Automatic Kronecker Product Model Based Detection of Repeated Patterns in 2D Urban Images[*]

Juan Liu
Graduate Center, CUNY
New York, NY
langelfly@hotmail.com

Emmanouil Psarakis
University of Patras
Patras, Greece
psarakis@ceid.upatras.gr

Ioannis Stamos
Hunter Coll. & Grad. Center, CUNY
New York, NY
istamos@hunter.cuny.edu

Abstract

Repeated patterns (such as windows, tiles, balconies and doors) are prominent and significant features in urban scenes. Therefore, detection of these repeated patterns becomes very important for city scene analysis. This paper attacks the problem of repeated patterns detection in a precise, efficient and automatic way, by combining traditional feature extraction followed by a Kronecker product low-rank modeling approach. Our method is tailored for 2D images of building façades. We have developed algorithms for automatic selection of a representative texture within façade images using vanishing points and Harris corners. After rectifying the input images, we describe novel algorithms that extract repeated patterns by using Kronecker product based modeling that is based on a solid theoretical foundation. Our approach is unique and has not ever been used for façade analysis. We have tested our algorithms in a large set of images.

1. Introduction

Urban scenes contain rich periodic or near-periodic structures, such as windows, doors, and other architectural features. Detection of the periodic structures is useful in many applications such as photorealistic 3D reconstruction, 2D-to-3D alignment, façade parsing, city modeling, classification, navigation, visualization in 3D map environments, shape completion, cinematography and 3D games, just to name a few. However it is a challenging task due to scene occlusion, varying illumination, pose variation and sensor noise.

The first step of all façade parsing algorithms (see [15] for an example) is the detection and rectification of individual façade structures. The first part of our pipeline (Sec. 3) adopts vanishing points detection to compute an initial transform matrix for rectification and for the automatic selection of a representative texture. The second part of our pipeline (Secs. 4,5), after rectification, is the detection of repeated patterns. Current state-of-the-art methods use classification [15] or statistical approaches [13]. We, on the other hand, provide a novel detection method that models repetition as a Kronecker product.

2. Related Work

In recent years, repeated patterns or periodic structures detection has received significant attention in both 2D images [17, 13] and 3D point clouds [3, 12]. Repeated patterns are usually hypothesized from the matching of local image features. They can be modeled as a set of sparse repeated features [11] in which the crystallographic group theory [8] was employed. The work of [14] maximizes local symmetries and separates different repetition groups via evaluation of the local repetition quality conditionally for different repetition intervals.

The work of [10] proposes an approach to detect symmetric structures in a rectified fronto-façade and to reconstruct a 3D geometric model. The work of [15] describes a method for periodic structure detection upon the pixel-classification results of a rectified façade. Shape grammars have also been used for 2D façade parsing [13]. Other similar grammar-based approaches include [1].

All the above-mentioned methods require as preprocessing image rectification. To solve this problem, low-rank methods were used and attracted a lot of attention in recent years [16]. A similar work was proposed by [4] in which the rank value N is assumed known. Another method for the recovery of both low-rank and the sparse components is presented in [2]. Finally, [7] describes a low-rank based method that detects the repeated patterns in 2D images for the application of shape completion.

The contributions of this paper with respect to earlier work are: (a) an automated method based on vanishing points for detection of representative texture within a façade, and (b) a novel algorithm based on Kronecker product for detection of repeated patterns within that texture of

[*] This work has been supported in part by NSF grants IIS-0915971 and CCF-0916452. We are grateful to the anonymous reviewers.

the façade image.

3. Texture Selection and Rectification

The input of this step is a 2D image of a building façade. The output is a representative texture on the façade as well as a transform matrix that is used to initialize the facade rectification. This representative texture is essential for an automated system, since it is used as input by the low-rank algorithm (named TILT) in [16] (this automation provides a performance improvement of 19.6% over manual selection; see the comparison in Table 1). This algorithm is implemented in three steps: (1) feature extraction, (2) block division, and (3) transform initialization and representative texture selection.

First of all, we extract Harris corners [5]. We also detect the two major vanishing points by using the method of [6]. We then divide the façade into blocks (quadrilateral) along vanishing points directions. Finally, we compute the homography matrix that rectifies the image and select the representative texture by combining Harris corners distribution information within the detected blocks. We observed that Harris corners are distributed almost uniformly in unoccluded façade areas that contain repeated patterns. Otherwise occlusions may produce a non-uniform distribution of the Harris corners. For example, the Harris corners in a tree area will be very dense and non-uniformly spaced.

In particular, starting from each detected vanishing point we draw hypothetical lines at angular intervals towards the image assuring that all Harris corners are included in the generated quadrilaterals (see Fig. 1(a)). This is achieved by computing the smallest angles θ_1 and θ_2 (one for each vanishing point) that ensure inclusion of all Harris corners. Then, we divide each range θ_i into m parts. The intersections of the imaginary lines thus create $m \times m$ quadrilaterals.

We observed that the number of corners in each block does not change much after perspective distortion of the façade image, although the distribution of the Harris corners depends on the location of the vanishing points. Excluding strong perspective distortions is not so crucial (in such cases even robust techniques fail to rectify the image). We thus assume that the ideal texture should consist of neighboring blocks that have a similar and uniform distribution of Harris corners. We then count the number of Harris corners in each block, and get an $m \times m$ matrix $\mathbf{C}$, where each element $\mathbf{C}_{i,\, j}, i,\; j = 1, \cdots, m$ is the number of Harris corners in the corresponding block.

In order to isolate the $r \times c$ submatrix of $\mathbf{C}$[1] containing the most representative texture of the given façade, let us consider that its elements are random samples from a double exponential distribution. We then compute the sample median $\mu_{\mathbf{C}}$ of the elements of matrix $\mathbf{C}$. Finally, we slide a window of size $r \times c$ along matrix $\mathbf{C}$, and compute in each location the sample mean deviation from the sample median, that is:

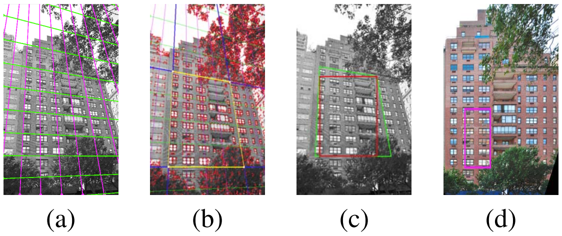


Figure 1. (a) 10 by 10 blocks divided along detected vanishing points directions. (b) Yellow quadrilateral shows $r \times c$ blocks that compose the representative texture. (c) Red box, defined by the largest rectangle within the representative texture, is the input to the TILT algorithm. In green is the yellow quadrilateral from (c). (d) Rectified façade by TILT, with automatically selected texture shown inside the pink box.

$$\mathbf{S}_{i,j} = \frac{1}{rc} \sum_{k=i}^{i+r-1} \sum_{l=j}^{j+c-1} |\mathbf{C}_{k,\, l} - \mu_{\mathbf{C}}| \tag{1}$$

thus forming a score matrix of size $(m{-}r{+}1) \times (m{-}c{+}1)$. It is well known that the sample median and the sample mean deviation from the sample median are the maximum likelihood estimators of the mean and standard deviation of the distribution. Thus, by choosing the sliding window with the highest score we actually choose the one with the minimum variance among all the best likelihood estimators of the mean value. This window will be selected and used as the input of the TILT algorithm in order to get the low-rank component and rectification of the façade image (see Fig. 1 for an example). Note that this low-rank component is not used by our algorithms, since we have a novel way of calculating it (see Sec. 4).

4. Façade Modeling via Kronecker Products

In this section we describe our Kronecker product modeling approach that is applied on a rectified façade image. It is a novel representation that describes a large subset of façade examples.

4.1. Ideal Façade Modeling

To this end, let us consider the partition of all ones orthogonal array $\mathbf{1}_{l_\mathbf{v} \times l_\mathbf{h}}$ of size $l_\mathbf{v} \times l_\mathbf{h}$ by using the following mutually exclusive, $1-0$ [2] matrices $\mathbf{M}_k,\ k = 1,\ 2, \cdots,\ K$ of size $l_\mathbf{v} \times l_\mathbf{h}$ each, that is:

$$< vec\{\mathbf{M}_k\},\ vec\{\mathbf{M}_l\} > = \begin{cases} ||vec\{\mathbf{M}_k\}||_0, & k = l \\ 0, & k \neq l \end{cases} \tag{2}$$

$$\sum_{k=1}^{K} \mathbf{M}_k \;=\; \mathbf{1}_{l_\mathbf{v} \times l_\mathbf{h}} \tag{3}$$

[1] In all of our experiments, we set m to 10 and fix r and c to a given percentage of m, that is $r = c = 0.4m$.

[2] Matrices that contain only combinations of 1s and 0s

where $vec\{\mathbf{X}\}$, $<\mathbf{x},\ \mathbf{y}>$ and $||\mathbf{x}||_0$ denote the *column-wise* vectorization of matrix $\mathbf{X}$, the inner product of vectors $\mathbf{x},\ \mathbf{y}$ and the l_0 *norm* of vector $\mathbf{x}$ respectively. As it is clear from Eqs. (2-3), different choices of matrices $\mathbf{M}_k$ result in different partitions of orthogonal block $\mathbf{1}_{l_{\mathrm{v}}\times l_{\mathrm{h}}}$.

Let us now associate with each component $\mathbf{M}_k,\ k = 1,\ 2,\ \cdots,\ K$ of the partition of array $\mathbf{1}_{l_{\mathrm{v}}\times l_{\mathrm{h}}}$ defined in Eq. (3), a 2-D pattern $\mathbf{P}_k$ of size $N_{\mathbf{v}}\times N_{\mathbf{h}}$ that is going to be repeated according to $\mathbf{M}_k$. The patterns should have a piecewise constant surface form. In particular, with the aim of patterns $\mathbf{P}_k$ several windows, doors and/or balconies of different architectures can be formed.

We can now define a subset of urban building façades that can be expressed as a sum of Kronecker products:

$$\mathcal{F}_{N\times M}=\sum_{k=1}^{K}\lambda_k(\mathbf{M}_k\otimes\mathbf{P}_k) \tag{4}$$

where $\mathbf{X}\otimes\mathbf{Y}$ is the *Kronecker* product of matrices $\mathbf{X},\ \mathbf{Y}$ and $\lambda_k,\ k_1=1,2,\cdots,K$ are weights. Finally, $N\times M$ is the size of the urban building façade image. By the definition of the Kronecker product it is obvious that $N=l_{\mathbf{v}}N_{\mathbf{v}}$ and $M=l_{\mathbf{h}}N_{\mathbf{h}}$. Please note that the urban building façade's model defined in Eq. (4) can be used even in cases where there is not any periodic structure in the given input façade we would like to model.

Generalizing Eq. (4) to permit a "wall" gray level λ_0, we get:

$$\mathcal{F}_{N\times M}=\lambda_0\mathbf{1}_N\mathbf{1}_M^t+\sum_{k=1}^{K}\lambda_k(\mathbf{M}_k\otimes\mathbf{P}_k). \tag{5}$$

Using the fact that the components of the partition of orthogonal array $\mathbf{1}_{l_{\mathrm{v}}\times l_{\mathrm{h}}}$ of Eq. (3) are mutually exclusive, we rewrite Eq. (5) as:

$$\mathcal{F}_{N\times M}=\sum_{k=1}^{K}\lambda_k(\mathbf{M}_k\otimes\hat{\mathbf{P}}_k),\hat{\mathbf{P}}_k=\mathbf{P}_k+\frac{\lambda_0}{\lambda_k}\mathbf{1}_{N_{\mathrm{v}}}\mathbf{1}^t_{N_{\mathrm{h}}} \tag{6}$$

where $\hat{\mathbf{P}}_k$ are modified patterns as defined above, and $\mathbf{1}_{N_{\mathrm{v}}},\ \mathbf{1}_{N_{\mathrm{h}}}$ are all ones vectors with the subscripts denoting their lengths.

4.2. Ideal Façade Model Approximation

In this section we would like to compute (or approximate) the components of the Kronecker product that generate a given ideal (i.e. noise-free) building façade $\mathbf{F}_{N\times M}\in\mathbb{R}^{N\times M}$ with $N=l_{\mathbf{v}}N_{\mathbf{v}}$ and $M=l_{\mathbf{h}}N_{\mathbf{h}}$. Using the model defined in Eq. (6) we can define the following cost function:

$$\mathcal{C}_{\mathbf{F}}(\mathbf{M}_k,\hat{\mathbf{P}}_k,\ \lambda_k,\ k=1,\cdots,K)=||\mathbf{F}_{N\times M}-\mathcal{F}_{N\times M}||_2^2$$
$$=\ ||\mathbf{F}_{N\times M}-\sum_{k=1}^{K}\lambda_k(\mathbf{M}_k\otimes\hat{\mathbf{P}}_k)||_2^2, \tag{7}$$

where $\mathbf{M}_k,\ \hat{\mathbf{P}}_k$ and $\lambda_k,\ k=1,2,\cdots,K$ denote the partition matrices, the patterns and the weighting factors of façade's model respectively. As it is clear from its definition $\mathcal{C}_{\mathbf{F}}(.)$ is a Frobenious norm based cost function that quantifies the error between the given matrix $\mathbf{F}_{N\times M}$ and the model $\mathcal{F}_{N\times M}$.

Therefore, the modeling problem of the given urban building façade $\mathbf{F}_{N\times M}$ can be expressed by the following minimization problem

$$\min_{\mathbf{M}_k,\hat{\mathbf{P}}_k,\lambda_k,\ k=1,\cdots,K}\ \mathcal{C}_{\mathcal{F}}(\mathbf{M}_k,\ \hat{\mathbf{P}}_k,\ \lambda_k,\ k=1,\cdots,K), \tag{8}$$

which is known as the Nearest Kronecker Product problem [9]. The following partition of the given matrix $\mathbf{F}_{N\times M}$ is key for the solution of the above problem:

$$\mathbf{F}_{N\times M}=\begin{bmatrix}\mathbf{F}_{11} & \mathbf{F}_{12} & \cdots & \mathbf{F}_{1l_{\mathrm{h}}}\\ \mathbf{F}_{21} & \mathbf{F}_{22} & \cdots & \mathbf{F}_{2l_{\mathrm{h}}}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{F}_{l_{\mathrm{v}}1} & \mathbf{F}_{l_{\mathrm{v}}2} & \cdots & \mathbf{F}_{l_{\mathrm{v}}l_{\mathrm{h}}}\end{bmatrix}, \tag{9}$$

where $\mathbf{F}_{ij}$ is a block of size $N_{\mathbf{v}}\times N_{\mathbf{h}}$. We can then form the matrix

$$\tilde{\mathbf{F}}_{l_{\mathrm{v}}l_{\mathrm{h}}\times N_{\mathrm{v}}N_{\mathrm{h}}}=\left[\ vec\{\mathbf{F}_{11}\}\ vec\{\mathbf{F}_{21}\}\dots vec\{\mathbf{F}_{l_{\mathrm{v}}l_{\mathrm{h}}}\}\ \right]^T \tag{10}$$

which constitutes a rearrangement of the given façade matrix $\mathbf{F}_{N\times M}$. Using the above defined quantities, the cost function of Eq. (7) can be equivalently expressed as:

$$\mathcal{C}_{\mathbf{F}}(\mathbf{m}_k,\ \hat{\mathbf{p}}_k,\lambda_k,\ k=1,\cdots,K)$$
$$=||\tilde{\mathbf{F}}_{l_{\mathrm{v}}l_{\mathrm{h}}\times N_{\mathrm{v}}N_{\mathrm{h}}}-\sum_{k=1}^{K}\lambda_k\mathbf{m}_k\hat{\mathbf{p}}_k^t||_2^2 \tag{11}$$

where $\mathbf{m}_k,\ \hat{\mathbf{p}}_k$ are the column-wise vectorized forms of matrices $\mathbf{M}_k,\ \hat{\mathbf{P}}_k$. By exploiting the above defined equivalent form of the cost function, the $Kronecker\ Product\ SVD$ [9] can be used to solve the optimization problem of Eq. (8):

Theorem 1: Let $\tilde{\mathbf{F}}_{l_{\mathrm{v}}l_{\mathrm{h}}\times N_{\mathrm{v}}N_{\mathrm{h}}}=\mathbf{V\Sigma U}^T$be the Singular Value Decomposition of the rearranged counterpart of matrix $\mathbf{F}_{N\times M}$. Let us also consider the following diagonal matrix

$$\mathbf{\Sigma}_K=diag\left\{\sigma_1\ \sigma_2\ \cdots\ \sigma_K\right\} \tag{12}$$

containing the first K singular values of matrix $\tilde{\mathbf{F}}_{l_{\mathrm{v}}l_{\mathrm{h}}\times N_{\mathrm{v}}N_{\mathrm{h}}}$, and let

$$\mathbf{V}_K=[\mathbf{v}_1\ \mathbf{v}_2\ \cdots\mathbf{v}_K],\quad \mathbf{U}_K=[\mathbf{u}_1\ \mathbf{u}_2\ \cdots\mathbf{u}_K] \tag{13}$$

be the K associated left and right singular vectors respectively. Then, the matrices $\mathbf{M}_k^\star$, the patterns $\hat{\mathbf{P}}_k^\star$, and the weighting factors $\lambda_k^\star$ that satisfy:

$$vec\{\mathbf{M}_k^\star\}=\mathbf{v}_k, vec\{\hat{\mathbf{P}}_k^\star\}=\mathbf{u}_k,\lambda_k^\star=\sigma_k,\ k=1,2,\cdots,K \tag{14}$$

constitute the optimal solution of the optimization problem of Eq. (8).

Using Theorem 1, we can find an optimal approximation that has the desired form, i.e. it is a sum of Kronecker products, that minimizes the cost function defined in Eq. (7). Note, however, that some of the characteristics of the optimal solution, are not consistent with the ingredients of the façade model defined in (5) thus making the direct use of Theorem 1 problematic. Specifically, neither the optimal matrices $\mathbf{M}_k^\star$ neither the optimal patterns $\hat{\mathbf{P}}_k^\star$ have, in the general case, the desired form, that is they are not 1-0 matrices and piecewise constant surfaces, respectively. In addition, the vectorized form of the optimal patterns are orthonormal to each other.

In order to impose one of the requirements of the proposed façade model, in the sequel we consider that matrices $\mathbf{M}_k$ have the desired $1-0$ form and are known. In such a case, we form the cost function:

$$\hat{\mathcal{C}}_{\mathbf{F}}(\hat{\mathbf{P}}_k,\ \lambda_k,\ k=1,\cdots,K|\mathbf{M}_k), \qquad (15)$$

which is the cost function of Eq. (8) but with the partition matrices known. We would like to minimize it with respect to the patterns $\hat{\mathbf{P}}_k$ and the weighting factors λ_k. The solution of the new optimization problem is the subject of the next lemma.

Lemma 1: Assuming that the matrices $\mathbf{M}_k,\ k=1,2,\cdots,K$ defined in Eqs. (2-3) are known, then the minimization of the cost function defined in Eq. (15) produces patterns $\hat{\mathbf{P}}_k$ and weighting factors λ_k that are related as follows:

$$\lambda_k^\star vec\{\hat{\mathbf{P}}_k^\star\}=\frac{\mathbf{U}\boldsymbol{\Sigma}^T\mathbf{V}^T vec\{\mathbf{M}_k\}}{||vec\{\mathbf{M}_k\}||_2^2}, k=1,2,\cdots,K(16)$$

Proof: Using the fact that $||\mathbf{F}||_2^2 = \text{trace}\{\mathbf{F}^T\mathbf{F}\}$, the SVD decomposition of the rearranged counterpart of matrix $\mathbf{F}_{N\times M}$, the linearity of the *trace* operator, and after some simple mathematical manipulations, the cost function defined in Eq. (11) can be rewritten as follows:

$$\mathcal{C}_{\mathbf{F}}(\hat{\mathbf{P}}_k,\ \lambda_k,\ k=1,\cdots,K|\mathbf{M}_k) = trace\left\{\mathbf{U}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{U}^T\right\} - trace\left\{\mathbf{U}\boldsymbol{\Sigma}^T\mathbf{V}^T\sum_{k=1}^{K}\lambda_k\mathbf{m}_k\hat{\mathbf{p}}_k^t+\sum_{k=1}^{K}\lambda_k\hat{\mathbf{p}}_k\mathbf{m}_k^t\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\right\} + trace\left\{\left(\sum_{k=1}^{K}\lambda_k\mathbf{m}_k\hat{\mathbf{p}}_k^t\right)\left(\sum_{k=1}^{K}\lambda_k\hat{\mathbf{p}}_k\mathbf{m}_k^t\right)\right\}.$$

Moreover, using the orthogonality of vectors $\mathbf{m}_k, k=1,2,\cdots,K$, the orthonormality of matrix $\mathbf{U}$, the commutative property of *trace* operator, and by interchanging the order of summations and *trace* operator, we obtain:

$$\mathcal{C}_{\mathbf{F}}(\hat{\mathbf{P}}_k,\ \lambda_k,\ k=1,\cdots,K\mid\mathbf{M}_k) = trace\left\{\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\right\} - 2\sum_{k=1}^{K}\lambda_k trace\left\{\hat{\mathbf{p}}_k^t\mathbf{U}\boldsymbol{\Sigma}^T\mathbf{V}^T\mathbf{m}_k\right\} + \sum_{k=1}^{K}\lambda_k^2||\mathbf{m}_k||_2^2||\hat{\mathbf{p}}_k||_2^2.$$

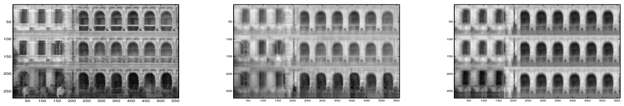
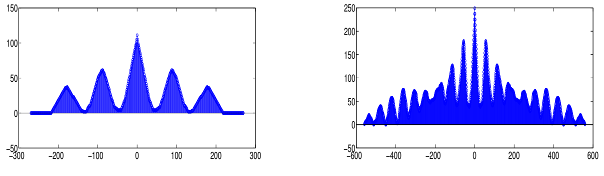


Figure 2. A real building façade (left), its optimal modeling of rank 4 (middle), obtained from the solution of the optimization problem of Eq. (8), and its optimal modeling of rank 4 (right), obtained from the minimization of the cost function of Eq. (15) with matrices $\mathbf{M}_k$ predefined (please see text).



(a)

(b)

Figure 3. Estimation of the spatial periods of façade shown in Fig. 2 (left). Cross-Correlation sequences used for the estimation of $N_{\mathbf{v}} = 90$ *pixels* (a) and estimation of $N_{\mathbf{h}} = 56$ *pixels* (b). Please enlarge the image to see the coordinates. Distance between the adjacent peaks provides the period information.

By taking the partial derivatives of the above function with respect to all components of the parameters $\hat{\mathbf{p}}_k,\ k=1,2,\cdots,K$, stacking and setting them to zero we obtain the desired result. ■

Note that if we substitute into (16) the optimal solution of Eq. (14) for $\mathbf{M}_k$, the optimal solution of the patterns as well as the weighting factors coincide with those in Eq. (14) as they owed to be. Note also that according to Eq. (16), the vectorized forms of the optimal patterns are not necessarily orthonormal to each other, unlike Theorem 1.

We applied both of the above optimal solutions for the modeling of urban building shown in Fig. 2(left) and the resulting rank 4 solutions are shown in Figs. 2(middle) and 2(right). The matrices $\mathbf{M}_k,\ k=1,2,3,4$ we used for the evaluation of the optimal solution of Eq. (16) are:

$$\mathbf{M}_1=\begin{bmatrix}\mathbf{1}_{2\times3} & \mathbf{0}_{2\times7}\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times7}\end{bmatrix} \qquad \mathbf{M}_2=\begin{bmatrix}\mathbf{0}_{2\times3} & \mathbf{0}_{2\times7}\\ \mathbf{1}_{1\times3} & \mathbf{0}_{1\times7}\end{bmatrix}$$
$$\mathbf{M}_3=\begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{1}_{3\times1} & \mathbf{0}_{3\times6}\end{bmatrix} \quad \mathbf{M}_4=\begin{bmatrix}\mathbf{0}_{3\times4} & \mathbf{1}_{3\times6}\end{bmatrix}.$$

It is evident from Figs. 2(middle) and 2(right) that the optimal solution resulting from the application of Eq. (16) outperforms the former one as expected.

Lemma 1 is a powerful tool that can be used for solving the modeling problem of urban building façades. However, its use demands knowledge on the partitioning $1-0$ matrices $\mathbf{M}_k,\ k=1,\ldots,K$. In the next section, inspired by Eq. (16), we present a clustering based technique to estimate them and finally solve the Kronecker approximation problem.

5. Algorithm

Most of the well known low-rank modeling techniques, are using the original image and try to minimize its rank.

We, on the other hand, use a Kronecker product based model $\mathcal{F}_{N\times M}$. We are thus able to express the cost function defined in Eq. (7) in an equivalent form (11). This is essential, since by transforming the given matrix $\mathbf{F}_{N\times M}$ into its rearranged counterpart $\tilde{\mathbf{F}}_{l_\mathbf{v}l_\mathbf{h}\times N_\mathbf{v}N_\mathbf{h}}$, we form a matrix whose rank is drastically reduced (it is upper bounded by the smallest dimension of the above mentioned matrix, which usually is equal to $l_\mathbf{v}l_\mathbf{h}$). Our algorithm starts with the estimation of the size $N_\mathbf{v}\times N_\mathbf{h}$ of the patterns (Sec. 5.4), continues with the estimation of K and the actual partition matrices (Secs. 5.1-5.2) and concludes with the computation of pattern matrices and weights (Sec. 5.3).

5.1. Estimating K by Clustering

Here we assume that we know the parameters $N_\mathbf{v}$ and $N_\mathbf{h}$ (we will show how to compute them in Sec. 5.4). We will describe an iterative technique which is based on the idea of clustering the rows of the rearranged counterpart of the given matrix $\mathbf{F}_{N\times M}$. In particular, we use a *partitional* (k-means) clustering algorithm in an iterative fashion in order to accurately estimate the rank of that matrix.

To this end, let us consider that matrix $\tilde{\mathbf{F}}_{l_\mathbf{v}l_\mathbf{h}\times N_\mathbf{v}N_\mathbf{h}}$ (for simplicity in the notation from now on we will denote it by $\tilde{\mathbf{F}}$), as well as the desired number of clusters we would like to group the rows of the matrix (let us denote it by K) be given, and let us define the following set consisting of K groups:

$$\mathcal{R}_k = \{\mathbf{f}_q^t : ||\mathbf{f}_q^t - \bar{\mathbf{r}}_k^t||_2^2 \le ||\mathbf{f}_q^t - \bar{\mathbf{r}}_l^t||_2^2,\ \forall\, 1 \le l \le K\} \qquad k = 1,2,\cdots,K \quad (17)$$

where $\mathbf{f}_q^t$ denotes the q-th row of matrix $\tilde{\mathbf{F}}$, and $\bar{\mathbf{r}}_k^t$ the mean of the k-th group of the rows respectively, as computed by k-means.

Let us also define the corresponding *indicator* vectors of length $l_\mathbf{v}l_\mathbf{h}$ each:

$$\mathbf{1}_{\mathcal{R}_k}[q] = \begin{cases} 1 & \text{if } \mathbf{f}_q^t \in \mathcal{R}_k \\ 0 & \text{otherwise}, \quad q = \{1,2,\cdots,l_\mathbf{v}l_\mathbf{h}\} \end{cases} \quad (18)$$

and the *element-wise* mean vectors of each group:

$$\bar{\mathbf{r}}_k^t = mean\{\mathcal{R}_k\},\ k = 1,2,\cdots,\ K \quad (19)$$

We can now define the following matrix:

$$\tilde{\mathbf{F}}_\mathcal{R} = \sum_{k=1}^{K} \mathbf{1}_{\mathcal{R}_k}\bar{\mathbf{r}}_k^t \quad (20)$$

which has the same size as $\tilde{\mathbf{F}}$. More importantly, if the given number of clusters K were the correct one, then K should equal to the rank of $\tilde{\mathbf{F}}$. If, on the other hand, the given number of clusters K is greater than the real rank of $\tilde{\mathbf{F}}$, then the rank of $\tilde{\mathbf{F}}_\mathcal{R}$ will be smaller than K. Hence, by defining the new number of the clusters as:

$$K = rank(\tilde{\mathbf{F}}_\mathcal{R}) \quad (21)$$

and repeating the above described procedure, we are expecting that after some iterations, $\tilde{\mathbf{F}}_\mathcal{R}$ will be the desired approximation of $\tilde{\mathbf{F}}$. Note that the computation of rank in Eq. (21) and as part of Algorithm 1, is a generic algorithm and not one that minimizes the rank of a matrix.

Algorithm 1: Kronecker Façade Modeling, noise-free ideal case. Input: $\tilde{\mathbf{F}}$, $K = rank(\tilde{\mathbf{F}})$

```
1: repeat
2:    Form groups $\mathcal{R}_k,\ k = 1,\dots,K$ via $k$-means (17)
3:    Form the indicator vectors $\mathbf{1}_{\mathcal{R}_k}$ of (18)
4:    Form the mean vectors $\bar{\mathbf{r}}_k^t$ of (19)
5:    Compute the matrix $\tilde{\mathbf{F}}_\mathcal{R}$ defined in (20)
6:    Compute its rank $K$ (21)
7:    Assign $\tilde{\mathbf{F}}_\mathcal{R}$ to $\tilde{\mathbf{F}}$
8: until convergence
9: Output: $\tilde{\mathbf{F}}_\mathcal{R}^\star$, $K^\star$, $\mathbf{1}_{\mathcal{R}_k}$.
   Note that $\bar{\mathbf{r}}_k^t, k = 1,2,\cdots,K^\star$ are the rows of $\tilde{\mathbf{F}}_\mathcal{R}^\star$.
```

Note also that the use of mean in Eq. (19) is in exact accordance with Lemma 1 (as will be seen in Sec. 5.3). This will provide the optimal result assuming an ideal noise-free case.

In practice though, due to variations caused by occlusions (such as trees, traffic lights, etc.), shadows, etc., instead of the mean in Step 4, we use the element-wise *median* operator:

$$\bar{\mathbf{r}}_k^t = median\{\mathcal{R}_k\}. \quad (22)$$

This is based on the robustness of the median operator (used for the estimation of the most characteristic values of rows that belong to the same cluster) and its optimality in the $\mathcal{L}_1$ sense.

A second modification is also essential. Unfortunately, Lemma 1 does not guarantee that the patterns are piece-wise constant. One way to enforce that constraint is by also forcing clustering in the columns of $\tilde{\mathbf{F}}$ as well (note that each column spans all patterns). We thus consider the matrix:

$$\tilde{\mathbf{G}} = \frac{1}{2}(\tilde{\mathbf{F}}_\mathcal{C} + \tilde{\mathbf{F}}_\mathcal{R}) \quad (23)$$

and the new number of the clusters:

$$K = \min\{rank(\tilde{\mathbf{F}}_\mathcal{R}),\ rank(\tilde{\mathbf{F}}_\mathcal{C})\}, \quad (24)$$

where $\tilde{\mathbf{F}}_\mathcal{C}$ is the column-wise clustering result. It is obtained by following the same k-means clustering, but now in the columns:

$$\mathcal{C}_k = \{\mathbf{f}_p : ||\mathbf{f}_p - \bar{\mathbf{c}}_k||_2^2 \le ||\mathbf{f}_p - \bar{\mathbf{c}}_l||_2^2,\ \forall\, 1 \le l \le K\} \qquad k = 1,2,\cdots,K \quad (25)$$

where $\mathbf{f}_p$ denotes the p-th column of matrix $\tilde{\mathbf{F}}$, and $\bar{\mathbf{c}}_k$ denotes the mean of the k-th group of the columns respectively. The corresponding *indicator* vectors of length

$N_{\mathbf{v}}N_{\mathbf{h}}$ is defined as:

$$\mathbf{1}_{\mathcal{C}_k}[p] \quad = \quad \begin{cases} 1, & \text{if } \mathbf{f}_p \in \mathcal{C}_k \\ 0 & \text{otherwise}, \quad p = \{1, 2, \cdots, N_{\mathbf{v}}N_{\mathbf{h}}\} \end{cases} \tag{26}$$

and the *element-wise* median vectors of each group:

$$\bar{\mathbf{c}}_k = median\{\mathcal{C}_k\},\ k = 1, 2, \cdots,\ K. \tag{27}$$

Then,

$$\tilde{\mathbf{F}}_{\mathcal{C}} = \sum_{k=1}^{K} \bar{\mathbf{c}}_k \mathbf{1}^t_{\mathcal{C}_k}. \tag{28}$$

Therefore, the algorithm we use in practice is shown below.

Algorithm 2: Kronecker Façade Modeling. Input: $\tilde{\mathbf{F}}$, $K = rank(\tilde{\mathbf{F}})$

```
1: repeat
2:     Form groups $\mathcal{R}_k, \mathcal{C}_k$ $k = 1, \ldots, K$ via $k$-means (17),(25)
3:     Form the indicator vectors $\mathbf{1}_{\mathcal{R}_k}$, $\mathbf{1}_{\mathcal{C}_k}$ of (18), (26)
4:     Form the vectors $\bar{\mathbf{r}}^t_k, \bar{\mathbf{c}}_k$ of (22), (27)
5:     Form the matrices $\tilde{\mathbf{F}}_R, \tilde{\mathbf{F}}_C$ and $\tilde{\mathbf{G}}$ of (20), (28) and (23)
6:     Set $K$ using (24)
7:     Assign $\tilde{\mathbf{G}}$ to $\tilde{\mathbf{F}}$
8: until convergence
9: Output: $\tilde{\mathbf{F}}^\star_{\mathcal{R}}$, $K^\star$, $\mathbf{1}_{\mathcal{R}_k}$.
   Note that $\bar{\mathbf{r}}^t_k, k = 1, 2, \cdots, K^\star$ are the rows of $\tilde{\mathbf{F}}^\star_{\mathcal{R}}$.
```

For the convergence condition in Algorithm 2, we can consider the convergence of the sum of all entries in $|\tilde{\mathbf{G}}_i - \tilde{\mathbf{G}}_{i-1}|$, where $\tilde{\mathbf{G}}_i$ denotes the $\tilde{\mathbf{G}}$ obtained in the i^{th} iteration, or set the number of iterations to a maximum prespecified number. Finally, the denoising of matrix $\tilde{\mathbf{F}}$ after Step 7 in the algorithm above, can drastically speed up the convergence.

5.2. Estimating Matrices $\mathbf{M}_k, k = 1, 2, \cdots, K^\star$

We estimate matrices $\mathbf{M}_k, k = 1, 2, \cdots, K^\star$ by reshaping each one of the $K^\star$ above mentioned $indicator$ vectors into their nominal form, that is, in a rectangular array of size $l_{\mathbf{h}} \times l_{\mathbf{v}}$ each.

Algorithm 3: Estimation of Matrices $\mathbf{M}_k$. Input: $\mathbf{1}_{\mathcal{R}_k}, K^\star$

```
for $k = 1$ to $K^\star$ do
    $\mathbf{m}_k = \mathbf{1}_{\mathcal{R}_k}$
    $\mathbf{M}_k = reshape(\mathbf{m}_k,\ l_{\mathbf{h}},\ l_{\mathbf{v}})$
end for
Output: $\mathbf{M}_k,\ k = 1, \ldots,\ K^\star$.
```

We must stress at this point that it is easy to validate that the vectorized forms of the estimated partition matrices satisfy the conditions of Eqs. (2) and (3).

5.3. Computing Patterns and Weighting Factors

At this point we have estimated all the quantities needed to find out the optimal patterns $\hat{\mathbf{P}}_k$ and weighting factors $\lambda_k,\ k = 1, 2, \cdots, K^\star$, as they are defined in Lemma1. Note that the estimated partition matrices have the desired optimal $1 - 0$ form. In addition, since each $\mathbf{m}_k$ coincides with the corresponding $indicator$ vector, and by the definition of $mean$ vectors $\bar{\mathbf{r}}^\star_k$ defined in (19), each term of the matrix $\tilde{\mathbf{F}}_{\mathcal{R}}$ of (20), has exactly the same form with the optimal patterns defined in Lemma 1. Indeed, by taking into account that by definition $\mathbf{m}_k = vec\{\mathbf{M}_k\}$, and because of the special $1 - 0$ form of the partition matrices $||\mathbf{m}_k||_2^2 = ||\mathbf{m}_k||_0$, the following is true:

$$\lambda^\star_k vec\{\hat{\mathbf{P}}^\star_k\} = \frac{\mathbf{U}\boldsymbol{\Sigma}^T\mathbf{V}^T vec\{\mathbf{M}_k\}}{||vec\{\mathbf{M}_k\}||_2^2} = \bar{\mathbf{r}}^\star_k, k = 1, ..., K^\star. \tag{29}$$

Therefore, the vectors $\bar{\mathbf{r}}^\star_k$, computed in Algorithm 1, provide us the weighted optimal patterns. In practice, as discussed in Sec. 5.1, we are using the results of Algorithm 2.

5.4. Estimating the Spatial Periods of the Patterns

In all the steps of the proposed algorithm we have assumed that the spatial periods of the patterns were known. However, they are unknown and must be estimated. Although well known methods ([3, 12]) can be used for that purpose, we propose the use of the algorithm in Sec. 5.1. The only difference is that the input to the algorithm is the actual façade matrix $\mathbf{F}$ and not its rearranged form $\tilde{\mathbf{F}}$. In particular, let us run Algorithm 2 for a predefined value K_0 of the parameter K once with input $\mathbf{F}$, and then with input $\mathbf{F}^t$. Then, we can compute the following:

$$||\mathbf{1}_{\mathcal{R}_{k^\star}}||_0 \quad = \quad \max_{k=1,2,\cdots,K_0} \{||\mathbf{1}_{\mathcal{R}_k}||_0\} \tag{30}$$

$$||\mathbf{1}_{\mathcal{C}_{l^\star}}||_0 \quad = \quad \max_{l=1,2,\cdots,K_0} \{||\mathbf{1}_{\mathcal{C}_l}||_0\}, \tag{31}$$

and the corresponding auto-correlation sequences:

$$r_{\mathcal{R}_{k^\star}} \quad = \quad \mathbf{1}_{\mathcal{R}^\star_k} * \mathbf{1}_{\mathcal{R}^\star_k} \tag{32}$$

$$c_{\mathcal{C}_{k^\star}} \quad = \quad \mathbf{1}_{\mathcal{C}^\star_l} * \mathbf{1}_{\mathcal{C}^\star_l} \tag{33}$$

where " $*$ " denotes the correlation operator. Note that by taking into account Eqs. (30-31), $indicator$ vectors $\mathbf{1}_{\mathcal{R}_{k^\star}}$, $\mathbf{1}_{\mathcal{C}_{l^\star}}$ are the vectors that define the dominant row and column spatial periods respectively and thus the computation of the corresponding auto correlation sequences makes sense. Note also that the vectors involved in the computation of the proposed auto-correlation sequences are based on $indicator$ vectors, that is $1 - 0$ vectors, and not on gray-value quantities.

Algorithm 4 : Estimation of Periods $N_{\mathbf{h}}$, $N_{\mathbf{v}}$.
Input: $\mathbf{F}_{N\times M}$, K_0

```
Form the vectors $\mathbf{1}_{\mathcal{R}_k}, k = 1, 2, ..., K_0$ using (18)
Form the vectors $\mathbf{1}_{\mathcal{C}_l}, l = 1, 2, ..., K_0$
Compute the quantities defined in Eqs. (30-31)
Compute the sequences defined in Eqs. (32-33)
Use them to estimate the desired spatial periods
Output: $\hat{N}_{\mathbf{h}}$ and $\hat{N}_{\mathbf{v}}$.
```

The results we obtained with $K_0 = 5$ in the urban building façade of Fig. 2 (left), are shown in Figs. 3 (a) and 3(b) respectively.

6. Experiments and Discussion

The experiments are implemented in Matlab, and run on a computer with an 1.8 GHz Intel Core i7 CPU and a 4GB memory. To evaluate the performance of our VPD initialization scheme (Sec. 3), we run TILT with its original branching initialization scheme and with our vanishing points initialization. The results are shown in Table 1. The urban images we use for test include 182 façade images we collected in New York City as well as 124 sample façades from TILT's web resources. The results clearly state that in urban environments the use of vanishing points significantly improve the quality of the results. We thus propose to use our automated initialization technique in those cases in order to first rectify and then use TILT (or other similar methods) for improvement.

In a separate experiment we tested our repeated pattern detection in 89 images for which we had ground-truth [13, 15]. Out of the 89 images we tested, only 4% resulted to failure detections (see failure cases in Fig. 6). The results from the remaining 96% were very similar to the ground-truth. We overlaid our results with the ground-truth pixel by pixel and had exact matches for 91% of the pixels.

A more extensive collection of results can be found in supplemental material. In this paper we present some representative images. Our low-rank method (Secs. 5.1, 5.2) enables us to remove occlusions, small illumination variations and photometric distortions as seen in the fourth column of Fig. 4, 5, and 6. Because of this we have very accurate detection of repeated patterns. Based on those clean patterns, we can easily obtain 1-0 patterns (i.e. refining the results) by applying classification methods, such as the rank-one algorithm [15], within each group. Examples of detected 1-0 patterns are shown in the last column of Fig. 4, 5, and 6. For example the method of [15] fails in the case of Fig. 5, due to tree occlusion. Our algorithm, however, can successfully detect four different clusters and clear pattern structures.

We can conclude that the block partition Sec. 5.4 is not a bottleneck of our algorithm. The partition lines may pass across the desired patterns, as shown in the second row of Fig.6. In such cases some pattern is divided into two adjacent partition blocks, such that the partition blocks don't contain the desired patterns completely. However our algorithm is robust enough to detected them separately. We must stress at this point that a better partition can definitely improve the performance. In order to have partition lines mostly passing through wall areas as desired, we can adopt methods proposed in [10, 3].

Initialization_method	Run Time	Success Rate
Branch-and-bound	36.63s	65%
VPD	38.6s	84.6%

Table 1. Performance of TILT with branching and VPD transform initialization (Sec. 3) respectively. The vanishing points initialization results in significant increase in success rate without a penalty in speed. The experiments are done on a set of 306 façade images.

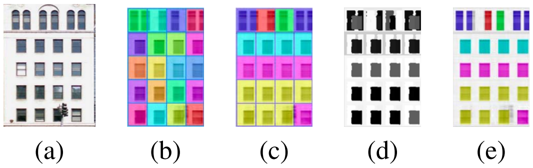


Figure 4. (a) Input image, (b) partition grid showing the periods estimated by Algorithm 4, with all partition blocks colored randomly, (c) grouped blocks generated by Algorithms 2 and 3, with each group having the same color, (d) low-rank component generated by Algorithms 2 - 4 in Sec. 5.1 - 5.4, and (e) estimated 1-0 repeated patterns by refining (d).

In the experiments, we found most of the common building façades to be able to be modeled by our Kronecker product structure. One limitation is that our method fails when a façade contains repeated structures that do not follow the Kronecker product model, such as in the bottom row of Fig. 6. Another limitation is the inability to handle large photometric variations, since they are causing ambiguity in the block partition (second to last row of Fig. 6). Unfortunately, currently there is no simple way for the system to automatically determine failure cases.

In conclusion, this paper describes a novel method for detection of repeated patterns following a Kronecker Product formulation. Our method is general and can be applied to a wide variation of façade structures and is being based on a solid theoretical foundation. The fact that we are utilizing the low-rank part of the rearranged input façade image allows us to handle problems of occlusion, shadows and illumination variations. Other image processing, low-rank and SVD-based techniques that can be used for increasing the robustness and speed of convergence are currently under investigation.

References

[1] O. Barinova, V. Lempitsky, E. Tretiak, and P. Kohli. Geometric image parsing in man-made environments. *ECCV*, 2010.

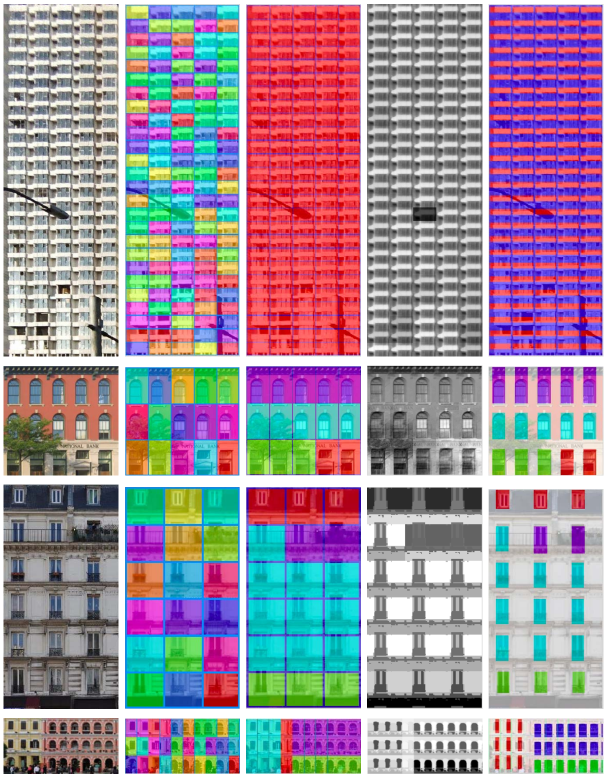


Figure 5. From left to right: follow the order of Fig. 4. The first row is a skyscraper example, where there is only one group. The second row is a failure case of method presented in [18] (large tree occlusion), but our method can successfully detect the repeated patterns as shown in the last column. The third row is an example from the dataset of [1]. The last row shows detected patterns for the example in Fig. 2.

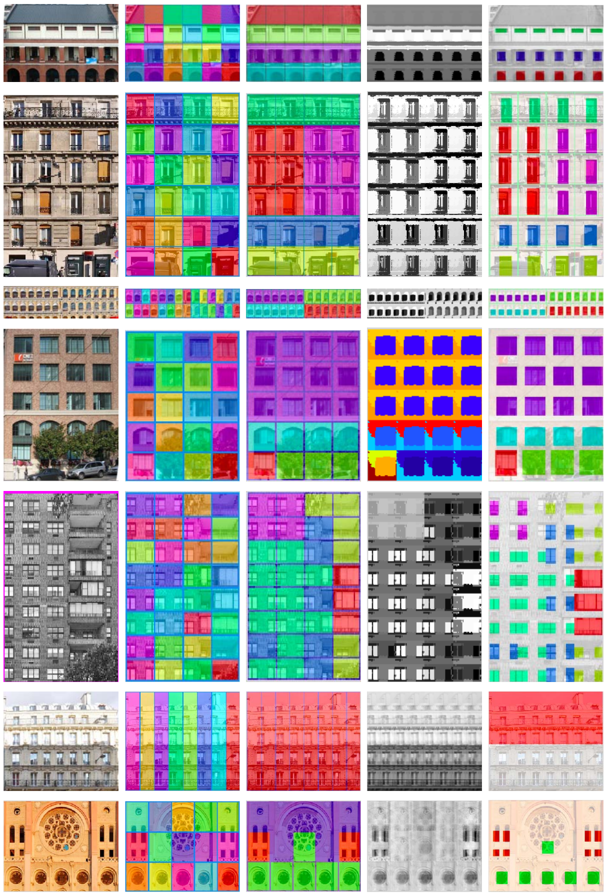


Figure 6. From left to right: follow the order of Fig. 4. First five rows show success cases (robustness to occlusions and different architectural styles). Last two rows are failure cases due to the photometric variation and inability to model via a Kronecker product model.

[2] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *ACM*, 2011.

[3] S. Friedman and I. Stamos. Online detection of repeated structures in point clouds of urban scenes for compression and registration. *IJCV*, 2013.

[4] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011.

[5] C. Harris and M. Stephens. A combined corner and edge detector. *Fourth Alvey Vision Conference*.

[6] B. Li, K. Peng, X. Ying, and H. Zha. Simultaneous vanishing point detection and camera calibration from single images. *ISVC*, II:151–160, 2010.

[7] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *PAMI*, 2009.

[8] Y. Liu, R. T. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *PAMI*, 2004.

[9] C. F. V. Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123:85–100, 2000.

[10] P. Muller, G. Zeng, P. Wonka, and L. V. Gool. Image-based procedural modeling of facades. *Siggraph*, 2007.

[11] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, and F. Dellaert. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. *CVPR*, 2008.

[12] C.-H. Shen, S.-S. Huang, H. Fu, and S.-M. Hu. Adaptive partitioning of urban facades. *ACM (TOG)*.

[13] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. *CVPR*, 2011.

[14] C. Wu, J.-M. Frahm, and M. Pollefeys. Detecting large repetitive structures with salient boundaries. *ECCV*, 2010.

[15] C. Yang, T. Han, L. Quan, and C.-L. Tai. Parsing façade with rank-one approximation. *CVPR*, 2012.

[16] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. Tilt: Transform invariant low-rank textures. *ACCV*, 2010.

[17] P. Zhao, T. Fang, J. Xiao, H. Zhang, Q. Zhao, and L. Quan. Rectilinear parsing of architecture in urban environment. *CVPR*, 2010.

[18] P. Zhao and L. Quan. Translation symmetry detection in a fronto-parallel view. In *CVPR*, 2011.