A Kronecker Product Model for Repeated Pattern Detection on 2D Urban Images

Juan Liu, Emmanouil Z. Psarakis, Yang Feng, and Ioannis Stamos

Abstract—Repeated patterns (such as windows, 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 pattern detection in a precise, efficient and automatic way, by combining traditional feature extraction with a Kronecker product based low-rank model. We introduced novel algorithms that extract repeated patterns from rectified images with solid theoretical support. Our method is tailored for 2D images of building façades and tested on a large set of façade images.

Index Terms—Repeated pattern detection, low-rank, Kronecker product model, urban façade

1 INTRODUCTION

Urban scenes contain rich periodic or near-periodic structures, such as windows, doors, and other architectural features. Detection of 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. However, it is a challenging task due to scene occlusion, varying illumination, pose variation and sensor noise.

A pre-processing rectification step is common to all façade parsing algorithms. This work also relies on such a pre-processing step based on the methods presented in [1] and [2]. Our pipeline focuses on the detection of repeated patterns in rectified façade images (Sections 3 and 4). State-of-the-art methods use classification [3], statistical and grammar-based approaches [4] and [5], as well as feature-based symmetry [6]. We, on the other hand, provide a novel detection method to model repetition as a Kronecker product.

2 RELATED WORK

In recent years, repeated pattern or periodic structure detection has received significant attention in both 2D images ([4], [7]) and 3D point clouds ([8], [9]). Repeated patterns are usually hypothesized from the matching of local image features. They can be modeled as point clouds ([8], [9]). Repeated patterns are usually hypothesized. The most prominent and significant features in urban scenes. Therefore, detection of these repeated patterns becomes very important for city scene analysis. This paper introduces novel algorithms that extract repeated patterns from rectified images with solid theoretical support. Our method is tailored for 2D images of building façades and tested on a large set of façade images.

Muller et al. [13] proposes an approach to detect symmetric structures in a rectified fronto-façade and to reconstruct a 3D geometric model. The work of [3] 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 [4]. Other grammar-based approaches include [14].


All the above-mentioned methods require image rectification as a pre-processing step. To solve this problem, low-rank methods were used and attracted a lot of attention in recent years [2]. A similar work was proposed by [17] 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 [18]. Finally, [19] describes a low-rank based method that detects the repeated patterns in 2D images for the application of shape completion.

3 FAÇADE MODELING VIA KRONECKER PRODUCTS

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

3.1 Ideal Façade Modeling

Let us consider the partition of all-ones matrix $I_{l \times l}$ of size $l \times l$ by using the following mutually exclusive 0–1 matrices $M_k$, $k = 1, 2, \cdots, K$ of size $l \times l$:

$$< \text{vec}(M_k), \text{vec}(M_l)> = \begin{cases} \|\text{vec}(M_k)\|_0, & k = l, \\ 0, & k \neq l. \end{cases}$$

where $\text{vec}(X)$, $X \in \mathbb{R}^{m \times n}$ denote the column-wise vectorization of matrix $X$, the inner product of vectors $x$ and $y$, and the $l_0$ norm of vector $x$, respectively. As it is clear from Eqs. (1)–(2), different choices of matrices $M_k$ result in different partitions of matrix $I_{l^2 \times l^2}$. Let us now associate with each component $M_k$, $k = 1, 2, \cdots, K$ of the partition of matrix $I_{l^2 \times l^2}$ defined in Eq. (2), a 2-D pattern $P_k$ of size $N_v \times N_v$ that is going to be repeated according to $M_k$. The patterns should have a piecewise constant surface form, with example structures including windows, doors and balconies.

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

$$\mathcal{F}_{N \times M} = \sum_{k=1}^{K} \lambda_k (M_k \otimes P_k),$$

where $X \otimes Y$ is the Kronecker product of matrices $X$ and $Y$, and $\lambda_k$, $k = 1, 2, \cdots, K$ are weights. Here, $N \times M$ is the size of the urban building façade image. It is obvious that $N = l_i N_i$ and $M = l_h N_h$. The urban building façade’s model defined in Eq. (3) can be used even in cases where there is no periodic structure.

Generalizing Eq. (3) to include a “wall” gray level $\lambda_0$, we get:

$$\mathcal{F}_{N \times M} = \lambda_0 I_{N \times M} + \sum_{k=1}^{K} \lambda_k (M_k \otimes P_k).$$

Using the fact that the components of the partition of array $I_{l_i \times l_h}$ of Eq. (2) are mutually exclusive, we rewrite Eq. (4) as:
where \( \mathbf{P}_k \) are modified patterns as defined above.

### 3.2 Façade Model Approximation

In this section we would like to decompose the components of the Kronecker product that generate an ideal (i.e. noise-free) building façade \( \mathbf{F}_{N \times M} \in \mathbb{R}^{N \times M} \) with \( N = l_1 N_v \) and \( M = l_2 N_h \). Using the model of Eq. (5) we can define the following cost function:

\[
\mathcal{C}_F(\mathbf{M}_k, \mathbf{P}_k, \lambda_k, k = 1, \ldots, K) = \| \mathbf{F}_{N \times M} - \mathcal{F}_{N \times M} \|_2^2
\]

\[
= \left\| \mathbf{F}_{N \times M} - \sum_{k=1}^{K} \lambda_k (\mathbf{M}_k \otimes \mathbf{P}_k) \right\|_2^2,
\]

where \( \| \cdot \|_2 \) represents the Frobenius norm of a matrix. As it is clear from its definition, \( \mathcal{C}_F(\cdot) \) is a Frobenius 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 façade \( \mathbf{F}_{N \times M} \) can be expressed by the following minimization problem

\[
\min_{\mathbf{M}_k, \mathbf{P}_k, \lambda_k, k = 1, \ldots, K} \mathcal{C}_F(\mathbf{M}_k, \mathbf{P}_k, \lambda_k, k = 1, \ldots, K),
\]

which is known as the nearest Kronecker product problem [20]. The following partition of \( \mathcal{F}_{N \times M} \) is key to the solution of Eq. (7):

\[
\mathcal{F}_{N \times M} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \cdots & \mathbf{F}_{1h} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{h1} & \mathbf{F}_{h2} & \cdots & \mathbf{F}_{hh} \end{bmatrix},
\]

where \( \mathbf{F}_{ij} \) is a block of size \( N_v \times N_h \). Define matrix

\[
\mathbf{A}_{N_v \times N_h} = [\text{vec}(\mathbf{F}_{11}) \text{ vec}(\mathbf{F}_{12}) \cdots \text{vec}(\mathbf{F}_{hh})]^T,
\]

which constitutes a rearrangement of \( \mathcal{F}_{N \times M} \). Using the above defined quantities, the cost function of Eq. (6) can be expressed as:

\[
\mathcal{C}_F(\mathbf{m}_k, \mathbf{p}_k, \lambda_k, k = 1, \ldots, K) = \| \mathbf{A}_{N_v \times N_h} - \sum_{k=1}^{K} \lambda_k \mathbf{m}_k \mathbf{p}_k^T \|_2^2,
\]

where \( \mathbf{m}_k, \mathbf{p}_k \) are the column-wise vectorized forms of \( \mathbf{M}_k, \mathbf{P}_k \). By exploiting the above defined equivalent form of the cost function, the Kronecker Product SVD [20] can be used to solve the optimization problem in Eq. (7).

**Theorem 1.** Let \( \mathbf{A}_{N_v \times N_h} = \mathbf{U} \Sigma \mathbf{V}^T \) be the Singular Value Decomposition (SVD) of \( \mathcal{F}_{N \times M} \). Consider the following diagonal matrix

\[
\Sigma_K = \text{diag}\{\sigma_1, \sigma_2 \cdots \sigma_K\}
\]

containing the first \( K \) singular values of matrix \( \mathbf{A}_{N_v \times N_h} \), and let

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

be the \( K \) associated left and right singular vectors respectively. Then, the matrices \( \mathbf{M}_k \), the patterns \( \mathbf{P}_k \), and the weighting factors \( \lambda_k \) that satisfy

\[
\text{vec}(\mathbf{M}_k) = \mathbf{v}_k, \quad \text{vec}(\mathbf{P}_k) = \mathbf{u}_k, \quad \lambda_k = \sigma_k, \quad k = 1, \ldots, K,
\]

constitute the solution of the Eq. (7).

Using Theorem 1, we can find an optimal approximation in the desired form, i.e. it is a sum of Kronecker products, that minimizes the cost function defined in Eq. (6). Note, however, that some of the characteristics of the optimal solution are not consistent with the ingredients of the model defined in Eq. (4), which makes the direct use of Theorem 1 problematic. In particular, neither the matrices \( \mathbf{M}_k \) nor the patterns \( \mathbf{P}_k \) have the desired form in general, i.e. they are not 1-0 matrices and piecewise constant surfaces, respectively.

In order to impose one of the requirements of the proposed model, in the sequel we assume that matrices \( \mathbf{M}_k \) have the desired 1-0 form and are known. We thus form the cost function:

\[
\hat{\mathcal{C}}_F(\mathbf{P}_k, \lambda_k, k = 1, \cdots, K(\mathbf{M}_k)),
\]

which is the cost function of Eq. (6) but with the partition matrices known. We would like to minimize it with respect to the patterns \( \mathbf{P}_k \) and the weighting factors \( \lambda_k \). The solution of the new optimization problem is the subject of Lemma 1 (refer to supplemental material for proof).

**Lemma 1.** Assuming that the matrices \( \mathbf{M}_k, \quad k = 1, 2, \cdots, K \) defined in Eqs. (1-2) are known, then the minimization of the cost function defined in Eq. (14) produces patterns \( \mathbf{P}_k \) and weighting factors \( \lambda_k \) that are related as follows:

\[
\lambda_k \text{vec}(\mathbf{P}_k) = \frac{\mathbf{U} \Sigma \mathbf{V}^T \text{vec}(\mathbf{M}_k)}{||\text{vec}(\mathbf{M}_k)||_2}, \quad k = 1, 2, \cdots, K.
\]
and the element-wise median vectors of each cluster:

\[ \hat{r}_l^e = \text{median}\{R_l\}, \quad l = 1, 2, \cdots, L. \]  

(18)

In order to be consistent with Lemma 1, Eq. (18) should be using the mean, since it provides the optimal result assuming an ideal noise-free case. However, due to variations caused by occlusions (such as trees, traffic lights, etc.) or shadows and lighting changes, the median is proven more robust in our empirical studies.

We can now define the following matrix:

\[ F_R = \sum_{l=1}^{L} I_{R_l} r_l^e, \]  

(19)

which has the same size as F. More importantly, if the given number of clusters L were the correct one, then L should be equal to the rank of F. If, on the other hand, L is greater than the rank of F, then the rank of F_R will be smaller than L. Hence, by computing the number of clusters as:

\[ L = \text{rank}(F_R), \]  

(20)

and repeating the above procedure, we expect that after some iterations, F_R will be the desired approximation of F.

**Algorithm 1.** Kronecker Façade Modeling, Noise-Free Ideal Case. Input: F

1: Initialize: L: L ← rank(F) − 1
2: repeat
3: Form clusters \( R_l \), \( l = 1, \cdots, L \) via K-means (16)
4: Form the indicator vectors \( I_{R_l} \) of (17)
5: Form the median vectors \( \hat{r}_l^e \) of (18)
6: Compute the matrix \( F_R \) defined in (19)
7: Compute its rank L in (20)
8: Assign \( F_R \) to \( F \)
9: until convergence
10: Output: \( F_R \), \( L', I_{R_l} \).

Note that \( r_l^e, l = 1, 2, \cdots, L' \) are the rows of \( F_R \).

Unfortunately, Lemma 1 does not guarantee that the patterns are piece-wise constant. One way to enforce that constraint is to force the clustering in the columns of F as well. We thus consider the matrix:

\[ G = \frac{1}{2} (F_C + F_R) \]  

(21)

and the new number of the clusters:

\[ L = \min\{\text{rank}(F_R), \text{rank}(F_C)\}, \]  

(22)

where \( F_C \) is the column-wise clustering result. It is obtained by following the same K-means clustering, but now in the columns:

\[ C_l = \{b_{l,i} : \|b_{l,i} - \bar{e}_l\|_2^2 \leq \|b_{l,i} - \tilde{e}_m\|_2^2, \quad 1 \leq m \leq L \}, \]  

(23)

where \( b_{l,i} \) denotes the \( p \)-th column of matrix F, and \( \bar{e}_l \) denotes the mean of the \( k \)-th cluster of the columns respectively. The corresponding indicator vectors of length \( M \) is defined as:

\[ I_{C_l}[p] = \begin{cases} 1, & \text{if } b_{l,i} \in C_l, \\ 0, & \text{otherwise}, \end{cases} \]  

(24)

where \( p = 1, \cdots, M \).

and the element-wise median vectors of each cluster:

\[ \tilde{c}_l = \text{median}\{C_l\}, \quad l = 1, 2, \cdots, L. \]  

(25)

Then,

\[ F_C = \sum_{l=1}^{L} c_l^e I_{C_l} \]  

(26)

The algorithm is summarized as follows.

**Algorithm 2.** Kronecker Façade Modeling. Input: F

1: Initialize: L: L ← rank(F) − 1
2: repeat
3: Form clusters \( R_l, C_l, l = 1, \cdots, L \) via K-means using (16) and (23)
4: Form the indicator vectors \( I_{R_l}, I_{C_l} \) of (17), (24)
5: Form the median vectors \( \hat{r}_l^e, \tilde{c}_l \) of (18), (25)
6: Form the matrices \( F_R, F_C \) and G of (19), (26) and (21)
7: Set L using (22)
8: Assign G to F
9: until convergence
10: Output: \( F_R, L', I_{R_l}, I_{C_l} \).

Note that \( r_l^e, l = 1, 2, \cdots, L' \) are the rows of \( F_R \).

The results for the urban building façade in Fig. 2, top, are shown in Fig. 1. The façade can be partitioned into blocks by using the estimated spatial periods, as illustrated in Fig. 2.

### 4.2 Estimation of K by Unbiased Estimator of the Degrees of Freedom

Given the computed \( N_s \) and \( N_h \), we can rearrange the façade F into A as described in Eq. (9) (see a representative image in the supplemental material).

If the façade contains repeated patterns, then the partition blocks can be clustered into groups. As we have re-arranged the partitioned blocks into vectors, each group of repeated patterns is in the form of a group of repeated vectors in A. Thus \( K \) represents the rank of A and the problem reduces to the estimation of rank of A. Therefore the problem can be formed as the following statistical problem of finding the correct rank of a perturbed low-rank matrix: given a noisy observation \( A = A^o + E \), the goal is to estimate an \( m_1 \times m_2 \) matrix \( A^o \), where \( m_1 = l_i J_h, m_2 = N_s N_h \) and \( \text{rank}(A^o) = K \). We
assume that the noise matrix $E$ follows a matrix normal distribution where $E_{ij} \sim N(0, \tau^2)$ and $m_1 \leq m_2$ so that the full rank is $m_2$.

A $K$-means clustering based iterative algorithm was used in [1] to estimate the rank of $A$. However, the $K$-means based approach is computationally expensive and unstable in cases where there is occlusion caused by illumination or shadows. In [21], an approach was proposed to address this problem via Degrees of Freedom estimators in an efficient and reliable way. Based on this idea, we will describe a statistical technique that estimates the rank $K$ of $A$.

In [22] and [23], a rigorous definition of degrees of freedom in the framework of Stein’s Unbiased Risk Estimate (SURE) was provided. For the classical linear regression, degrees of freedom is often associated with the number of variables in the model. However the parallel interpretation is unclear in the context of low rank matrix estimation problems where the estimators are highly non-linear in nature. The number of free parameters in specifying a low rank matrix is often used as the degrees of freedom in this case. It was shown in [21] that the number of free parameters incorrectly measures the complexity of the rank constrained estimator.

Let $A = UΣV^T$ be the SVD of $A$. The estimator of $A$ with rank $K$, denoted as $\hat{A}_K$, is defined as:

$$\hat{A}_K = \sum_{k=1}^{K} u_k v_k^T,$$

(31)

where $u_k$ and $v_k$ are the $k$th column of $U$ and $V$ respectively.

The formal definition of the optimization function for estimating $K$ is formulated as:

$$\ell(\hat{A}_K) = \|\hat{A}_K - A\|_F^2 = \|\hat{A}_K - (A - E)\|_F^2$$

$$= \|\hat{A}_K - A\|_F^2 + 2 \times \langle \hat{A}_K - A, E \rangle + \|E\|_F^2$$

$$= \|\hat{A}_K - A\|_F^2 + 2 \times \langle \hat{A}_K, E \rangle$$

+ (terms not depending on $\hat{A}_K$),

(32)

where $K \in \{1, \ldots, m_1\}$ is a tuning parameter and $\langle \cdot, \cdot \rangle$ stands for the inner product. The first term measures the goodness of fit of $\hat{A}_K$ to the observation $A$. The second term can be interpreted as the cost of the estimating procedure and can be estimated by using degrees of freedom as shown in [21].

$$df(\hat{A}_K) = \frac{1}{\tau^2} \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \operatorname{cov}(\hat{A}_{Kij}, E_{ij}).$$

(33)

We refer the interested readers to [22] and [23] for further discussions about the general theory regarding degrees of freedom. Once the degrees of freedom are defined, the rank estimator can be constructed by using the following $C_p$ type statistic:

$$C_p(\hat{A}_K) = \|\hat{A}_K - A\|_F^2 + 2\tau^2 df(\hat{A}_K).$$

(34)

where $\tau^2$ is defined as $\operatorname{var}(\hat{A}_K - A)$, and $\hat{A}_K$ corresponds to a low-rank estimate which explains $\lambda$ proportion of the total variance, where $0 \leq \lambda \leq 1$. Here, we fix $\lambda = 0.935$ through all the experiments.

Usually, the degrees of freedom defined in Eq. (33) are not directly computable. The unbiased estimator proposed in [24] lacks analytical expressions and requires numerical methods such as data perturbation and resampling techniques which are computationally prohibitive in large scale problems. For our specific rank regularized estimation problem, we employ the following unbiased estimator of degrees of freedom (see details in Theorem 1 of [21]):

$$df(\hat{A}_K) = (m_1 + m_2 - K)K + 2 \sum_{k=1}^{K} \sum_{l=k+1}^{m_1} \frac{\sigma_l^2}{\sigma_k^2 - \sigma_l^2}.$$  

(35)

Using Eq. (35), we arrive at the following estimator of the $C_p$ statistic for each candidate rank $K$:

$$\hat{C}_p(\hat{A}_K) = \|\hat{A}_K - A\|_F^2 + 2\tau^2 df(\hat{A}_K).$$

(36)

The estimated rank $K^*$ is then defined as follows:

$$K^* = \arg \min_{1 \leq K \leq m_1} \hat{C}_p(\hat{A}_K).$$

(37)

The quantity $C_p(\hat{A}_K)$ as a function of $K$ for the façade in Fig. 2 (top) is shown in Fig. 3. As we described in Section 3, each row of this low-rank matrix indicates one partition block of the input façade. By reshaping each row to a matrix in the original partition block size and re-arranging the blocks in their original order, we obtain a “clean” façade image where the noise and occlusions have
mean vector for cluster information, we estimate matrices in Eq. (9).

In Eq. (3) by reshaping each indicator vector 13:

Fig. 5. Illustration of classification and refinement steps.

been largely removed, as shown in Fig. 4 (bottom). Simultaneously these re-shaped patterns can be easily clustered into K groups as illustrated in Fig. 4.

The algorithm described in this section can be summarized as follows.

Input: A of size $m_1 \times m_2$. A is the rearranged matrix as defined in Eq. (9).

1: $C_{\text{min}} \leftarrow -\infty$, $K^* \leftarrow m_1$
2: Compute the SVD of $A$, $A = U\Sigma V^T$
3: for $K = 1$ to $m_1$ do
4: $\hat{A}_K \leftarrow \sum_{k=1}^{K} \sigma_k u_k v_k^T$
5: $f_1(K) \leftarrow \|\hat{A}_K - A\|_2^2$
6: $df(\dot{A}_K) = (m_1 + m_2 - K) + 2 \sum_{k=1}^{K} \sum_{l=K+1}^{m_1} \frac{\sigma_l^2}{\sigma_k^2}$
7: $\tau^2 \leftarrow \text{var}(\hat{A}_K - A)$
8: $C_p(K) \leftarrow f_1(K) + 2\tau^2 df(\dot{A}_K)$
9: if $C_p(K) < C_{\text{min}}$ then
10: $C_{\text{min}} \leftarrow C_p(K)$, $K^* \leftarrow K$
11: end if
12: end for
13: Output: $\hat{A}_{K^*}$, $K^*$.

We apply K-means to the rows of the matrix $\hat{A}_{K^*}$ with $K^*$ clusters. The result can be represented in terms of length-$l_h \times l_v$ indicator vectors (similar to Eq. (17)) which are denoted as $\mathbf{m}_i^k$, $k = 1, \ldots , K^*$, i.e. $\mathbf{m}_i^k(i)$ has value 1 if and only if the row $i$ belongs to cluster $k$. The mean vector for cluster $k$ from the K-means is denoted by $\mathbf{p}_k^i$ for $k = 1, \ldots , K^*$.

To get the detected repeated patterns along with the cluster information, we estimate matrices $\mathbf{M}_k$ and $\mathbf{P}_k$ by reshaping each indicator vector $\mathbf{m}_i^k$ and $\mathbf{p}_k^i$ into a rectangular array of size $l_v \times l_h$ and $N_v \times N_h$, respectively.

Fig. 6. Left: The original façade. Right: The recovered “clean” façade structure hidden beneath the noise.

Fig. 7. (a) Input image. (b) Partition grid showing the estimated periods. (c) Low-rank component generated by the method in Section 4.2. (d) Estimated 1-0 repeated patterns computed by Algorithm 3 and the refinement step. Each color represents one group. (e) Ground truth.

Fig. 8. (a) A failure case in [3]. (b) Our method detects all patterns. (c) In [3], the bottom two patterns are not detected. (d) Ground truth.

4.3 Pattern Refinement

The low-rank method described above enables us to remove occlusions, small illumination variations and photometric distortions as seen in Fig. 6. As a result, we have very accurate detection of repeated patterns. We next classify each detected pattern $\mathbf{P}_k^i$ into wall vs. non-wall pixels, using the pre-trained random forest classifier in [3]. That classifier was trained on 140 facade images that do not overlap with our testing images. Each classified pattern is further refined based on the rank-one algorithm of [3], which minimizes the $l_0$-norm of the approximation error. The classification and refinement steps are illustrated in Fig. 5 with an example of detected 1-0 patterns shown in Fig. 7d.

5 EXPERIMENTS AND DISCUSSION

The experiments were executed in Matlab, and run on a computer with an 1.8 GHz Intel Core i7 CPU and a 4GB memory.

We have created a ground truth dataset for a set of 104 façade images from [3], [4]. See Fig. 8(d) for an example. To this end we have manually marked the groups of repeated patterns for each image. The number of groups for each image is the parameter $K$.

Fig. 4. Top: Each color represents one group. Bottom: The reconstructed façade image.

Fig. 7. (a) Input image. (b) Partition grid showing the estimated periods. (c) Low-rank component generated by the method in Section 4.2. (d) Estimated 1-0 repeated patterns computed by Algorithm 3 and the refinement step. Each color represents one group. (e) Ground truth.

Fig. 8. (a) A failure case in [3]. (b) Our method detects all patterns. (c) In [3], the bottom two patterns are not detected. (d) Ground truth.

4.3 Pattern Refinement

The low-rank method described above enables us to remove occlusions, small illumination variations and photometric distortions as seen in Fig. 6. As a result, we have very accurate detection of repeated patterns. We next classify each detected pattern $\mathbf{P}_k^i$ into wall vs. non-wall pixels, using the pre-trained random forest classifier in [3]. That classifier was trained on 140 facade images that do not overlap with our testing images. Each classified pattern is further refined based on the rank-one algorithm of [3], which minimizes the $l_0$-norm of the approximation error. The classification and refinement steps are illustrated in Fig. 5 with an example of detected 1-0 patterns shown in Fig. 7d.

5 EXPERIMENTS AND DISCUSSION

The experiments were executed in Matlab, and run on a computer with an 1.8 GHz Intel Core i7 CPU and a 4GB memory.

We have created a ground truth dataset for a set of 104 facade images from [3], [4]. See Fig. 8(d) for an example. To this end we have manually marked the groups of repeated patterns for each image. The number of groups for each image is the parameter $K$.

Fig. 5. Illustration of classification and refinement steps.

Group 1
Group 2
Group 3
Group 4

1-0 patterns re-construction via Kronecker Product Model

\[ \sum_{k=1}^{K^*} (\mathbf{M}_k^i \otimes \mathbf{P}_k^i) \]
being estimated by Algorithm 3. We tested our method on all 104 images. The detection of repeated patterns in one image is considered a failure when the estimated $K$ is smaller than the true $K$, because in that case it is impossible to recover all the repeated patterns. On the other hand, when the estimated $K$ is the same as or larger than the true $K$ we could recover the repeated patterns. But the larger the estimated $K$, the more fragmented the detection will be. That is why in addition of counting the number of failures, we also report the estimation error $\tilde{K} - K$, where $\tilde{K}$ represents the estimated $K$. The smaller this difference the better.

Out of the 104 images, our method has 4 failures, while the method in [1] produces 6 failures in a subset consisting of 95 images. The sample means and variances of $\tilde{K} - K$ for our method and [1] are presented in Table 1. It is clear that by using our low-rank approach, the resulting estimate is more accurate (in terms of sample mean) and more robust (in terms of sample variance). To show the improvement over [1], we present some representative images in Figs. 10 and 11.

We evaluated the accuracy of the detected repeated pattern pixels versus the wall pixels, using an additional metric. For this we first created a binary image, in which 1 represents a detected pattern pixel, and 0 the wall pixels. We overlaid this binary image with the corresponding ground truth binary image pixel by pixel and had exact matches for 93 percent of the pixels on average over the 104 images.

Fig. 8 presents an image where the proposed method recovered all the patterns while the method in [3] did not. A more extensive collection of results can be found in supplemental material.

We found that in the experiments most of the common building façades can be captured by the 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 the Penrose tiling style in the third row of Fig. 9. Another limitation is the inability to handle large photometric variations, since they are causing ambiguity in the block partition (refer to the first two rows of Fig. 9 for examples).

Table 1

<table>
<thead>
<tr>
<th>Method</th>
<th>Sample mean</th>
<th>Sample variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method of [1]</td>
<td>1.95</td>
<td>6.09</td>
</tr>
<tr>
<td>Our method</td>
<td>0.81</td>
<td>2.01</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENTS

We thank the editor, the AE and two anonymous referees for the insightful comments which has greatly improved the paper. Feng was partially supported by NSF CAREER Grant DMS-1554804. Stamos was partially supported by NSF Grant IIS-0915971 and CUNY bridge funding.
REFERENCES


