Analysis of Building Textures for Reconstructing Partially Occluded Facades

Thommen Korah[1] and Christopher Rasmussen[2]

[1] HRL Laboratories, LLC
Malibu, CA, USA
tkorah@hrl.com
[2] University of Delaware
Newark, DE, USA
cer@cis.udel.edu

Abstract. As part of an architectural modeling project, this paper investigates the problem of understanding and manipulating images of buildings. Our primary motivation is to automatically detect and seamlessly remove unwanted foreground elements from urban scenes. Without explicit handling, these objects will appear pasted as artifacts on the model. Recovering the building facade in a video sequence is relatively simple because parallax induces foreground/background depth layers, but here we consider static images only. We develop a series of methods that enable foreground removal from images of buildings or brick walls. The key insight is to use *a priori* knowledge about grid patterns on building facades that can be modeled as Near Regular Textures (NRT). We describe a Markov Random Field (MRF) model for such textures and introduce a Markov Chain Monte Carlo (MCMC) optimization procedure for discovering them. This simple spatial rule is then used as a starting point for inference of missing windows, facade segmentation, outlier identification, and foreground removal.

1 Introduction

An important step in vision-based architectural modeling [1–3] is the creation of texture maps representing each planar section of a building's facade. A frequent complicating factor is the presence of other, unknown objects in the scene between the camera and building plane—e.g., trees, people, signs, poles, and other clutter of urban environments. In a similar class are objects reflected in building windows. Without explicitly recognizing and removing them, these foreground objects will be erroneously included in the building appearance model, as can be seen in the results of [4, 5] among others. Many artifacts also arise due to the lack of strict constraints on parallelism, continuity of linear edges, and symmetry. With manual intervention one can cut out such features and replace them with nearby symmetric or repeated building features [6]. The larger problem that motivates this paper is whether and how foreground objects can *automatically* be eliminated.
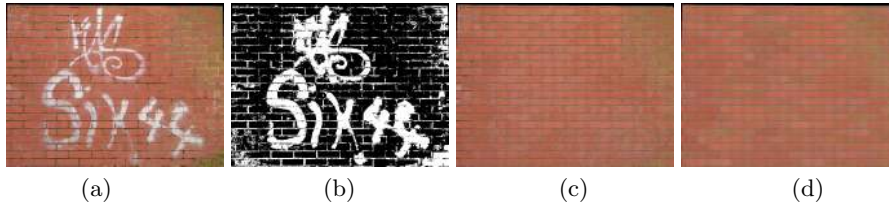
Fig. 1. Virtual graffiti removal. (a) Original image; (b) Automatically detected foreground pixels (c) Tile-aligned exemplar-based inpainting (d) Eigenimage reconstruction. Tiles with $> 25\%$ outliers were sampled. While there is some loss of detail in (d), many local characteristics are retained.

Considering Fig. 1, two major obstacles to overcome for background recovery are (i) identifying the problem areas and, (ii) actually removing foreground objects to reveal the building structure behind them. Given a sequence captured by a translating camera, parallax is an obvious cue to identify foreground objects at different depths [7]. However, even from a single image, humans are adept at "mentally scrubbing" away distracting elements and envisioning the appearance of the obliterated regions.

We make the simplifying assumption that the background is strongly structured and exhibits characteristics of a near-regular texture that dominates the image. This is frequently the case for close-up images of sections of buildings with brick patterns or window grids. The basic idea is that by discovering these textures automatically and collecting statistics on tile appearance, we can automatically segment foreground objects as texture outliers and either reconstruct them from or replace them with unoccluded patterns elsewhere in the texture. We demonstrate how these spatial priors enable subsequent operations such as inference of missing windows, facade segmentation, outlier identification, and foreground removal.

1.1   Previous Work

Modeling of urban and architectural environments has been studied for several years. Government agencies have traditionally used it for development planning or military strategy. The success of recent products like Google Earth, SketchUp, and Microsoft Virtual Earth has enabled urban modeling to be done in a distributed, voluntary, and "wikified" manner. However, the photogrammetric techniques used to compute the geometry of a scene can be brittle when dealing with large-scale and unconstrained urban scenes.

In this work, we ignore geometry and focus on recovering a "clean" mosaic of the facade that can subsequently be used as a texture map. With the exception of the MIT City Scanning Project [2] and the 3D City Model Generation work at Berkeley [5], none of the systems listed previously address issues of missing data, occlusions, perspective or other factors that degrade the visual quality of

the texture maps. Even these two systems employ very crude methods of photometric blending and interpolation with little consideration of topology. There are no safeguards against wall pixels being copied over windows or other misalignment issues. Debevec's view-dependent texture mapping [1] gets around this issue without explicitly handling the occluding elements. Once the problem areas are identified, inpainting techniques based on PDEs [8] or non-parametric exemplar methods [9, 10], combined with some prior knowledge of the architectural domain, offer a principled way to remove large foreground elements.

We specifically try to interpret the building facade from a single static image. Dick at al. [3] were among the first to attempt automatic labeling of architectural elements. Mayer and Reznik [11] focused on facade interpretation using implicit shape models to extract windows. Although both methods used Markov Chain Monte Carlo (MCMC) [12] to simulate the posterior, there was no information about the connectivity between the detected elements. This makes high-level analysis and manipulation difficult. In graphics, split grammars were introduced by Wonka [13] to formally describe the derivation of architectural shapes for procedural modeling. A few researchers combined the grammar-based procedural modeling with the concept of *parsing* images of buildings [14], although inconsistencies and occlusion cause these systems to fail. Recently, Mueller [15] presented an impressive interactive system that takes a single rectified image of a building as input and computes a 3D geometric and semantic model with much greater visual quality and resolution.

The above methods use very specialized models and show examples on a restricted set of images. Like us, they require that the facades contain repetitive elements (typically windows) exhibiting regularity. However, the input needs to be clipped and rectified before any processing. The case of detecting occluding elements or seeing through them is seldom handled, primarily because of stringent assumptions on the nature of symmetric patterns. Instead of tuned window detectors, we develop a generic grouping framework to detect near-regular lattice structures. Texture-specific models (such as those for windows under perspective) can be easily incorporated, while still being robust to occlusions and small irregularities. Reliably discovering the underlying symmetry in texture and structure is crucial for facade analysis and reconstruction.

Approaches based on RANSAC [16] and the cascaded Hough transform [17] have been used to find regular, planar patterns. However, many buildings in our test set do not exhibit the "checkerboard" style consistency these methods require. Our definition of a lattice structure is derived from the literature on Near-Regular Textures (NRT) [18, 19]. An NRT is a geometric and photometric deformation from its regular origin of a congruent wallpaper pattern formed by 2D translations of a single tile or *texel*. Any warped 4-connected lattice constitutes an NRT. An iterative algorithm for NRT discovery by higher-order correspondence of visually similar interest points was described by Hays et al. in [20]. Besides being computationally expensive, directly applying their method might retrieve tiles which do not correspond exactly to semantically meaningful units. A technique to extract texels from homogeneous, 2.1D, planar texture
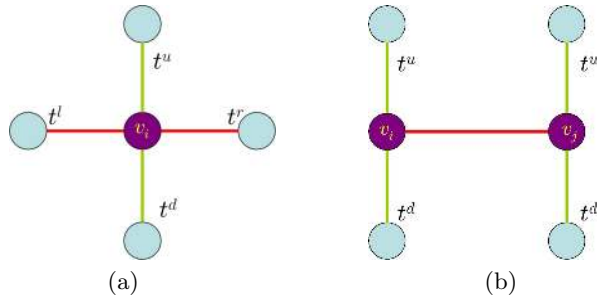
Fig. 2. MRF (a) local node and (b) clique potential "neighbor" vectors.

with partial occlusion was presented in [21], but assumed that the placement of the texels was statistically uniform without any global structure. Bayesian approaches based on Markov Random Fields [22] have also been utilized for localizing grid structures in genome sequencing [23]. We adopt a similar approach that more generally applies to many different types of NRTs. Lin and Liu [24] used an MRF model to track dynamic deformable lattices, but assumed that the texels had already been discovered,

The next section describes our MRF/MCMC approach for efficient NRT discovery. We then describe a series of methods, woven together by the common goal of background recovery, to extract additional properties of the facade. This kind of information efficiently computed on-board an autonomous platform could also assist in view planning and focus of attention control. Finally, results are shown on a variety of building images.

2 Discovering Building Texture

For brick images (Fig. 1), we used an efficient power spectrum approach [25] that although simple, worked well on a variety of images. This section describes our more general MCMC algorithm for discovering rectangular NRTs. Previous algorithms for texture discovery [16, 17] have used interest point or corner detectors to demarcate potential texels. Since we are interested in rectangular shaped windows, straight lines are first detected in the input image. Similar to [26], rectangles are hypothesized from pairs of approximately parallel line segments resulting in hundreds of rectangles. Under perspective, each rectangle v_i is defined as $(p_k^0, p_k^1, p_k^2, p_k^3)$ denoting the 4 corners of a quadrilateral in anti-clockwise order with p_{k1} as the upper-left position. We do not represent them using vanishing points as in [26] to avoid estimation errors in earlier stages from cascading through the pipeline.

The process results in a couple of thousand rectangles for a typical image. Some amount of pruning can be done by sorting the rectangles based on mean gradient strength along its boundaries and removing ones that are not well aligned with the edges. We conservatively keep the top 700 such rectangles (the

average number of windows in our images is 15). By overestimating this number, the grouping algorithm is allowed to recover the best possible lattice without using hard thresholds early in the pipeline. Other image discretization methods such as interest points, correlation peaks [20], or color segmentation may also be used to generate v_i.

Given the set of tokens $v_i \in V$, we construct a pairwise MRF $G = (V, E)$. Each token is a random variable that constitutes a node of the undirected graph G, with edges $e_{ij} \in E$ representing the dependency between v_i and v_j. Since the probability of the states of a texton in an NRT is only locally dependent, the MRF model naturally preserves this Markov property. While [24] exploited it for tracking by preserving the structure over time, our goal is to build up the initial grid by linking together image tokens that exhibit the lattice topology.

The solution involves gradually evolving the lattice configuration by iteratively adding and removing edges within an MCMC framework. At the end of the process, links are created along vectors $\mathbf{t}_i^o : o \in \{r, l, u, d\}$ to the most likely right, left, up and down neighbors of v_i (Fig. 2) without violating grid constraints. Similarly, $v_i^o : o \in \{r, l, u, d, NULL\}$ denote its neighbors, if any, in each direction. Since we do not assume that texels are tightly packed and adjacent in the image, each node can potentially be linked to several others, increasing the combinatorics of the problem. Given image I, we wish to obtain the MAP estimate for the graph configuration

$$p(G|I, T, S) \propto p(I|T, S, G) \; p(S|G) p(T|G) \; p(G). \tag{1}$$

The image likelihood $p(I|T, S, G)$ is encapsulated in the rectangle detection and is neglected here. Color histograms, proximity of rectangle boundaries to image edges, or learned appearance models are all possible likelihood models. The shape prior $p(S|G)$ can be used to favor known shape models, though here we set it to unity since we are only dealing with rectangles. The graph prior $p(G)$ models any global intuition about the nature of the grid or the degree of connectedness. We set this to be unity.

The topology prior $P(T|G)$ is represented as a pairwise MRF whose joint can be factored into a product of local node potentials Φ and clique potentials Ψ:

$$P(T|G) \propto \prod_i \Phi(v_i) \prod_{i,j \in E} \Psi(v_i, v_j).$$

To model a grid, we measure the symmetry of direction vectors from a node to its neighbors. Let $\delta(\mathbf{t}_1, \mathbf{t}_2) = e^{-\beta||\mathbf{t}_1 - \mathbf{t}_2||}$ be a similarity measure between two neighbor vectors assuming both edges are in G. The potentials are now defined as:

$$\Phi(v_i) = e^{-\gamma(4-n_i)} * \delta(\mathbf{t}_i^r, -\mathbf{t}_i^l) * \delta(\mathbf{t}_i^u, -\mathbf{t}_i^d), \tag{2}$$

$$\Psi(v_i, v_j) = \delta(\mathbf{t}_i^u, \mathbf{t}_j^u) * \delta(\mathbf{t}_i^d, \mathbf{t}_j^d) * \mathcal{B}(v_i, v_j). \tag{3}$$

where n_i denotes the degree of node v_i. Thus we encourage increased connectivity as well as left/right and up/down edge pairs to be 180 degrees apart with similar
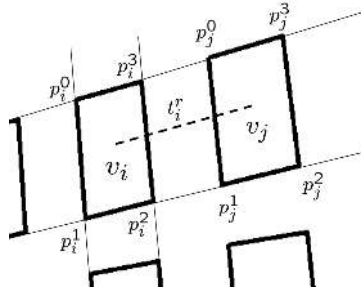
Fig. 3. Illustration of windows under perspective

magnitudes. The interaction potential between horizontal neighbors forces their vertical edges to be approximately parallel. For missing edges, a small fixed value of 0.2 is assigned to δ. These functions effectively model the generic lattice configuration as will be shown.

The function $\mathcal{B}(v_i, v_j)$ is used to specify any texture-specific pair-wise relationships between the texels. For building images and windows under perspective, we incorporate constraints such as overlap, cross ratio, and appearance similarity. Using Fig. 3 to illustrate, we list various heuristics that reflect the probability of v_i and v_j being connected by a horizontal edge. The case of vertical neighbors is analogous.

- Windows on the same level have their bottom and top edges aligned with each other, implying that points $(p_i^0, p_i^3, p_j^0, p_j^3)$ and $(p_i^1, p_i^2, p_j^1, p_j^2)$ are collinear.
- One projective invariant is the cross ratio $Cross$. Assuming parallel window sides and negligible noise, the 4 upper and lower points in Fig. 3 should have approximately the same cross ratio. We define $CR = \left| 1.0 - \frac{Cross(p_i^0, p_i^3, p_j^0, p_j^3)}{Cross(p_i^1, p_i^2, p_j^1, p_j^2)} \right|$ to quantify this measure. For horizontal neighbors, this essentially measures how parallel the vertical edges of the windows are.
- The horizontal dimensions of windows under perspective should vary smoothly on both the upper and lower sides.
- Windows and texels in general should not overlap.
- The corners of each polygon are correlated with each other to ensure appearance similarity. Correlating the entire window would be sensitive to occlusions and pose variations, while the corners are typically more distinctive. Formally, $XC(v_i, v_j) = \sum_{k=0}^{3} \frac{NCorr(Patch(p_i^k), Patch(p_j^k))}{4}$ where XC is the mean normalized cross correlation $NCorr$ of 11×11 patches centered at each of the 4 window corners.

These heuristics are converted into Gaussian likelihood functions that penalize deviations from our assumptions, and incorporated into $\mathcal{B}$. They are then used in conjunction with the generic lattice potentials defined in (2) and (3).
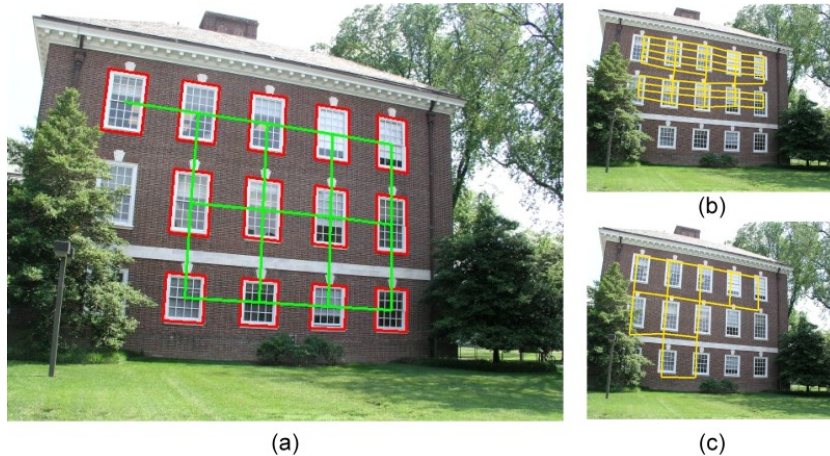
Fig. 4. (a) Result of our window grid discovery method; (b) reported best lattice from [20] which does not repeatably find the correct scale or centering while being less efficient; (c) best handpicked lattice from the various iterations of [20].

2.1 Optimization

We use a Markov Chain Monte Carlo (MCMC) framework to iteratively maximize (1), probabilistically adding and removing edges from the initial graph G_0 in a fashion similar to the multi-target tracking method of [27]. A Markov chain is defined over the space of configurations $\{G\}$ and the posterior is sampled using the Metropolis-Hastings [28] algorithm. A new state G'_t is accepted from state G_t with probability

$p = \min(1, \frac{p(G'_t|I,T,S)q(G_t|G'_t)}{p(G_t|I,T,S)q(G'_t|G_t)})$.

The graph G_0 is initialized by connecting each node with its lowest cost neighbor. The cost $E_{score}(i,j)$ for each pair of nodes v_i, v_j is measured as the total number of other nodes within a threshold distance of the line parametrized by the two nodes, scaled by rough shape similarity $\mathcal{B}$. Distracting elements or other deformations will cause inconsistencies in this Maximum Likelihood estimate; nevertheless, it provides a useful starting point for the MCMC simulation. Proposal updates $\mathcal{Q}(G'_t|G_t)$ for MCMC consist of edge additions or removals applied to a node v_k. Modifying edges one component at a time leads to better success rate for transitions. The transitions are made only in the up and right directions in order to keep the reverse transition probability simple. Two functions, picked probabilistically, govern how v_k is selected in each MCMC iteration: (i) an unguided scheme $\mathcal{Q}_1$ in which v_k is chosen uniformly from all nodes, and (ii) a guided hypothesis generation $\mathcal{Q}_2$ in which the edge is selected from a dynamic pool $\mathcal{P}_q$ of potentially good connections. As the grid converges to the correct solution, $\mathcal{Q}_2$ facilitates lattice growth and completion by hypothesizing edges close to the good parts of the lattice.
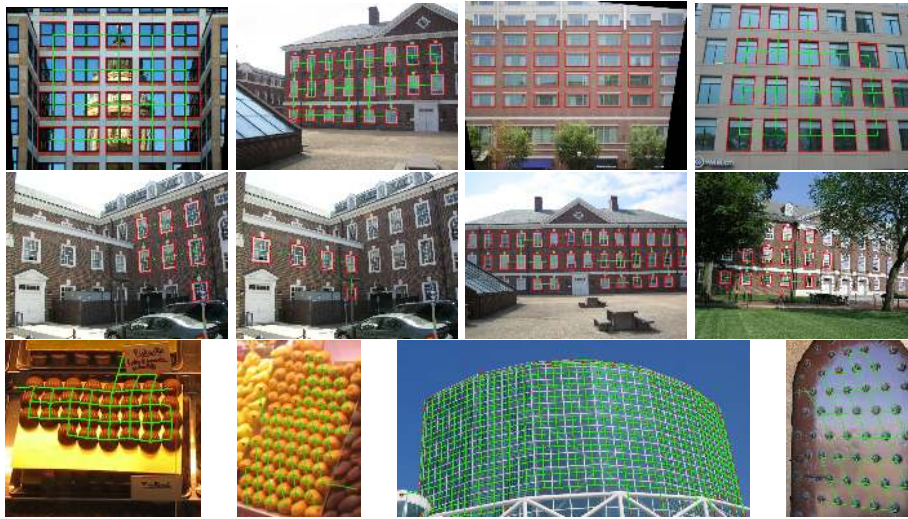
Fig. 5. Inferred lattice for various building images (top two rows). Bottom row shows results on a few images from the NRT database [29] by grouping purely on structure. Using a texture-specific $\mathcal{B}$ function could have prevented spurious links between dissimilar texels.

Let $e_{kl}^o : l \in \{1, \ldots, n_k\}$ be potential edges from v_k to its neighbors in direction o. In $\mathcal{Q}_1$, o is uniformly chosen from the up and right directions. The edge $\mathbf{t}_k^o$ from node v_k is turned off with fixed probability p_{off} or assigned a neighbor by sampling from $E_{score}(k, \cdot)$. Neighbors that seem to conform to the topological and visual priors are picked more often. Random selection alone can be inefficient in steering the optimization towards completing the lattice. When a node v_k is visited in $\mathcal{Q}_1$, its unbalanced edges, if any, are identified. A new link is hypothesized and added to priority queue $\mathcal{P}_q$ with a ranking function that encourages symmetry between opposite edges. A new proposal in $\mathcal{Q}_2$ simply involves removing the highest priority edge from $\mathcal{P}_q$ and adding it as an edge in state G_t'. The chain is *irreducible* because a series of edge additions and deletions can take the graph from one state to any other state. The stochastic elements also guarantee *aperiodicity* by not getting trapped in cycles. Together, they satisfy the MCMC conditions of ergodicity to ensure that the chain will converge to the stationary distribution.

2.2 Experimental Results

We show grouping results on building images as well as textures from the PSU NRT database [29]. The number of MCMC iterations was set to 10000 and the best MAP estimate chosen as the final configuration. A breadth-first traversal separates out the connected components. A user can then iterate over the

larger ones to pick a best lattice, or the selection can be done automatically. For building images, we use edge alignment to rank each lattice. Figures 4(a) and 5 demonstrate our results on several images. The building images (top two rows) are characterized by occlusions, shadows, reflections, and variation among windows; purely appearance-based systems can be sensitive to these effects. Both windows as well as its topological structure in the form of a neighborhood graph has been captured. By only enforcing local smoothness in appearance and geometry, the grouping is robust to small changes in window dimensions and perspective effects.

Figure 4 compares the result of our method with the algorithm of Hays [20]. One of the main disadvantages of [20] is efficiency. Generating the results for an image took approximately 30 minutes. In contrast, our method takes less than a minute in total on a 1.6GHz Pentium M laptop. Execution times for our method on the image of Fig. 4a are (i) rectangle hypotheses - 20.2 sec (in Matlab), (ii) initial graph construction - 15.9 sec (C++), and (iii) MCMC grouping - 1.9 sec (C++). Rectangle hypotheses can be speeded up by porting to C++ while the main bottleneck in the graph construction is due to exhaustive searching among nodes for nearest neighbors. The correlation peaks used by [20] also suffer from the lack of centering on windows or other semantically meaningful aspects of the image. Finally, after repeated iterations, the best lattice that it eventually outputs based on their metric is not perceptually the best and requires handpicking.

In order to test the adherence of our potential functions to the 4-connected definition of an NRT, we applied the grouping on pure NRTs after setting $\mathcal{B}$ to unity. Firstly, candidate tokens of like elements need to be extracted from the image. Similar to Hays, for images where MSER [30] gave a reasonably good initialization, these point features were grouped. If not, we applied the Matlab functions used by [20] to detect correlation peaks from a user-selected patch identifying a texel. This interaction only involves drawing a bounding box and eliminates some of the spurious lattices discovered from randomly selected patches. Even though the window detection is automatic, the interaction here is purely to evaluate the grouping and serves to separate out the token extraction from the overall framework. MCMC grouping is then performed by considering the dominant peaks (or MSER features) as tokens. Fig. 5 (bottom row) show the inferred lattice configuration for a few NRT images. Note that only the generic lattice functions in (2) and (3) have been used here. A suitable image-based $\mathcal{B}$ function could have prevented some of the incorrect links between unlike elements. On images undergoing significant non-rigid distortions, correlation may also not produce a strong enough peak, causing some interior texels to be suppressed. Being less relevant to the current theme on buildings, we do not explicitly address these issues in this work.
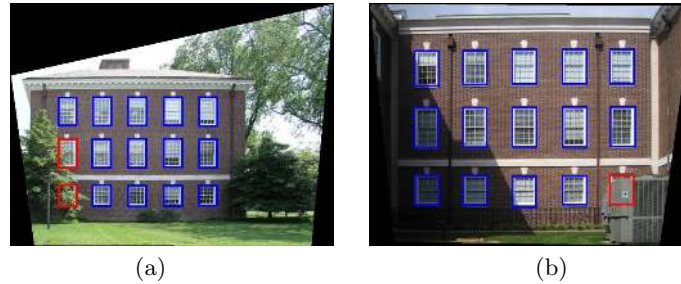
(a)   (b)

Fig. 6. Discovered grid shown as blue rectangles. Images were automatically rectified as a pre-processing step. Rectangles plotted in red are occluded or missing windows inferred from the result of grouping.

3   Facade Analysis and Manipulation

The window grid provides vital cues about positioning, layout, and scale. We now give a brief overview of some simple techniques for extracting additional properties . To ease processing, we work with rectified images in this section.

3.1   Lattice Completion

Occlusions or errors in the rectangle hypotheses can cause missing nodes in the lattice. The already grouped elements facilitate parameter inference of the regular grid. The median height, width, and magnitudes of the horizontal and vertical **t** vectors are computed first. We then pick the node with the highest likelihood according to (1) as an origin. This completely specifies a regular grid that can be overlaid on the image to hypothesize missing or occluded lattice elements. When window dimensions deviate from the perfect grid assumption, we observed that windows on the same floor are similar and centered horizontally and vertically with its neighbors. After overlaying the regular grid on the image, each location is tested for detected windows. If the test fails, a missing grid element is inferred at the location aligned with its neighbors in the two principal directions. Figure 6 shows examples where the discovered grid (drawn in blue) is used to identify occluded grid elements (drawn in red).

3.2   Facade Segmentation

By making assumptions founded on common architectural trends, building pixels can be segmented out from a static image without any prior knowledge of appearance models. We first assume that the pixels immediately around the perimeter of a window belong to the building wall and exhibit a Gaussian distribution W. Although not valid for walls with multiple colors or shadows, many buildings do exhibit such uniform color properties. Similar to MRF-based segmentation [31], we assume that a color Gaussian Mixture Model (GMM) can
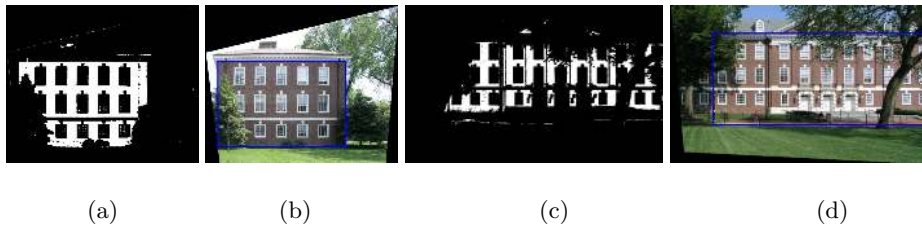
Fig. 7. (a) and (c): Mask of the wall pixels after maximum likelihood classification; (b) and (d): thresholding on the row and column sum differences between adjacent locations can be used to approximate the facade extent (blue rectangles).

describe the majority of pixels in the image. The RGB values are clustered into N (typically less than 10) distributions $\mathcal{G}_i : i \in 1..N$. Based on the homogeneous texture assumption, the mean and covariance of the distributions can be used to compute the Maximum Likelihood cluster $\mathcal{G}_w$ that corresponds best to the wall color W. The left column of Figure 7 shows the mask separating out the complicated foreground for a couple of images. We can also use the mask to compute approximate facade boundaries (right column) by thresholding on adjacent row and column sum differences.

3.3 Foreground Removal

Texture discovery gives a set of subimages centered on the tile elements that should be very similar to one another. Appearance discrepancies arising from material variations, spatial resolution, and non-planar features can be described with low-dimensional Gaussian models or blurring and shifting of patches. However, foreground elements or reflections are best treated as outliers of the building tile pixel mode. To first detect foreground, we look at pixel values in corresponding locations over all tiles under the assumption that the background is visible in a majority of them. A robust measure of spread, the median absolute deviation (MAD) [32], can be used to assess which pixel values vary enough across the tiles to arouse suspicion of foreground. [3] Unreliable pixels are identified by thresholding their MADs—these are so-called *MAD outlier* pixels.

An obvious approach to background reconstruction is to do spatial inpainting on pixels marked out according to the MAD criterion. Figure 1(c) shows the result of inpainting MAD-induced holes in Figure 1(a) with the method of [9] under the special case that the inpainting source patches are the same sizes as and perfectly aligned with the discovered tiles. In spite of this, the more complex the tile interior is, the less effective inpainting would be in avoiding geometric and photometric artifacts.

---

[3] A scalar MAD value is obtained at each pixel by computing it separately for each color channel and summing
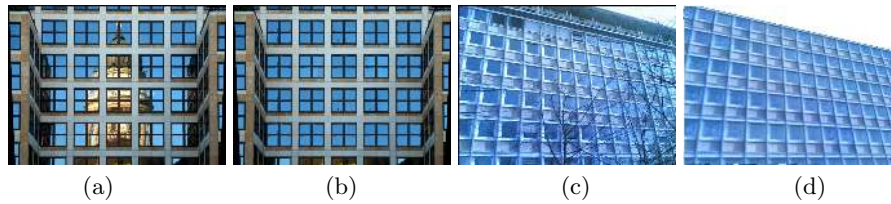
Fig. 8. Foreground removal by eigenimage reconstruction for tile aligned images (a) and (c).

Another possibility suggested by the alignment of patches is to treat the problem as one of eigenimage reconstruction. Assuming that a Gaussian process describes inter-tile appearance variation fairly well, we can use principal components analysis (PCA) to model it (e.g., [33, 18]). The intuition is that we want to take each occluded tile in which some background is visible and "project" it down onto a set of background-only bases in order to lessen the foreground influence. However, with some fraction of the tiles "polluted" by unknown foreground elements, robust PCA (RPCA) techniques [34, 32] are required. In practice, these methods had problems with our data when there were too many outlier pixels in a tile. Sine the MAD mask identifies the outlier pixels well, we use another PCA variant called probabilistic PCA (PPCA) [35] which works when missing data is explicitly identified beforehand. As the percentage of pixels occluded in a given tile rises, however, the reliability of the reconstruction naturally deteriorates. We mitigate this issue by reconstructing only tiles that have $\leq 25\%$ outliers in them. All other tiles are treated as fully occluded and simply sampled *de novo* from the learned PPCA basis. Figures 1(d) and 8 show results of using PPCA on images containing complicated foreground or reflections within windows.

4   Conclusion

We draw the analogy that building facades are often examples of Near-Regular Textures, and showed that discovering these textures could provide valuable insight into the rest of the facade. We introduce a novel MRF/MCMC approach to discover grid patterns from images. We then presented techniques that use a partially discovered grid to infer occluded windows, segment out wall texture, identify foreground pixels, and reconstruct the background – all from a single image.

All the components described in the paper have much scope for extension. Foremost among these, an extensive evaluation of our lattice discovery technique to more general NRTs is being done. An image-specific likelihood function is required to prevent false links that might be topologically correct. One shortcoming of our proposal function is the inability to hypothesize a new token during MCMC, which would require Reversible Jump MCMC. We have also had encouraging results in parsing window interiors to describe them with split

grammars. The PCA approach to recover the background currently disallows tile non-regularity such as those in Fig. 7. However, the segmentation mask and window grid provide enough information to allow inpainting of foreground pixels and copying of whole windows to maintain coherency. It is also important to note that each of the techniques presented in Section 3 can be replaced with more sophisticated methods. Alternative approaches such as gradient-domain methods [6] could be used for texture replacement as long as the facade structure discovered by the grouping framework is not violated during synthesis. Future work includes comparing some of these methods based on the quality of the recovered texture map.

References

1. Debevec, P., Taylor, C., Malik, J.: Modeling and rendering architecture from photographs. In: SIGGRAPH. (1996)
2. Teller, S., Antone, M., Bodnar, Z., Bosse, M., Coorg, S., Jethwa, M., Master, N.: Calibrated, registered images of an extended urban area. Int. J. Computer Vision **53** (2003) 93–107
3. Dick, A., Torr, P., Cipolla, R.: Modelling and interpretation of architecture from several images. Int. J. Computer Vision **60**(2) (November 2004)
4. van den Heuvel, F.: Automation in Architectural Photogrammetry; Line-Photogrammetry for the Reconstruction from Single and Multiple Images. PhD thesis, Delft University of Technology, Delft, The Netherlands (2003)
5. Früh, C., Zakhor, A.: An automated method for large-scale, ground-based city model acquisition. Int. J. Comput. Vision **60**(1) (2004) 5–24
6. Wilczkowiak, M., Brostow, G., Tordoff, B., Cipolla, R.: Hole filling through photomontage. In: Proc. British Machine Vision Conference. (2005)
7. Korah, T., Rasmussen, C.: Spatiotemporal inpainting for recovering texture maps of occluded building facades. IEEE Transactions on Image Processing **16**(9) (2007) 2262–2271
8. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: SIGGRAPH. (2000) 417–424
9. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Processing **13**(9) (2004)
10. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. ACM Transactions on Graphics **24** (2005) 861–868
11. Mayer, H., Reznik, S.: Building faade interpretation from image sequences. In: Proc. of the ISPRS Workshop CMRT 2005 - Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation. (2005)
12. Neal, R.M.: Probabilistic inference using markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto (1993)
13. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. ACM Transactions on Graphics **22** (2003) 669–677
14. Alegre, F., Dellaert, F.: A probabilistic approach to the semantic interpretation of building facades. In: International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres. (2004)

15. Mueller, P., Zeng, G., Wonka, P., Gool, L.V.: Image-based procedural modeling of facades. In: Proceedings of ACM SIGGRAPH 2007, New York, NY, USA, ACM Press (2007)
16. Schaffalitzky, F., Zisserman, A.: Geometric grouping of repeated elements within images. In: Proceedings of the 9th British Machine Vision Conference, Southampton. (1998)
17. Turina, A., Tuytelaars, T., Gool, L.V.: Efficient grouping under perspective skew. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (2001)
18. Liu, Y., Lin, W.C., Hays, J.H.: Near regular texture analysis and manipulation. ACM Transactions on Graphics (SIGGRAPH 2004) **23**(3) (August 2004) 368 – 376
19. Liu, Y., Collins, R., Tsin, Y.: A computational model for periodic pattern perception based on frieze and wallpaper groups. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(3) (March 2004) 354 – 371
20. Hays, J.H., Leordeanu, M., Efros, A.A., Liu, Y.: Discovering texture regularity as a higher-order correspondence problem. In: 9th European Conference on Computer Vision. (May 2006)
21. Ahuja, N., Todorovic, S.: Extracting texels in 2.1d natural textures. In: Proc. IEEE Int. Conf. Computer Vision (ICCV07). (2007)
22. Li, S.Z.: Markov random field modeling in computer vision. Springer-Verlag (1995)
23. Hartelius, K., Carstensen, J.M.: Bayesian grid matching. IEEE Trans. Pattern Anal. Mach. Intell. **25**(2) (2003) 162–173
24. Lin, W.C., Liu, Y.: Tracking dynamic near-regular textures under occlusion and rapid movements. In: 9th European Conference on Computer Vision. (2006)
25. Matsuyama, T., Miura, S., Nagao, M.: A structural analysis of natural textures by fourier transformation. In: Proc. Int. Conf. Pattern Recognition. (1982)
26. Han, F., Zhu, S.C.: Bottom-up/top-down image parsing by attribute graph grammar. In: Proc. of the IEEE International Conference on Computer Vision (ICCV05). (2005)
27. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. Pattern Analysis and Machine Intelligence **27**(11) (November 2005) 1805–1918
28. Hastings, W.: Monte carlo sampling methods using markov chains and their applications. Biometrika **57(1)** (1970) 97–109
29. Lee, S., Liu, Y.: Psu Near-Regular Texture Database. Available at `http://vivid.cse.psu.edu/texturedb/gallery/`. (2007)
30. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the British Machine Vision Conference. (2002)
31. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004)
32. la Torre, F.D., Black, M.: A framework for robust subspace learning. Int. J. Computer Vision **54 (Aug 2003)** (2003) 117–142
33. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition. (1991)
34. Xu, L., Yuille, A.: Robust principal component analysis by self-organizing rules based on statistical physics approach. IEEE Transactions on Neural Networks **6**(1) (1995) 131 – 143
35. Roweis, S.: EM algorithms for PCA and SPCA. In: Advances in Neural Information Processing Systems. (1997)