Automatic Extraction of Regular Grids from Rectified Facade Image

Anton Yakubenko, Ivan Mizin, Anton Konushin
Graphics & Media Lab, Department of Computational Mathematics and Cybernetics
Lomonosov Moscow State University, Moscow, Russia
{toh, ivan.mizin, ktosh}@graphics.cs.msu.ru

Figure 1: Input rectified image of building's facade (left) and automatically recognized structure, presented as a set of non-overlapping grids of similar windows (right).

Abstract

Automation of facade images interpretation is a crucial task for efficient 3D modeling of urban areas. Information about regularities in facade images can be used for image completion of texture areas occluded with foreground objects, 3D model compression and texture quality enhancement. Existing approaches cannot handle the whole variety of buildings facades and often fail in cases of occluded textures. We present a novel algorithm for automatic extraction of facade structure from single rectified image. We describe facade structure as a set of non-overlapping grids of similar facade parts, usually containing windows. The most frequent possible grid cells sizes and grid cells centers are estimated by detecting and analyzing similar rectangles in input image. Greedy iterative algorithm sequentially chooses grids with maximum quality measure that incorporates empirical observations about similarities and symmetries. Comparison on the database, created from publically available sources with additional complex examples from our photos, shows that the proposed algorithm outperforms two other state-of-the-art methods and can handle occlusions.

Keywords: *Facade, Structure, Regularity, Grid, Lattice, Urban, 3D City Map, Image-based, Rectified, Texture, Single-view, Occlusion, Recognition, Reconstruction.*

1. INTRODUCTION

Efficient mass-production of 3D cities maps is one of the topical problems at the junction of geo informatics and computer vision. 3D maps are natural evolution of ordinary flat 2D maps. They can be used in a variety of common geo informational tasks, like navigation, city planning, etc., and also afford new opportunities for virtual and augmented reality, location-based services, etc.

3D maps topic has gained a lot of attention in last few years. Common Internet users use 3D geo information web services like Google Maps [36] and Bing Maps [37]. Industrial software leaders like Autodesk develop special products for 3D city maps management like LandXplorer [38]. Joint community develops CityGML [39] standard for the modeling and exchange of 3D city and landscape models.

3D models of urban buildings are the main content for 3D geo information systems. There are multiple ways for acquisition of these models. They can be created from sparse terrestrial photos [3, 40] automatically or manually using photogrammetric and 3D modeling software. These geometry modeling tools are usually coupled with Adobe Photoshop or similar photo editing software for texture editing. More automated approaches include processing of point clouds [4] and video streams [5] from land based mobile mapping systems. Aerial imagery and aerial LIDAR point clouds can also be used as source data [6].

3D model of urban building can be roughly divided in two parts – roof and facades. Roofs can be modeled from aerial imagery [7] or LIDAR point clouds using manual tools or by automatic planes fitting or template matching. Reconstruction of building facades is a more complicated task. But buildings facades frequently have structure and regularity. This semantic knowledge can help solving a number of problems arising in facade modeling, processing and interpretation.

Facade textures are often occluded by foreground objects like vegetation, advertisement billboards, electrical cables, other buildings, bulging parts of the building, people, cars, etc. Occlusion areas can be segmented and reconstructed [8, 13] by comparing similar parts of the facade. Textures created from aerial views often lack quality and resolution on the lower floors of the building. If building's floor were extracted from image these textures can be enhanced [9] by transferring texture information from higher floors to the lower.

Detected facade elements, like windows and balconies, can be replaced by more detailed template 3D models from a database for overall visual quality improvement [10]. Knowledge of repeated texture parts and according geometry elements can be used to compress 3D model geometry and texture without visible loss of quality [11] that is required in web applications. Facade structure information can be also used for image-based geo locating [12], and other tasks.

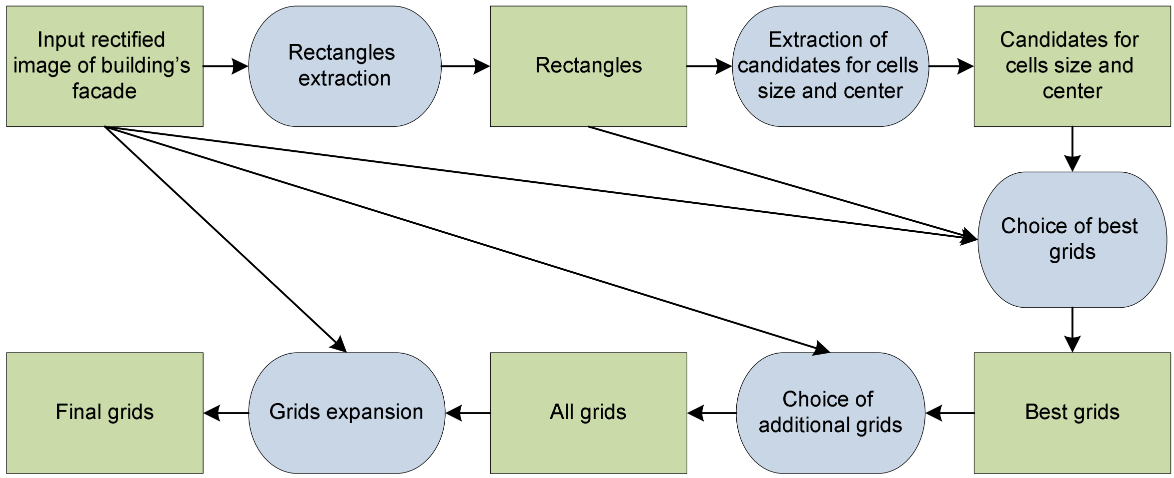

Figure 2: Scheme of the proposed algorithm. Green rectangles show data flow. Blue rounded rectangles show algorithm's parts.

The paper is organized as follows. Brief analysis of the related work is given in Section 2. We present a novel method for automatic determining of a set of non-overlapping grids of similar rectangular facade parts in a given single rectified image of urban building's facade in Section 3. Results of our algorithm and comparison with two state-of-the-art methods [1, 2] results are given in Section 4. Finally, we give the conclusions and discussion in Section 5.

2. RELATED WORK

Repeatability and regularity are the main cues to deal with in facade interpretation tasks. They can be frequently met in man-made environments, not only in urban scenes. The classic solution for extraction of regularity from a single image of a plane object without perspective distortions is analysis of autocorrelation function peaks [19]. This approach treats image globally and works only if regularity is strict, occupies large part of image and there are almost no occlusions.

To deal with near-regular textures, perspective distortions and partial occlusions sparse set of image patches can be analyzed. These patches can be sampled randomly [20] or chosen by interest point [1] or rectangles detector [8]. Similar patches can be grouped together to fit homography [12], grid [8, 22], near-regular lattice [1, 20] or more complex models [2]. Patches are compared using metrics like SSD (sum of squared distances) [21] or NCC (normalized cross-correlation) [1, 22], or by matching descriptors like SIFT [12]. Depending on the complexity of model it can be estimated using Hough transform [23], RANSAC [12], iterative lattice spreading [1] or MRF (Markov random field) optimization [8]. These approaches have shown good results on a variety of images. Due to general problem formulation and using only small local patches for analysis results of these algorithms dramatically deteriorate on facades with complex look of similar parts, for example older buildings, several regular grids, large occlusions.

Approaches, mentioned above do not focus on facade images particularly. A group of methods aims to detect windows, which frequently form the facade structure. This can be done by training Viola-Jones like detector using Haar features and SVM ([24] or AdaBoost [25]. Windows differ greatly from each other, thus these approaches work acceptably for small databases of similar images. To deal with this problem incremental learning approaches were suggested [26], but they require user interaction.

Heuristic approaches make use of similarity of windows corners [27] and presence of gradients on windows perimeter [28]. Akaike's information criterion [27] or minimum description length [2] can be used to consider rows and columns regularities. Windows detection can be also formulated as optimal labeling problem [29] or as problem of finding optimal non-overlapping subset or rectangles [30]. Current windows detection results do not allow using them as final result. But detected windows can be used as lower level information for more intelligent algorithms, which will take regularities into account. General semantic image segmentation [31] and wall area extraction [32] results are also too far from desirable.

State-of-the-art algorithms in automatic facade interpretation try to describe input rectified facade image as a higher level model. This model can be presented as partition of facade image into floors, tiles, windows, etc. More generally this partition can be described as grammar inference [9, 33]. The choice of grammar rules (for example vertical or horizontal splits) and their parameters (coordinates of splits) can be done by reversible jump Markov Chain Monte Carlo (rjMCMC) sampling [33] and comparing neighboring candidates, for example, using mutual information and edges inside and between candidate patches [9]. Metric information can also reduce parameter space [9]. This is one of the most promising approaches, but current results can hardly handle occlusions and strongly depend on the grammar rules formulation and inference criteria.

3. PROPOSED METHOD

We take rectified image of building's facade as input. In assumption of plane facade, rectified image (Figures 3, 4) compensates all projective distortions. This image can be acquired directly from the already created 3D model as its texture or from input photo. The latter can be done by either manually selecting four points in photo, which correspond to a rectangle in the 3D scene, or by automatic algorithms [14], which find lines in image and group them according to vanishing points.

Our aim was to develop algorithm which would work properly on a wide variety of rectified facade images with occlusions. Thus it should use not very strict model. We have chosen a set of non-overlapping grids of similar rectangular image parts (Figure 1) as a simple and flexible model, which describes the absolute majority of urban buildings. Grid lines are parallel to image axes.

Thus each grid is described by 6 parameters:

(cx, cy) – X and Y coordinates of the top left corner of the grid,

(sx, sy) – cells size,

$(rows, columns)$ – number of rows and columns in the grid.

The scheme of the proposed algorithm is shown in Figure 2. We find rectangles in input image (Section 3.1). We choose the most frequent distances between them as possible grid cells size (sx, sy). In assumption that some of the rectangles are windows we remember their centers as possible grid cells centers. In combination with grid cell size this information reduces number of possible (cx, cy) values. Thus we can significantly reduce the total number of grids for further analysis (Section 3.2). We sequentially choose the best grid non-overlapping with already extracted grids considering neighboring cells similarity, cells symmetry and existence of rectangle in the middle of cells (Section 3.3). Due to sampling and grid quality constraints some grids can be missed. We search for additional grids with cells, which are similar to already extracted grids (Section 3.4). We than expand extracted grids using weaker thresholds to handle occlusions (Section 3.5). Algorithm's stages are described in details below.

Figure 3: Input image.

Figure 4: Rectified image.

3.1 Rectangles extraction

Existing approaches like Hough transform [34] can be used for extraction of rectangles with sides parallel to image sides. But we can extract even better rectangles. We build image pyramid by performing image downsampling with factor 2 and factor 4. We perform separate vertical and horizontal edge detection for each image in the pyramid by using Sobel operator. Pixel in the original image is labeled as edge pixel only if it is edge pixel in all the images of image pyramid. This approach filters out small and noisy edges, including undesirable edges between regular structures like wall bricks (Figure 5).

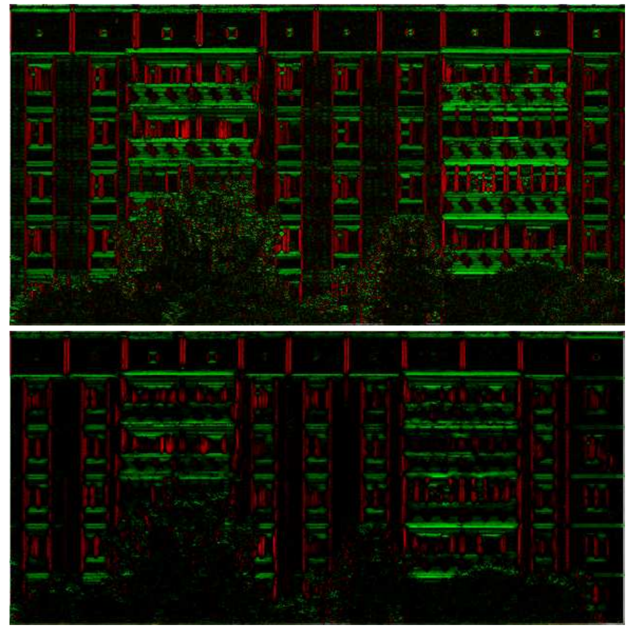

Figure 5: Extracted vertical (red) and horizontal (green) edges applied only to original resolution (top) and applied to image pyramid (bottom).

If pixel has enough horizontal edges in its right neighborhood and enough vertical edges in its bottom neighborhood, it is labeled as top left corner candidate pixel. Other three corners types can be processed in the same way. We iterate through all pairs of corners. If this pair can form a rectangle, i.e. corners lie on the same horizontal or vertical line, and have proper type (for example, are "top left" and "bottom left" corner accordingly), the distance between the corners is stored. There are about ten thousand of such corners in current example. We can then perform exhaustive search of rectangles with selected corners and sizes, which form the resulting set of rectangles (Figure 6). There are about thousand rectangles in current example.

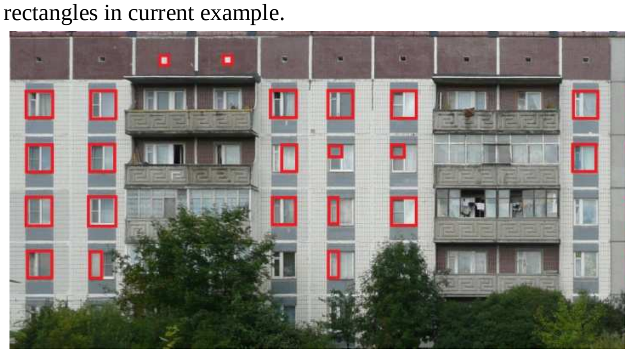

Figure 6: Some of the extracted rectangles.

3.2 Extraction of candidates for cells size and center

We compare each pair of image areas corresponding to rectangles with equal sizes using normalized cross-correlation (NCC) and store the distances between them if NCC value is large enough. The most frequent distances are selected as candidates for cells sizes (Figure 7).

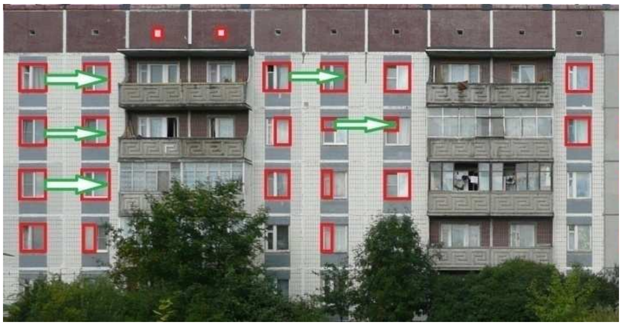

Figure 7: Extracted most frequent distances between rectangles.

Resulting grids are naturally looking if its cells have facade element, for example, a window, in their center. Such centering of grids is not obligatory, but it reduces possible grids locations and avoids losing of a single side row or column near image bounds due to large grid shift.

3.3 Greedy iterative choice of best grids

The number of grids with previously determined sizes and centers can be analyzed exhaustively. Grid quality G is computed as follows:

$$G = (M * N)^{-t_1} \sum_{i=1}^{M} \sum_{j=1}^{N} Q_{i,j},$$

Where M is the number of rows in the grid, N is the number of columns in the grid, t_1 = -0.8 is a regularizing parameter that penalties too small grids and $Q_{i,j}$ is quality measure for cell $c_{i,j}$:

$$Q_{i,j} = Similarity(c_{i,j}, c_{i-1,j}) + Simimilarity(c_{i,j}, c_{i,j-1}) + Symmetry(c_{i,j}) + HasRect(c_{i,j}).$$

$Similarity(c_{i,j}, c_{k,l})$ is similarity measure between cells $c_{i,j}$ and $c_{k,l}$. It is equal to NCC of corresponding image patches if it is larger than threshold t_2, or -t_2 otherwise. $Symmetry(c_{i,j})$ is $c_{i,j}$ cell's measure of symmetry along horizontal direction. It is equal to NCC of left and right parts of the cell if it is larger than threshold t_2, or -t_2 otherwise. $HasRect(c_{i,j})$ is equal to t_2 if it there is a rect inside $c_{i,j}$ cell and -t_2 otherwise. We took $t_2 = 0.4$ in all of our experiments. Thresholds t_1 and t_2 were selected empirically. One can vary thresholds values in parts cell quality terms.

We sequentially choose the best grid that does not overlap already extracted grids (Figure 8). We stop when new grid quality is much lower than the quality of the best grid.

3.4 Search for additional grids

In some cases some grids are missed. So we perform a search for additional grids (Figures 10, 12). We make an assumption, that they are similar to already extracted grid. We compute median cell for each already extracted grid and correlate it with the part of the image not covered by grids. If the maximum of correlation map is relatively large we start a new grid with the center in the point of this maximum. We iterate this algorithm until no new grids can be found.

3.5 Grids expansion

Extracted grids do not usually spread over the occluded regions. For each grid we analyze four grids which are created from the original grid by adding single row or column at its side. If this grid does not intersect other grids and new cells are similar to their neighbors with a lower threshold, then we replace original grid with the expanded one (Figures 9, 11, 13, 14). We apply this algorithm several times. First it is applied after best grids are chosen to join partly occluded cells to them before searching for additional grids (Figure 9). Then it is applied after any new additional grid has been found (Figures 11, 13). Finally it is applied with extremely low threshold at the very end to recover much occluded cells (Figure 14).

Figure 8: Best grid.

Figure 9: Expansion of best grid.

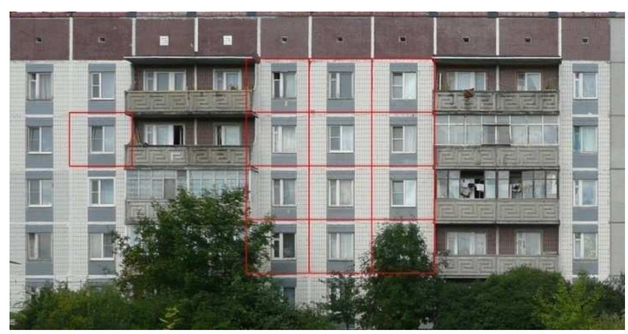

Figure 10: Start of additional grid.

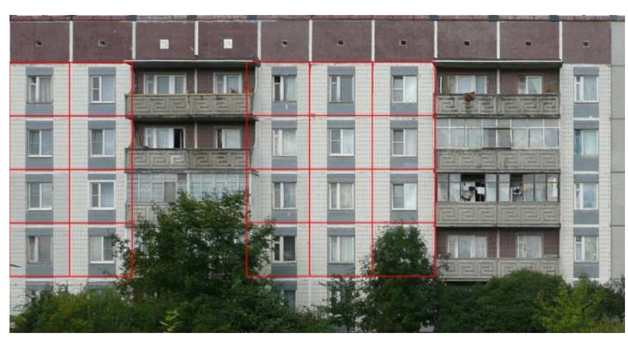

Figure 11: Expansion of additional grid.

Figure 12: Starting another additional grid.

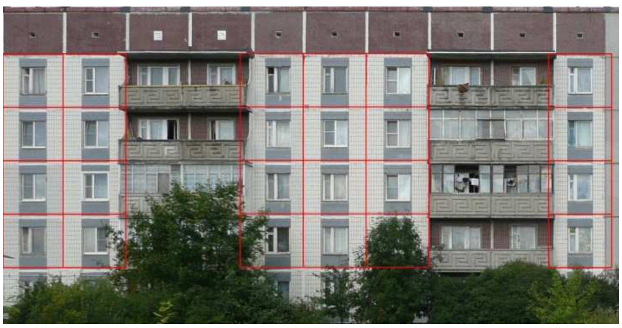

Figure 13: Expansion of additional grid.

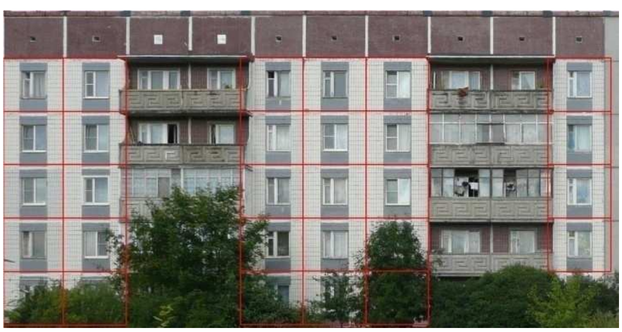

Figure 14: Final expansion to occluded areas.

4. EVALUATION

4.1 Database

Unfortunately, there is neither conventional metrology nor at least image database for testing and comparison of facade interpretation algorithms. We have chosen 25 images from Facade Data Base Vienna [15], eTRIMS Image Database [16], ZuBuD Image Database [17], CGTextures.com [18] web portal, and our own photos. Thought it is a small database, but it is representative, including images of old and modern architecture, built of various materials, with different numbers of floors and tiles, different regularities types, taken under various lighting conditions. All images were manually rectified.

4.2 Competitors

Problem statement notably differs among various scientific papers. We have chosen [1] as a modern representative of algorithms for recovering general regularities. It is based on interest points extraction and comparison. Similar shifts between similar points are stored. Frequent shifts become hypotheses for starting lattice size. Lattice is build iteratively by adding new nodes, which are similar to existing ones and met lattice size condition. Some parts of this algorithms inspired extraction of candidates for cells sizes and centers positions in our algorithm.

Algorithm [2] represents the trend of using more complex model for facade description as a linear combination of axis parallel basis translations and the appropriate coefficients and uses MDL principle for regularization. This algorithm extracts SIFT features and get the pairs of similar features. Best parameters of the model move similar points to each other.

4.3 Grid cells sizes extraction evaluation

Competitor methods do not try to center grids relatively to windows, thus they can lose some rows or columns. We evaluate grids location and number of cells in Section 4.4. In this section we evaluate only the results of cells sizes extraction. The proposed algorithm finds right grids cells sizes among 10 most frequent steps for all images in test database. But some of proper grids are missed in the final result. We consider result of algorithm right if there is a grid with proper cells size in the final set of grids. Results of grids steps extraction evaluation are shown in Figure 15.

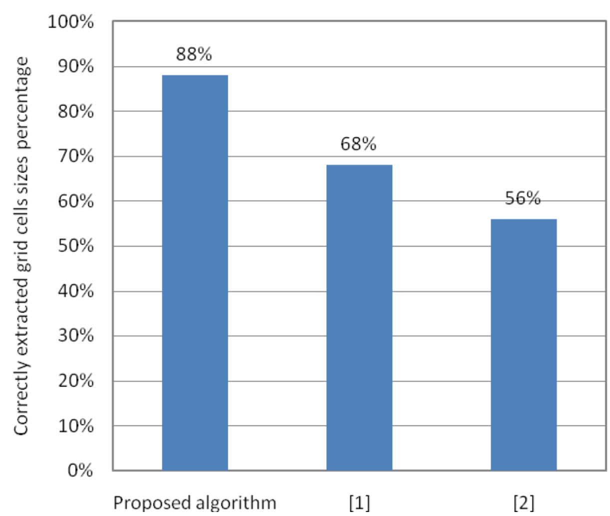

Figure 15: Grid cells size extraction evaluation results.

4.4 Evaluation of percentage of correctly extracted similar elements

We consider extracted grid to be correct if it has visually similar cells, which cannot be divided further, and has right number of rows and columns. Local shifts of grids are not taken into account, because it is not vital for further processing of extracted information, for instance, for performing texture completion. But these shifts can influence the number of rows and columns in the extracted grid. Total number of cells in each image is known. For evaluation we can count ratio of number of correctly extracted cells to total number of cells (see Figure 16). The results are shown in Figure 17. We do not provide comparison to [2], because it requires manual setting of starting point for grid positioning.

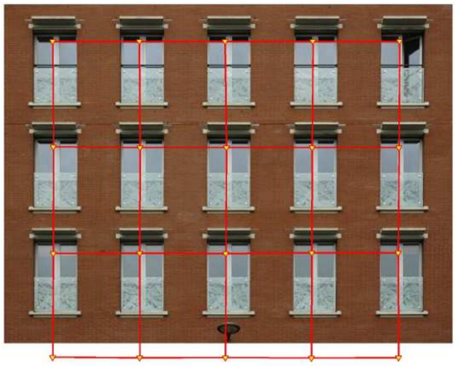

Figure 16: Example of [1] results. Ground truth grid contains 15 cells; this grid contains only 12 cells. It means that 80% of cells were detected correctly.

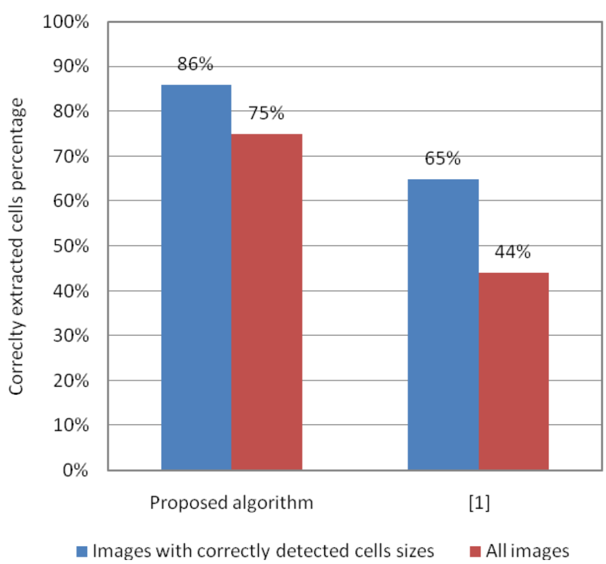

Figure 17: Evaluation of percentage of correctly extracted similar elements.

5. CONCLUSIONS AND DISCUSSION

We have proposed a novel fully automatic algorithm for extraction of information about repeatability of similar rectangular image patches from a single rectified image of urban building's facade. Comparison with two competing algorithms showed advantages of our method.

This can be first of all explained by focusing only on rectified facade images, but not a general class of images with more complex repeatability models and image distortions. In contrast to sophisticated grammar-based models we have introduced a simple model for facade description – a set of non-overlapping grids, which can describe the majority of buildings facades. Proposed model quality measure incorporates empirical knowledge about building's facade, including similarity between neighboring cells, horizontal grid symmetry and existence of rectangle in the center of each cell.

We make use of analyzing rectangles, which can be frequently met in facade images and showed to be more robust than feature points or random patches.

We reduce possible number of grid cells size by gathering the most frequent distances between extracted rectangles. Rectangles positions give us candidates for possible grid position. Due to such sampling of parameters space we can use exhaustive search for best grids. The proposed method can handle occlusions by lowering thresholds for grid quality in grid spreading process.

Though current results are promising (75% of cells are extracted correctly) there are still some problems left. Due to iterative greedy choice of best grids some grids are missed and not all occlusions are handled. Also a grid divided by another grid is considered as two separate grids. Global optimization formulation could do better. Rectangles are rather robust and easy to find, but additional detection of windows, based on machine learning techniques, can improve the overall robustness. We assume facade and facade elements to be almost flat. Bulging balconies and deeply sunken windows requires 3D analysis [35]. Non-overlapping grids model is enough for texture completion and compression tasks. But it can be enhanced by adding hierarchy and more complex models, for example, repeated pairs of elements. Higher level semantic information could be also extracted for more intelligent tasks.

6. ACKNOWLEDGMENTS

The work was partially supported by the Federal Target Program

"Scientific and scientific-pedagogical personnel of innovative Russia" and Russian Foundation for Basic Research grants 09-01-92470-MHKC_a and 08-01-00883-a.

7. REFERENCES

[1] Minwoo Park, Robert T. Collins, and Yanxi Liu "Deformed Lattice Discovery Via Efficient Mean-Shift Belief Propagation", Proc. of the 10th European Conf. Computer Vision (ECCV), Vol. 2, pp. 474 – 485, 2008.

[2] S. Wenzel, M. Drauschke, W. Forstner "Detection of Repeated Structures in Facade Images", Pattern Recognition and Image Analysis, Vol. 18, No. 3, pp. 406 – 411, 2008.

[3] F Remondino, S El-Hakim "Image-based 3D modelling: a review" Photogrammetric Record, Vol. 21, No. 115, pp. 269 – 291, 2006.

[4] C. Frueh, A. Zakhor "An Automated Method for Large-Scale, Ground-Based City Model Acquisition" International Journal of Computer Vision, Vol. 60, No. 1, pp. 5 – 24, October 2004.

[5] Nico Cornelis, Bastian Leibe, Kurt Cornelis, Luc Van Gool "3D Urban Scene Modeling Integrating Recognition and Reconstruction" International Journal Of Computer Vision, Vol. 78, No. 2 – 3, pp. 121 – 141, 2008.

[6] J. Hu, S. You, U. Neumann, K.K. Park "Building modeling from LIDAR and Aerial Imagery" ASPRS'04, Denver, Colorado, USA. May 23-28, 2004.

[7] Theo Moons, David Frere, Jan Vandekerckhove, Luc Van Gool "Automatic Modelling and 3D Reconstruction of Urban House Roofs from High Resolution Aerial Imagery" In Proc. 5th European Conference on Computer Vision (ECCV), 1998, Freiburg, Germany, pp. 410–425.

[8] T. Korah, C. Rasmussen "Analysis of Building Textures for Reconstructing Partially Occluded Facades", Proc. of the 10th European Conference on Computer Vision (ECCV), Part I, pp. 359–372, 2008.

[9] P. Muller, G. Zeng, P. Wonka, and L. V. Gool. "Image-based Procedural Modeling of Facades", Proceedings of ACM

SIGGRAPH'07, ACM Transactions on Graphics. 2007. Vol 26. N. 3. P. 1 – 9.

[10] Julien Ricard, Jerome Royan, Olivier Aubault "Visualization of Real Cities Based on Procedural Modeling", IEEE Virtual Reality Workshop on Virtual Cityscapes: Key Research Issues in Modeling Large-Scale Immersive Urban Environments, 2008.

[11] Huamin Wang, Yonatan Wexler, Eyal Ofek, Hugues Hoppe "Factoring Repeated Content Within and Among Images", ACM Transactions on Graphics, vol. 27, no. 3, pp. 1–10, 2008.

[12] Schindler G., Krishnamurthy P., Lublinerman R., Liu Y., Dellaert F. "Detecting and Matching Repeated Patterns for Automatic Geo-tagging in Urban Environments" Proceedings of CVPR 2008, pp. 1 – 7.

[13] Vadim Konushin, Vladimir Vezhnevets "Automatic building texture completion", GraphiCon'2007.

[14] Liebowitz D., Criminisi A., Zisserman A. "Creating Architectural Models from Images", In Proc. EuroGraphics, Vol. 18, pp. 39–50, 1999.

[15] Facade Data Base Vienna http://info.tuwien.ac.at/ingeo/research/FdbV/index.htm

[16] eTRIMS Image Database http://www.ipb.uni-bonn.de/projects/etrims_db/

[17] ZuBuD Image Database http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html

[18] http://cgtextures.com

[19] Hsin-Chih Lin, Ling-Ling Wang, and Shi-Nine Yang "Extracting periodicity of a regular texture based on autocorrelation functions" Pattern Recognition Letters, Vol. 18, pp. 433 – 443, 1997.

[20] James Hays, Marius Leordeanu, Alexei A. Efros, Yanxi Liu "Discovering Texture Regularity as a Higher-Order Correspondence Problem", In ECCV, Vol. 2, pp. 522 – 535, 2006.

[21] Thomas Leung, Jitendra Malik "Detecting, localizing and grouping repeated scene elements from an image", In Proc. of European Conf. Computer Vision (ECCV), pp. 546 - 555, 1996.

[22] Frederik Schaffalitzky, Andrew Zisserman "Geometric Grouping of Repeated Elements within Images", In Proc. of BMVC, pp. 13 - 22, 1998.

[23] Andreas Turina, Tinne Tuytelaars, Theo Moons, Luc Van Gool "Grouping via the Matching of Repeated Patterns", In Proc. of CAPR, pp. 250 – 259, 2001.

[24] Bjorn Johansson, Fredrik Kahl "Detecting Windows in City Scenes", ICPR 2002.

[25] Haider Ali, Christin Seifert, Nitin Jindal, Lucas Paletta, Gerhard Paar "Window Detection in Facades", International Conference on Image Analysis and Processing (ICIAP), pp. 837-842, 2007.

[26] Susanne Wenzel, Wolfgang Forstner "Semi-Supervised Incremental Learning of Hierarchical Appearance Models", 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS), Beijing, China, 2008.

[27] Helmut Mayer, Sergiy Reznik "Implicit Shape Models, Model Selection, and Plane Sweeping For 3D Facade Interpretation", Photogrammetric Image Analysis (PIA) 2007.

[28] Sung Chun Lee, Ram Nevatia "Extraction and Integration of Window in a 3D Building Model from Ground View Images", In CVPR, IEEE Computer Society, pp. 113 – 120, 2004.

[29] Jan Cech, Radim Sara "Windowpane Detection based on Maximum Aposteriori Probability Labeling", In International Workshop on Combinatorial Image Analysis (IWCIA), pp. 3–11, 2008.

[30] Filip Korc, Wolfgang Forstner "Finding Optimal Non-Overlapping Subset of Extracted Image Objects", 2008.

[31] Floraine Grabler "Parsing Images of Architectural Scenes", In ICCV, pp. 44 – 57, 2007.

[32] R. G. Laycock, A. M. Day "Towards the Automatic Integration of Ground Level Images into a Virtual Urban Environment", WSCG 2004.

[33] Nora Ripperda, Claus Brenner "Reconstruction of Facade Structures Using a Formal Grammar and RjMCMC", in Pattern Recognition, pp. 750–759, 2006.

[34] Claudio Rosito Jung, Rodrigo Schramm "Rectangle Detection based on a Windowed Hough Transform" Proceedins of the XVII Brasilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04), pp. 17-20.

[35] Luc Van Gool, Gang Zeng, Filip Van den Borre, Pascal Muller "Towards Mass-Produced Buildings Models", Photogrammetric Image Analysis (PIA) 2007.

[36] Google Maps http://maps.google.com

[37] Bing Maps http://www.bing.com/maps/

[38] Autodesk LandXplorer http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=12561246

[39] CityGML http://www.citygmlwiki.org/index.php/Main_Page

[40] Autodesk Photofly http://labs.autodesk.com/technologies/photofly/