PCA-BASED RECOGNITION FOR EFFICIENT INPAINTING

Paper Number: P1463

ABSTRACT

We present a technique for efficiently constructing a "clean" texture map of a partially occluded building facade from a series of images taken by a moving camera. After a robust registration procedure, building regions blocked by trees, signs, people, and other foreground objects are automatically inferred via the median absolute deviation of colors from different source images mapping to the same mosaic pixels. In previous work we extended an existing non-parametric inpainting algorithm for filling such holes to incorporate spatiotemporal appearance and motion cues in order to correctly replace the outlier pixels of the texture map. In contrast to other inpainting techniques that perform an exhaustive search over the image, in this work we introduce a principal components-based method that learns to recognize patches that locally adhere to the properties of the building being mapped, resulting in a significant performance boost with results of indistinguishable quality. Results are demonstrated on sequences where previous stitching and inpainting algorithms fail.

1. INTRODUCTION

As part of a vision-based architectural modeling project, we want to capture the visual appearance of buildings via robot-based "scanning." Assuming a polyhedral model of a building's structure [1, 2], a major subgoal of the task is to obtain a high-fidelity texture map of each planar section of its facade. Creating such a *mosaic* from a sequence of overlapping images via homography estimation has been thoroughly studied [3, 4, 5]. However, a complicating factor that motivated our work in [6] is the possible 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. These create "holes" in the mosaic by occluding parts of the building wall from particular views.

Image/video inpainting [7, 8, 9, 10], a method for image restoration or object removal, offers a principled way to fill such holes from contextual information surrounding them either spatially or temporally. In [6] we introduced two novel methods: (1) a technique for automatically identifying occluded regions (i.e., the areas to be filled) in building facade sequences, in contrast to existing inpainting algorithms that rely on manual segmentation; and (2) a novel spatiotemporal inpainting algorithm that combines spatial information from pixels in a partially-completed mosaic with temporal cues provided by images in the *timeline*, or sequence of images captured. Like other non-parametric texture synthesis methods [8, 11], our algorithm required an exhaustive search to identify the most likely candidate pixels for replacement—over both the temporal and spatial domains. Though the visual results were satisfactory, for long sequences they could be very expensive to obtain, requiring on the order of hours to complete the inpainting procedure.

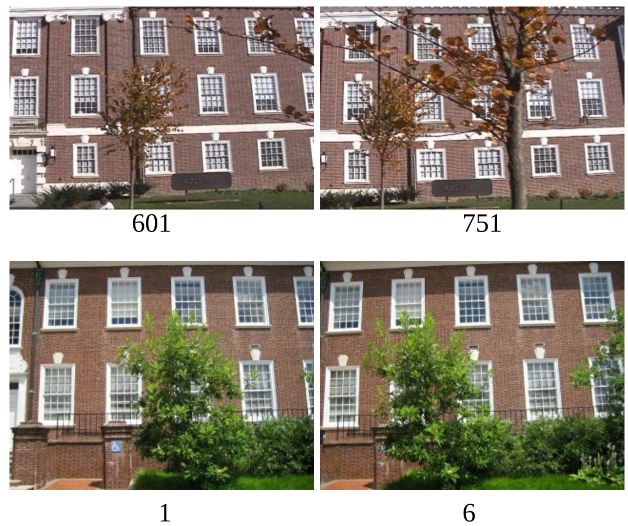


Fig. 1. Raw frames from Wolf Hall sequence (top row) and Hullihen sequence (bottom row)

The key motivation of this work is to improve our earlier algorithm by framing the search problem in inpainting as one of learning and recognizing object classes, which is much more efficient than traditional Sum-of-Squared Distances (SSD)-based searching. In a similar vein to this work, eigenface methods for face identification [12] represent the whole image as a vector of weights in a linear subspace, and some recent techniques in image retrieval [13, 14] model the appearance of object classes with a constellation of discriminative features. Here we use Principal Component Analysis (PCA) to learn a lower dimensional representation of image patches that facilitates easy recognition of the most appropriate patch. Applied to building sequences, we exploit motion cues from the timeline to restrict the number of candidate pixels that will be filled. The problem then becomes one of "building-patch recognition", akin to the face recognition methods in [12]. The most likely building pixels can then be efficiently retrieved from these candidates using the PCA-based representation.

In the rest of the paper, we first explain our PCA-based inpainting technique that searches over a much lower dimensional feature space compared to other exemplar based methods. We demonstrate it on example images widely used by the inpainting community. We then extend our synthesis from the spatial domain to include temporal information also and apply it to a vision-based application that aims to recover texture maps of occluded building facades. We compare these results to a previous technique and show equally good results at vastly improved efficiency.

2. INPAINTING BY PCA-BASED RECOGNITION

In this section we present an algorithm for filling holes in images that is built upon the work in Criminisi, Pérez, and Toyama [8], a patch-based copying method combining ideas from non-parametric texture synthesis and diffusion-based inpainting. We will refer to their method as *CPT inpainting* and briefly recapitulate the algorithm.

An empty target region Ω's pixels are filled from its border $d\Omega$ inward by copying square image patches from a source region Φ to target patches $\Psi_{\mathbf{p}}$ centered on $\mathbf{p} = (x, y) \in d\Omega$. Given the next target patch $\Psi_{\hat{\mathbf{p}}}$, an *exemplar* patch $\Psi_{\hat{\mathbf{q}}}$ is selected from Φ and pixels are copied to the unfilled portion of the target patch $\Psi_{\hat{\mathbf{p}}} \cap \Omega$ from the corresponding part of $\Psi_{\hat{\mathbf{q}}}$. Letting the entire image region be denoted by $\mathcal{I}$, $\Psi_{\hat{\mathbf{q}}}$ is chosen as the source patch with the minimum distance d (commonly the SSD) between it and the already-filled part of the target patch $\Psi_{\hat{\mathbf{p}}} \cap (\mathcal{I} - \Omega)$ (normalized for area). As inpainting proceeds Ω shrinks while Φ remains constant, leaving a band of filled pixels $\Omega_0 - \Omega_t$ at step t.

In the mold of [15, 8], a priority function $P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$ sets the order in which patches along $d\Omega$ are filled. $C(\mathbf{p})$ is a *confidence* term that measures the amount of reliable information around $\mathbf{p}$ with the formula

$$\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap (\mathcal{I} - \Omega)} \frac{C(\mathbf{q})}{|\Psi_{\mathbf{p}}|}$$

. Initially, $C(\mathbf{p}) = 0 \, \forall \mathbf{p} \in \Omega_0$ and $C(\mathbf{p}) = 1 \, \forall \mathbf{p} \in \mathcal{I} - \Omega_0$. When pixels in $\Psi_{\hat{\mathbf{p}}} \cap \Omega$ are filled in, their confidence values are updated from 0 to $C(\hat{\mathbf{p}})$, having the effect of preferring higher confidence sections of $d\Omega$ to grow before low confidence regions. $D(\mathbf{p})$ is a *data* term proportional to the dot product of the tangent vector to $d\Omega$ at $\mathbf{p}$ and the gradient vector $\nabla_{\mathbf{p}}$ with the maximum magnitude in $\Psi_{\mathbf{p}} \cap (\mathcal{I} - \Omega)$.

Fig. 2. Given a set of image patches, classify them as belonging to bulding or foreground. Note the analogy with face recognition/detection schemes.

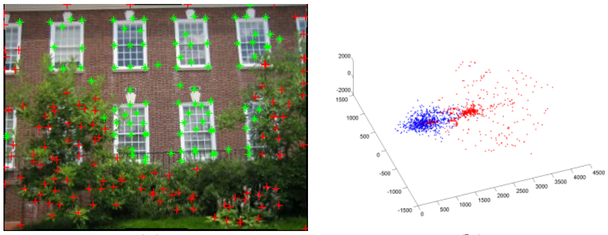
This encourages the extension of linear structures by boosting the priorities of patches with a strong edge "flowing into" them.

Most exemplar-based methods [11, 15] use the SSD as the distance function $d(\cdot, \cdot)$ between two image patches. In addition to the lack of perceptual uniformity in RGB space, for large search regions (as typically occurs with panoramas or videos), this could be very inefficient. For an 11×11 color image patch, the SSD to find the closest matching feature in Φ would require matching pairs of 363-element vectors over Φ . This can be potentially unmanageable. We therefore choose to encode image patches from Φ as a set of compact feature vectors in a lower dimensional eigenspace that allows much more efficient matching.

2.1. Computing the patch eigenspace

Given several image patches from Φ, we wish to capture almost all the variability across those patches with as few dimensions as possible. PCA has been a very popular dimensionality reduction technique widely used in recognition. It generates a set of orthonormal basis vectors, that maximize the scatter of all training samples. In spite of various limitations (gaussian distribution, orthogonal linear combinations), we have found it to be simple and adequate for the task at hand. Given an image to be inpainted and the source region Φ, we extract $n \times n$ patches from Φ that will be used to guide the inpainting. For regular inpainting, we extracted patches at every pixel, but this can be a more coarse sampling as we show in the timeline mosaicing application. Typical patch sizes that we've used are $n = 9$ and $n = 11$. We then create a vector out of these patches by concatenating all 3 color channels.

PCA is then applied to the set of $3n^2$-element vectors to build the eigenspace of patches that capture the statistics of these image patches. A similar method was also used in PCA-SIFT [16] to encode SIFT features for image retrieval applications. After PCA, each $n \times n$ patch is expressed as a vector of coordinates along the first k principal components. The value of k is chosen based on the decreasing magnitude of eigenvalues as well as empirical evaluation of the quality of reconstruction. Given a new high-dimensional patch, it is projected into feature space, where euclidean distance between points can be used to measure similarity.



(a) (b)

Fig. 3. (a) KLT features labeled as RANSAC inliers (green) and outliers (red) can be used to extract training examples. (b) Plot of first 3 principal components in feature space for the training set

3. PCA-BASED TIMELINE INPAINTING FOR MOSAICING

In this section we present an efficient algorithm for filling holes in sequence-based mosaics using the PCA-based recognition scheme. The goal of the application is to construct high-fidelity texture maps of building facades from an image sequence, even though parts of the building might be occluded by foreground objects such as trees or signs in a majority or even all of the views. Assuming that the building plane accounts for the majority of pixels in the sequence, with robust methods we can estimate the dominant motion of the building and stabilize it against the camera motion. If the foreground objects are small or fleeting, a temporal median filter can effectively recover the background from the stabilized sequence. Here we describe how our recognition-based inpainting method can efficiently recover the background even when these assumptions do not hold.

3.1. Pre-processing

Image registration is carried out to warp each frame in the sequence to a mosaic-aligned frame $\mathbf{W}_t$. Every location $\mathbf{p} = (x, y)$ in the mosaic reference frame has a set of pixels from the warped images $\{\mathbf{W}_t(\mathbf{p})\}$ associated with it which we call its *timeline* $\mathcal{T}(\mathbf{p})$. The size of each timeline $|\mathcal{T}(\mathbf{p})|$ may vary from 0 to N depending whether the pixel at $\mathbf{p}$ was imaged or not in each frame. Intuitively, since all pixels on the building facade exhibit the dominant motion, they should appear stationary in the mosaic whereas foreground objects such as trees and signs move due to parallax. This variability is measured using the median absolute deviation (MAD), and a high MAD at $\mathbf{p}$ indicates an outlier pixel in the median mosaic $\mathbf{M}_{med}(\mathbf{p})$ that needs to be inpainted.

Given that each $\mathcal{T}(\mathbf{p})$ contains an unknown mixture of background and foreground object pixels, our goal is to correctly pick or estimate each background pixel $\mathbf{M}(\mathbf{p})$ where $|\mathcal{T}(\mathbf{p})| > 0$, forming a building mosaic $\mathbf{M}$. Our inpainting framework from the previous section fits in well with the solution of this problem. The temporal information available from the timeline has already limited the possible number of candidate pixels that can be copied into the mosaic. The appearance matching problem has now become one of "recognizing" the appropriate background from a set of patches consisting of building and foreground objects (Fig. 2).

As explained in the previous section, we use PCA to project a set of labeled training image patches into a lower dimensional feature subspace. The positive examples of building patches are automatically extracted from Φ by uniformly sampling from 11×11 grids. The negative patches belonging to trees, grass and so on could either be marked manually in a semi-supervised learning fashion or automatically inferred from the RANSAC outliers in the image registration step (Fig 3a). Since the labeling of negative training examples is performed only once and offline, it doen not affect the run time. The original patches used to construct the eigenspace can be discarded after this step.

3.2. Timeline Inpainting by Recognition

Let the MAD outlier pixels be the target region Ω and the rest of the median mosaic $\mathbf{M}_{med}$ be the source region Φ. Our problem differs from pure spatial inpainting in that the timeline $\mathcal{T}$ for each $\mathbf{p} \in \Omega$, provided it contains at least one background pixel, should constrain the filling process. Thus, our major goals are to determine which, if any, pixels in $\mathcal{T}(\mathbf{p})$ are from the building background, and to integrate this information into the inpainting process. Letting $\mathcal{T}(\Psi_{\mathbf{p}}) = \{\Psi_{\mathbf{p}}^1, \ldots, \Psi_{\mathbf{p}}^{|\mathcal{T}(\mathbf{p})|}\}$ be the timeline of patches centered on $\mathbf{p}$, we create a *timeline mosaic* $\mathbf{M}_{time}$ by modifying CPT inpainting in three major ways:

1. In the first of two stages, each patch-wise pixel copy to Ω comes *from one timeline patch* $\Psi_{\hat{\mathbf{p}}}^* \in \mathcal{T}(\Psi_{\hat{\mathbf{p}}})$ maximally likely to have come from the building

2. During stage one, the updated confidences $C(\mathbf{p})$ of newly-filled pixels are set to the motion-based *background likelihoods* $p_{motion}^*(\mathbf{p})$ of the pixels in $\Psi_{\hat{\mathbf{p}}}^*$

3. If the mean background likelihood $\bar{p}_{motion}(\Psi_{\hat{\mathbf{p}}}^t)$ for every patch in $\mathcal{T}(\Psi_{\hat{\mathbf{p}}})$ is below a threshold τ_{motion}, $\Psi_{\hat{\mathbf{p}}}$ is *not filled* at that time. Stage two begins when all remaining areas of Ω meet this definition, and consists simply of CPT inpainting

Each of these three modifications is explained below:

Timeline patch selection Consider a patch $\Psi_{\hat{\mathbf{p}}}$ in the mosaic $\mathbf{M}_{time}$ that is the next to be inpainted. Pixels in its unfilled part $\Psi_{\hat{\mathbf{p}}} \cap \Omega$ will come from the corresponding part of one timeline patch $\Psi_{\hat{\mathbf{p}}}^* \cap \Omega$. We copy pixels from the timeline rather than Φ to maximize correctness, improve feature alignment, and allow for the retention of unique features not present in Φ. To pick a $\Psi_{\hat{\mathbf{p}}}^*$ that is most likely to contain building pixels rather than foreground pixels, we rely upon two cues: (1) *Appearance-based similarity* to other features in the presumed "all-building" region Φ; and (2) *Minimal motion energy* (indicating no occlusion in that frame).

Most buildings have repeated patterns such as windows, doors, columns, bricks, etc., so building (as opposed to foreground) timeline patches in Ω are likely to have a similar appearance to features in Φ. However, appearance matching alone is a less reliable indicator of "buildingness" in homogeneous areas, and can be improved by incorporating the likelihood that motion occurred in that patch in a particular timeline frame. By combining the unfilled portions of each timeline patch with the filled part from the mosaic to create a timeline of *composite patches* $\mathcal{T}(\tilde{\Psi}_{\hat{\mathbf{p}}}) = \{(\Psi^t_{\hat{\mathbf{p}}} \cap \Omega) \cup (\Psi_{\hat{\mathbf{p}}} \cap (\mathcal{I} - \Omega))\}$, we jointly measure patch t's building similarity and motion energy with the formula

$$B(\tilde{\Psi}^t_{\hat{\mathbf{p}}}) = p_{app}(\tilde{\Psi}^t_{\hat{\mathbf{p}}})\bar{p}_{motion}(\Psi^t_{\hat{\mathbf{p}}})$$

, where the probabilities measure the likelihood of a patch belonging to the background building based on appearance and motion cues respectively. Pixels are then copied from $\Psi^*_{\hat{\mathbf{p}}}$ determined by $* = \mathrm{argmax}_t\, B(\tilde{\Psi}^t_{\hat{\mathbf{p}}})$.

The evaluation of p_{app} can be expressed in a probabilistic framework using the N-Nearest Neighbor rule. Given a test patch $\Psi_{\mathbf{y}}$, we can classify it as belonging to class $\hat{\nu}$ that has the maximum posterior probability:

$$\hat{\nu} = \mathrm{argmax}_{\nu \in V} P(\nu|\Psi_{\mathbf{y}})$$

. V is the set of classes and in our case would be building and foreground. A straightforward method of computing the likelihood for each class is based on a voting scheme that returns the fraction of N-neighbors belonging to that class, but this is sub-optimal if the number of training image patches from each class is not guaranteed to be approximately the same. To evaluate the appearance properties, we first project the patch $\Psi_{\mathbf{y}}$ into the k-dimensional eigenspace. Let $(< \mathbf{x}_1, V(\mathbf{x}_1) > \ldots < \mathbf{x}_N, V(\mathbf{x}_N) >)$ be the N nearest neighbors and their associated labels from the training examples. Then we return a distance weighted likelihood

$$p_{app}(\Psi_{\mathbf{y}}) = \frac{\sum_{i=1}^{N} w_i(\Psi_{\mathbf{y}}, \mathbf{x}_i)\delta(Building, V(\mathbf{x}_i))}{\sum_{i=1}^{N} w_i(\Psi_{\mathbf{y}}, \mathbf{x}_i)}$$

where $w(\cdot,\cdot)$ is the reciprocal of the euclidean distance between the two patches and $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. Compared to [6], computing distances in a 25-dimensional eigenspace that captures almost all the variance across the patches is much more efficient than performing the SSD over the whole timeline for 11×11 patches.

The intersection of a pair of successive, thresholded difference images was suggested in [17] as a method for identifying foreground pixels. By converting the warped images to grayscale and scaling their intensity values to $[0, 1]$ to get $\{\mathbf{W}'_t\}$, we can adapt this approach to define a motion energy or *foreground image* at time t as $\mathbf{F}_t = (|\mathbf{W}'_t - \mathbf{W}'_{t-1}|) \otimes (|\mathbf{W}'_{t+1} - \mathbf{W}'_t|)$ where $|\cdot|$ is the absolute value and $\otimes$ is the pixelwise product.[1] Letting μ be the mean foreground image value over all t, we define the *background likelihood* for pixel $\mathbf{p}$ in warped image t as $p^t_{motion}(\mathbf{p}) = e^{-\mathbf{F}_t(\mathbf{p})/\mu}$, and $\bar{p}_{motion}(\Psi^t_{\hat{\mathbf{p}}})$ as the mean pixelwise background likelihood over all pixels in $\Psi^t_{\hat{\mathbf{p}}} \cap \Omega$.

Confidence term The background likelihoods $p^*_{motion}(\Psi_{\hat{\mathbf{p}}} \cap \Omega)$ are copied as the confidence values of the newly filled-in pixels in $\Psi_{\hat{\mathbf{p}}} \cap \Omega$. This tends to limit the propagation of bad choices in subsequent iterations—i.e., patches bordering areas of higher motion energy are bypassed for low motion energy areas first. The decaying confidence scheme of CPT inpainting does not apply in our case because timeline patch pixels in the interior of Ω are no less reliable than those near its edges.

Stopping criterion With no patch in $\mathcal{T}(\Psi_{\hat{\mathbf{p}}})$ from the background, there are no temporal constraints on what pixels to fill it with. Because unique features in Ω may not be similar to any patches in Φ, we detect all-foreground timelines solely on the basis of excessive motion energy. Specifically, if for every patch in $\mathcal{T}(\Psi_{\hat{\mathbf{p}}})$ the mean background likelihood $\bar{p}_{motion}(\Psi^t_{\hat{\mathbf{p}}}) < \tau_{motion}$, $\Psi_{\hat{\mathbf{p}}}$ is not filled. Subsequent inpainting in adjacent areas may allow some skipped pixels to be filled later, but stage one halts when this condition is true at every remaining $\mathbf{p} \in \Omega$. The holes that are left are generally much smaller than Ω_0, with more building structure revealed, and thus stage two can consist of pure CPT inpainting with much better results than if it had been run in place of stage one.

4. RESULTS

We show the result of our facade construction algorithm on image sequences that would not work well with current stitching or inpainting algorithms . The Wolf Hall sequence consists of 17 subsampled images from an 801 frame sequence, and captured at 30 fps from a camera moving parallel to a building facade. Examples of these are shown in the top row of Fig. 1. Several objects at different depths occlude parts of the building including trees, bushes, and a large sign. The sequence was taken in early fall and some of the leaves closely match the color of the brick, making the case for highly discriminative encoding - even in a low dimensional space. We have found our technique to be robust to these effects . The Hullihen Hall sequence is a short sequence of 6 images taken by a camera, meant to illustrate the efficacy of our technique in recovering even unanimously occluded building regions. The first and last frames, shown in the bottom row of Fig. 1, emphasize how some parts of the facade behind the bushes are never seen

[1] This of course excludes the timeline's first and last images

(a)

(b)

(c)

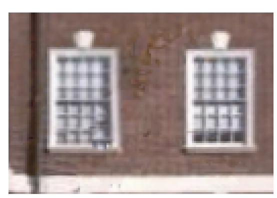
(d)

Fig. 4. (a) Median mosaic outliers for Wolf Hall sequence to be inpainted; (b) Result of PCA-based timeline inpainting followed by CPT inpainting after affine rectification (c) Median mosaic outliers for Hullihen Hall sequence (d) Result of inpainting and rectification as in (b)

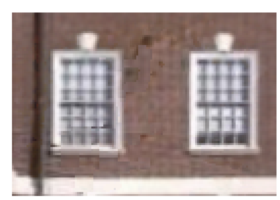
throughout the sequence.

Fig. 4 shows the result of our recognition-based inpainting algorithm that looks into the timeline of image sequences. The initial set of positive training patches to construct the eigenspace was selected from Φ. The negative examples of trees and leaves were extracted from a manually marked section in a single frame. RANSAC outliers could also be used for automatic segmentation of negative examples. In both mosaics, the ground plane outside the region of the facade was excluded from timeline inpainting.

Fig. 5 compares the result of our technique to [6]. Compared to [6] that used optimized SSD code in C as the distance function, our recognition-based approach was as much as 30 times faster even with unoptimized Matlab code. There are a couple of factors that have contributed to this improvement. Firstly, the reduced number of dimensions from 363-element vectors to $k = 25$ dimensions in the PCA eigenspace, while still retaining the distinctiveness of the patch improves the search procedure. Secondly, by our use of temporal information, we have at most $|\mathcal{T}(\mathbf{p})|$ patches that can be copied to the mosaic at $\mathbf{p}$. Since these frames are all aligned in the mosaic frame, it is theoretically enough to give a binary classification of $\{Builiding, Foreground\}$. However, by using $N = 10$ nearest neighbors, we are able to give a probabilistic likelihood without having to do the fine-grained appearance matching over every pixel as is done with the SSD function.



(a)



(b)

Fig. 5. Comparison of solutions in a problem area around the central window of the Wolf Hall Sequence. (a) Result of timeline inpainting using SSD measure; (b) Result of timeline inpainting using the PCA-based recognition scheme. Results are comparably good, but the runtime for (b) was many times faster.

5. CONCLUSION

We have presented a novel approach to inpainting using a PCA-based recognition as opposed to exhaustive searching. We claim that representing image patches in a lower dimensional search space can vastly improve the efficiency of the search, especially in spatio-temporal analysis. We demonstrate the effectiveness of our technique in removing occlusions of building facades in image sequences using a combination of temporal and spatial inpainting.

There are several aspects of the problem that is the current focus of research. An important unaddressed image processing issue is the photometric artifacts that can be introduced due to shadows or different lighting conditions through a long sequence. Much work has been done in the face recognition community to make PCA robust to illumination. We would like to examine the adaptability of those techniques to smaller patches. We could also potentially have more speedup with better searching to find the N-nearest neighbors in the PCA eigenspace. Methods such as k-means or locality-sensitive hashing can be used to index into the feature vectors. We are also examining low-level texture-based segmentation for recovery of the building planes that will be fed to the inpainting.

6. REFERENCES

[1] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, "Calibrated, registered images of an extended urban area," *Int. J. Computer Vision*, 2003.

[2] F. van den Heuvel, *Automation in Architectural Photogrammetry; Line-Photogrammetry for the Reconstruction from Single and Multiple Images*, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2003.

[3] J. Davis, "Mosaics of scenes with moving objects," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998.

[4] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction," in *DARPA Image Understanding Workshop*, 1994.

[5] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 22–30, 1996.

[6] Reference withheld for anonymity

[7] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH*, 2000, pp. 417–424.

[8] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Processing*, vol. 13, no. 9, 2004.

[9] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video repairing: Inference of foreground and background under severe occlusion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[10] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[11] A. Efros and W. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH*, 2001.

[12] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1991.

[13] T. Deselaers, D. Keysers, and H. Ney, "Discriminative training for object recognition using image patches," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[14] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[15] R. Bornard, E. Lecan, L. Laborelli, and J. H. Chenot, "Missing data correction in still images and image sequences," in *ACM Multimedia*, 2002.

[16] Y. Ke and R. Suthanker, "A more distinctive representation for local image descriptors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[17] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers:, "Principles and practice of background maintenance," in *Proc. Int. Conf. Computer Vision*, 1999.