^i))^2] \quad (6)$$

where $p1$ and $p2$ are the two profiles in question and $real(*)$ and $imag(*)$ represent the real and the imaginary part of a complex number while D represents a descriptor and i is the i^{th} element of the descriptor. We then normalize the similarity value to the range between 0 and 1 and set the threshold for similar profiles to 0.3 after experimenting with different values.

With the similarity measure defined, the next step is applying comparison results to the detection process of homogeneous regions. Here horizontal scan lines are used as example to convey the main idea and the vertical case can be carried out similarly. We start with the first scan line l_1 on the top of a hue image and compare its profile with the ones of subsequent scan lines. If a scan line l_i is encountered whose profile is different from the one of l_1, we have found a possible homogeneous region delimited by l_1 and l_{i-1}. However, since a single scan line cannot delimit a region, we label a new homogeneous region only if l_{i-1} and l_1 are not the same line (this could happen when l_1 and l_i are neighbors). This process is then repeated starting with l_i until the last scan line at the bottom of the image is reached. The results for both directions are shown in Fig. 4 (left and middle) respectively. These two groups of regions are then intersected with one another, which leads to a subdivision of the image into rectilinear patches shown in Fig. 4 (right). As can be seen in Fig. 4 (right), coherent regions in the image are not well represented by one single rectilinear patch. Instead, they are fragmented. For instance, the sky, the façade and the flower bed all consist of multiple patches. Consequently, we introduce a fragment merging process, which will be discussed in Section IV.

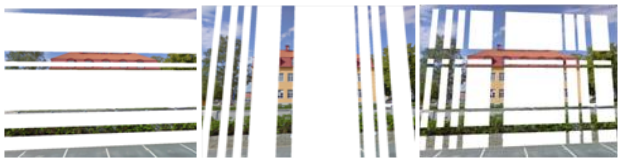

Figure 4. The white regions are (left) horizontal homogeneous regions; (middle) vertical homogeneous regions; (right) intersections of the two

IV. Merging Fragments

In order to remedy the fragmented result, we need to detect patches of the same coherent region and merge them. Any two patches detected in Section III are considered to be fragments if they are from the same scene object. This definition entails two criteria that dictate the merging process: first, the image contents of two patches must be similar; second, they need to be neighboring. In this section we present features to determine if two patches have similar contents. Subsequently, we describe the merging process itself taking the second criterion into consideration.

A. Patch Description and Similarity Measure

Due to the assumption that façades usually exhibit repetitive patterns, we use texture properties as descriptive features. Ohanian and Dubes [18] reported that co-occurrence features performed very well with small image patches. Such features are derived from a co-occurrence matrix, which comprises the occurrence frequencies of different pixel pairs in a gray-level image given a direction and a distance measured in pixels. In this work, we choose the distance to be 1 pixel (namely, neighboring pixels) and the directions to be east, northeast, north and northwest. Hence, there are 4 co-occurrence matrices. We then derive four features from each matrix: contrast, correlation, energy and homogeneity. For each rectilinear patch found in Section III, we thus construct a 16-dimension feature vector as the descriptor.

The straightforward Euclidean distance between feature vectors is employed to measure patch similarity. The challenge is, however, to set a threshold for the comparison, since a constant threshold for all images is assumed to perform poorly. Through studying a set of testing images, we observed that the center part of an image most likely contained complete scene objects, including façades. For example, in the middle portion of each row in Fig. 4 (right) we find sky, a façade and a flower bed from top to bottom. Similarly, in each column, from left to right we have trees, a façade and trees. Hence, we can assume if there are multiple patches in the center part of a row or a column, they are very likely to be fragments. Based on this assumption, we designed an algorithm to adaptively compute a threshold for each row and column of the patches. The detailed steps of computing a row threshold is listed in Algorithm 1. It starts with two distance values between three central patches in the row (see line 10 and 11). The branching logic between line 12 and line 23 is introduced to determine which distance value should be used to derive the threshold. The basic criterion is a range of possible threshold values bound by α_{lower} and α_{upper}. These bounding values were derived from observed distance values between pairs of fragments as well as non-fragment patches in a set of testing images. The threshold t is acquired by increasing the candidate distance value (d_1 or d_2) by a little amount δ so that the center patches can be merged due to our aforementioned assumption. After running this algorithm for all the rows, we replace those zero thresholds with the mean of non-zero thresholds. If all thresholds are zero, a default value m is assigned. At last, we relax the thresholds by adding a little value ε ($\varepsilon < \delta$) to them. The column thresholds can be found similarly. In that case, line 6 and line 7 become the patches above and below $patch1$. Table I lists values for various parameters used in this work.

TABLE I. PARAMETER VALUES FOR COMPUTING THRESHOLDS

δ	α_{lower}	α_{upper}	m	ε
0.1	0.3	0.75	0.5	0.05

input: a list of rectilinear patches p from one row of the grid with c columns
output: a threshold t for this row

```
1   if c is even then //determine 3 central patches
2     patch1 ← p[c/2];
3   else
4     patch1 ← p[(c + 1)/2];
5   endif
6   patch2 ← the left patch of patch1;
7   patch3 ← the right patch of patch1;
8   if patch1, patch2, patch3 all exist then
9     //3 patches are needed for two distance values
10    d1 ← distance between patch1 and patch2;
11    d2 ← distance between patch1 and patch3;
12    if d1 < αupper and d2 < αupper
13      t ← min(d1,d2) + δ;
14    else if d1 < αupper and d2 ≥ αupper
15      t ← d1 + δ;
16    else if d1 ≥ αupper and d2 < αupper
17      t ← d2 + δ;
18    else if d1 ≥ αupper and d2 ≥ αupper
19      t ← 0;
20    endif
21    if t ≠ 0 and t < αlower
22      t ← αlower;
23    endif
24  else // cannot find all 3 central patches
25    t ← 0;
26  endif
```

Algorithm 1. Compute a row threshold

B. Merging Process

As discussed earlier in this section, one of the criteria of two patches being fragments is that they need to be neighbors. Therefore, the merging process will always operate on adjacent patches. Meanwhile, in view of the grid layout of patches (e.g., Fig. 4 right), we split the merging process into two passes. The first pass merges neighboring patches within each row while the second pass continues with the results from the first pass and merges patches between neighboring rows. Within each row, starting from the left, let us assume there are n patches and they are denoted $p_1, p_2, \ldots, p_n$. We take p_1 and test if its content is similar to the one of its neighbor p_2 to the right. If they have similar contents, we merge p_1, p_2 into a new patch p_{12} and compute the feature vector of p_{12} and then compare p_{12} with its neighbor p_3 to the right. On the other hand, if p_1 and p_2 have different contents, we move on to p_2 and compare it with p_3. This process is repeated until p_n is processed. Fig. 5 (left) displays the result of this pass.

In the second pass, beginning with the result of the first pass, we try to merge between rows. Given a patch in a row, denoted row_i, we compare it with patches in the immediate row below row_i, namely, row_{i+1}. If two patches from these two rows are similar, we merge them and then move on to process the next patch in row_i until the last one. After that, the process starts in row_{i+1} and is repeated until it reaches the last row. The result after the second pass is shown in Fig. 5 (right).

V. FAÇADE REGION SELECTION

After the merging process, we obtain a few coherent image regions which approximate various objects in the scene. Given our objective, which is to delineate the

region of the main façade of interest, the last step in our method is to identify a trait which is unique to the façade region and distinguish it from other regions generated in Section IV. As we mentioned in the introduction, most horizontal and vertical line segments in a building image come from the building (refer to the resulting line segment groups in Fig. 2 for an example). Based on this observation, a façade region selection criterion can be to select the region that scores the greatest number of horizontal and vertical line segments. In the example image of Fig. 5 (right), region 7 has 158 horizontal and vertical line segments, far more than 63 within region 10, which has the second most such line segments (Fig. 6 right). The final look of region 7 is displayed in Fig. 6 (left), which overlaps with the façade region well.

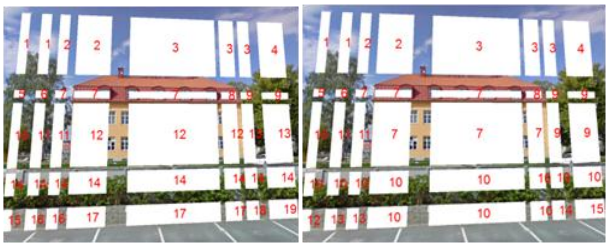

Figure 5. (left) patches after the row merging; (right) after merging between rows. In both figures, patches with the same number are merged.

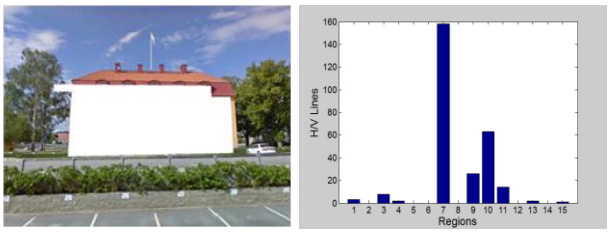

Figure 6. (left) the final region representing the façade; (right) horizontal and vertical line counts (ordinate) of each region (abscissa) from Fig. 5 (right)

VI. Experimental Validation and Results

To validate our method, we conducted a two-fold experiment. At first, the ground truth for the experiment was established by letting a human observer (O1) manually delineate the region of the predominant façade in street-view images of buildings. To that end, the user was provided with a software tool that allowed manual editing of a closed polyline contour in the images. The user was instructed to identify and coarsely delineate the most prominent façade in every image. While the polyline tool generally allowed for almost pixel accurate drawings of façade borders, regions were typically delineated by simple polyline contours with a few control points. Examples of manually defined façade regions can be seen in the middle column of Fig. 8.

For the testing data we used images from the ZuBud database [19]. This database contains 1005 street view images from 201 buildings. Most buildings were taken from different viewing angles with a few exceptions taken under different lighting conditions or with different cameras. To limit workload and to maintain the user's attention during manual segmentation, we chose 201 images from the database. Each of these selected images represents one of the 201 buildings in order to assure full coverage of the variations in the different building scenarios of the ZuBud image database. The image selection criterion hereby is predominance of the major façade in an image. This criterion is plausible considering the intended use case in an MAR system where the user would point the hand-held device towards the building of interest (compare Fig. 1).

In the first part of the experiment another human observer (O2) manually delineated façade regions according to the same procedure as described above. In the second part our algorithm was executed to automatically perform the same task. This allows us to compare the agreement between our algorithm and a human observer against the agreement between two human observers.

Region detection from the previous steps amounts essentially to a binary classification. Hence we established the confusion matrix for every image by counting the number of correctly and incorrectly classified pixels (in comparison with ground truth). Subsequently, we derived the following metrics from the confusion matrix for detection result analysis: accuracy, precision, recall and specificity.

The box-plots in Fig. 7 give an account on the positives and negatives as well as the derived metrics in both tests. Exact average numbers for the metrics used in both tests are summarized in Table II. Fig. 8 shows some images from the dataset along with their corresponding ground truth (O1) and automatic delineation.

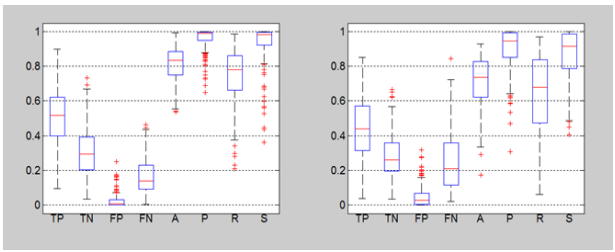

Figure 7. Summary of false and positive classifications as well as various metrics from two tests. (left: human detector (O2) vs. ground truth (O1); right: our method vs. ground truth (O1); A: Accuracy, P: Precision, R: Recall and S: Specificity)

VII. Discussions and Conclusions

Fig. 8 rows a~e show that the method presented in this paper is capable of excluding most non-building objects in the scene (e.g., trees and electricity wires) while complying with the shapes of the façades. Since we scan the image along lines incident with vanishing points, another advantage is that it is robust against perspective distortions.

Nevertheless, the algorithm does not result in a crisp shape representation of buildings. In contrary, revisiting our initial objective of delineating the region of the dominant building façade in street view images, we do not expect total agreement of our algorithm with an observer established ground truth. In fact, even for human observers the decision in regard to what constitutes a façade region is not always congruent (e.g., in cases of

partially occluded buildings). For instance, comparing the images delineated by O2 with the ground truth images delineated by O1 results in a mean accuracy of 0.8136. In light of this the mean accuracy of 0.7229 achieved by our algorithm in comparison with O1 seems to be quite a promising agreement. Visual inspection of classification results in Fig. 8 provides some explanations for differences between human delineation and the algorithm. Since our algorithm is based on image fragments that are bound by the major vanishing lines, in Fig. 8 row a~e, it fails to capture façade boundaries that deviate from this overall regular façade structure, such as diagonal or curved pediments. These structures tend to be either partially captured (leading to false negatives) or to be contained within rectilinear patches that also contain background (leading to false positives). Meanwhile, Fig. 8 row f and g show other limitations of the automatic detection. In row f, the façade does have different types of layouts. In this case, our method treats different kinds of layouts as different homogenous regions, thus, different façades and returns the most evident one as the final façade region according to Section V. Some façades have more irregular layouts, e.g., the one showed in row g. In such case, there will not be many homogenous regions detected on the façade region, which can lead to a complete misclassification.

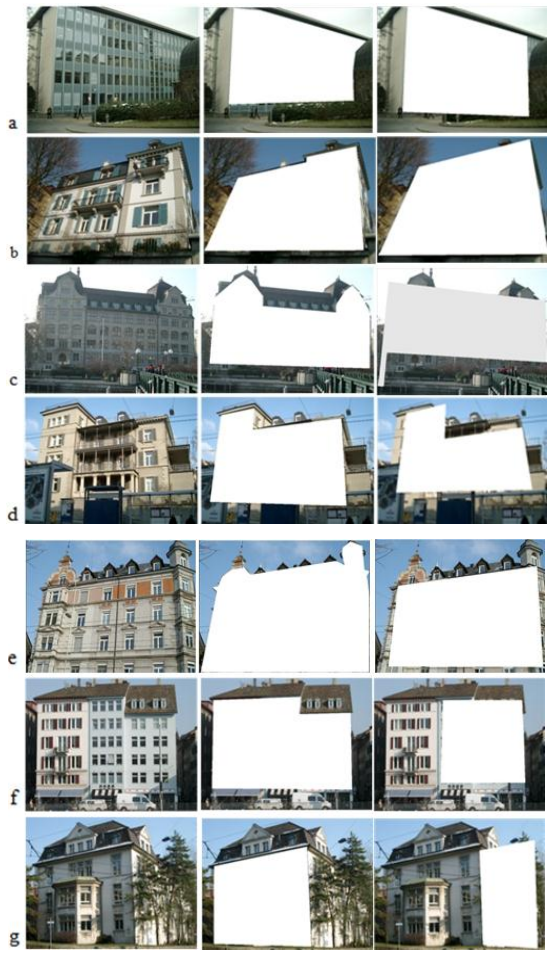

Figure 8. Examples of the detection results with their respective ground truth (left column: original images; middle column: ground truth (O1); right column: results from our method)

Although false positive (FP) and false negative (FN) in an ideal classifier should be close to zero, we can still accept a higher FN while the FP should be minimal. In other words, for any subsequent image recognition process, we accept to miss some of the actual façade pixels, rather than incorrectly including background into the recognition (FP) process, as it contains features which are not characteristics of the façade. If we accept manual delineations of O1 as a suitable ground truth, this characteristic (lower FP compared to higher FN) is equally evident for the human classifier and the automatic classifier (see Fig. 7 and Table II). Similarly, precision is a more relevant metric to look at, which is at clearly higher levels than recall in both comparisons.

TABLE II. MEAN VALUES FOR VARIOUS MEASURES IN BOTH TESTS (A: ACCURACY, P: PRECISION, R: RECALL and S: SPECIFICITY)

	FP	FN	A	P	R	S
O2 vs. O1	0.0213	0.1651	0.8136	0.9601	0.7482	0.9338
Our method vs. O1	0.0495	0.2276	0.7229	0.9027	0.6665	0.8709

As discussed above, we are aware of some potential for further refined region detection in our method. Given the purpose of the development, our ongoing research is directed towards evaluating how the façade region detection method presented here can help improve building identification and localization in the subsequent computer vision pipeline.

ACKNOWLEDGMENT

This research was supported by funding from Faculty of Engineering and Sustainable Development at University of Gävle. The authors acknowledge Dr. Julia Åhlén and Prof. Ewert Bengtsson for fruitful technical discussions.

REFERENCES

[1] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 934-941.

[2] G. Takacs, V. Chandrasekhar, B. Girod, and R. Grzeszczuk, "Feature tracking for mobile augmented reality using video coder motion vectors," in *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 141-144.

[3] D. N. Ta, W. C. Chen, N. Gelfand, and K. Pulli, "SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2937-2944.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*, Springer Berlin Heidelberg, 2006, pp. 404-417.

[6] V. Chandrasekhar, G. Takacs, *et al.*, "Compressed histogram of gradients: A low-bitrate descriptor," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2504-2511.

[7] D. M. Chen, G. Baatz, *et al.*, "City-Scale landmark identification on mobile devices," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 737-744.

[8] T. Korah and C. Rasmussen, "Analysis of building textures for reconstructing partially occluded façades," in *Proc.10th European Conference on Computer Vision*, Marseille, France, 2008, pp. 359-372.

[9] A. Wendel, M. Donoser, and H. Bischof, "Unsupervised façade segmentation using repetitive patterns," in *Proc. 32nd DAGM Conference on Pattern Recognition*, Darmstadt, Germany, 2010, pp. 51-60.

[10] P. David, "Detecting planar surfaces in outdoor urban environments," ARMY Research Lab Adelphi MD. Computational and Information Sciences Directorate, Tech. Rep., 2008.

[11] Y. Li and L. G. Shapiro, "Consistent line clusters for building recognition in CBIR," in *Proc. 16th International Conference on Pattern Recognition*, 2002, pp. 952-956.

[12] H. H. Trinh and K. H. Jo, "Image-based structural analysis of building using line segments and their geometrical vanishing points," in *Proc. International Joint Conference SICE-ICASE*, 2006, pp. 566-571.

[13] J. A. Delmerico, P. David, and J. J. Corso, "Building façade detection, segmentation, and parameter estimation for mobile robot localization and guidance," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1632-1639.

[14] P. D. Kovesi. MATLAB and octave functions for computer vision and image processing. [Online]. Available: http://www.csse.uwa.edu.au/~pk/research/matlabfns/

[15] R. Pflugfelder and H. Bischof, "Online auto-calibration in man-made worlds," in *Proc. Digital Image Computing: Techniques and Applications*, 2005, pp. 519-526.

[16] H. Wildenauer and M. Vincze, "Vanishing point detection in complex man-made worlds," in *Proc. 14th International Conference on Image Analysis and Processing*, 2007, pp. 615-622.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[18] P. P. Ohanian and R. C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognition*, vol. 25, no. 8, pp. 819-833, 1992.

[19] H. Shao, T. Svoboda, and L. V. Gool, "ZUBUD-Zurich buildings database for image based recognition," *Technical Report No. 260*, Swiss Federal Institute of Technology, 2003.

Fei Liu received his M.Sc. degree in computer science from Department of Information Technology, Uppsala University, Sweden in 2010. He is now a Ph.D. candidate at University of Gävle. His current research interests are pattern recognition and computer vision.

Stefan Seipel graduated with an M.Sc. in Medical Informatics from Heidelberg University, Germany, in 1992. He obtained a Ph.D. in theoretical medicine from the medical faculty at Heidelberg University, Germany, in 1997 for his thesis on 3D simulation of surgical procedures and surgical instrument navigation. He was appointed professor of Computer Graphics at Uppsala University, Sweden, in 2003. He currently holds the position of a Professor of Computer Graphics and Head of the research group in Geospatial Information Technology at the University in Gävle, Sweden, as well as he has a part time position as Professor of Computer Graphics at Uppsala University, Sweden. Previous academic posts include Postdoctoral Researcher in Human-Computer Interaction, Senior Lecturer in Graphics, and Senior Lecturer in Visualization. His published research has been directed towards applied computer graphics and interactive visualizations within the medical field and various other fields of applications. His particular research interests include usability of interactive visualizations, spatial and temporal visualization techniques, virtual reality and augmented reality, as well as geographical visualization. Prof. Seipel is a member of the Eurographics and member of the Swedish Computer Graphics Association, which he is currently co-chairing.

