Removing pedestrians from Google Street View images

Arturo Flores and Serge Belongie
Department of Computer Science and Engineering
University of California, San Diego
{aflores,sjb}@cs.ucsd.edu

Abstract

Since the introduction of Google Street View, a part of Google Maps, vehicles equipped with roof-mounted mobile cameras have methodically captured street-level images of entire cities. The worldwide Street View coverage spans over 10 countries in four different continents. This service is freely available to anyone with an internet connection. While this is seen as a valuable service, the images are taken in public spaces, so they also contain license plates, faces, and other information information deemed sensitive from a privacy standpoint. Privacy concerns have been expressed by many, in particular in European countries. As a result, Google has introduced a system that automatically blurs faces in Street View images. However, many identifiable features still remain on the un-blurred person. In this paper, we propose an automatic method to remove entire pedestrians from Street View images in urban scenes. The resulting holes are filled in with data from neighboring views. A compositing method for creating "ghost-free" mosaics is used to minimize the introduction of artifacts. This yields Street View images as if the pedestrians had never been there. We present promising results on a set of images from cities around the world.

1. Introduction

Since its introduction in 2007, Google Street View (GSV) has rapidly expanded to provide street-level images of entire cities all around the world. The number and density of geo-positioned images available make this service truly unprecedented. A Street View user can wander through city streets, enabling a wide range of uses such as scouting a neighborhood, or finding specific items such as bike racks or mail boxes.

GSV has become very popular and proven to be a useful service. However, many believe it is also an invasion of individual privacy. The street-level images contain many personally identifiable features, such as faces and license plates. Some European countries have claimed Google is in breach of one or more EU privacy laws [3]. As a result, Google has introduced a sliding window based system that automatically blurs faces and license plates in street view images with a high recall rate [9]. While this goes a long way in addressing the privacy concerns, many personally identifiable features still remain on the un-blurred person. Articles of clothing, body shape, height, etc may be considered personally identifiable. Combined with the geo-positioned information, it could still be possible to identify a person despite the face blurring.

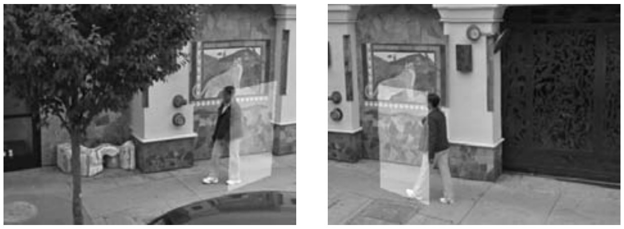


Figure 1. Two neighboring GSV images, the approximate portion of the image occluded by the pedestrian in the other view is highlighted. These highlighted regions can be warped and used to replace the pedestrian in the other view.

To address these concerns, we propose an automated method to remove entire persons from street view images. The proposed method exploits the existence of multiple views of the same scene from different angles. In urban scenes, a typical scenario is a pedestrian walking or standing on the sidewalk. In one view of the scene containing a pedestrian, part of the background is occluded by the pedestrian. However, in neighboring views of the same scene, a different part of the background is occluded by the pedestrian. Using this redundant data, it is possible to replace pixels occupied by the pedestrian with corresponding pixels from neighboring views. See figure 1 for an illustration. In urban scenes, it is also common to have a dominant planar surface as part of the image in the form of a store front or building facade. This makes it possible to relate neighboring views by a planar perspective transformation.

This method works well under the assumption that redundant data does exist. There are certain situations where

978-1-4244-7028-0/10/$26.00 ©2010 IEEE

this assumption does not hold. For example, if the pedestrian is moving in the same direction as the camera such that from the camera's perspective, the same part of the background is blocked. The proposed method can also fail if there are many pedestrians in the scene blocking the majority of the background. In this paper, we focus on removing one pedestrian from the scene.

Careful attention is paid to minimize the introduction of artifacts in the process. However, stitching artifacts already exist in many GSV images, as can be seen in figure 2. Therefore, the proposed method would be consistent with the current quality of images in GSV.

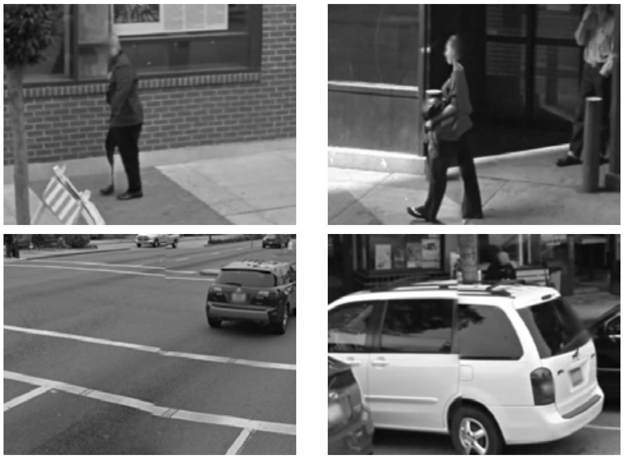


Figure 2. Unprocessed Google street view images exhibiting stitching artifacts. Images are from Berkeley, CA and New York, NY.

In section 2 we first review methods related to object removal. In section 3 we describe the proposed method in detail. In section 4 we describe the data used to qualitatively evaluate the proposed method. In section 5 we show promising results on the evaluation dataset.

2. Related works

Bohm et al [4] proposed a multi-image fusion technique for occlusion free facade texturing. This method uses a technique similar to background subtraction. In a set of registered images, corresponding pixels are clustered based on their RGB values, and outliers are discarded. The pixel with the most "consensus votes" is selected as the background pixel. An example is shown where images taken from 3 different locations of a building facade occluded by a statue. After applying their method, the occluding statue is removed yielding an unobstructed view of the facade. However, this method requires at least 3 images to work, and a relatively small baseline. In Google street view images, the baseline between neighboring views can range between 10-15 meters. This baseline was determined experimentally using Google Maps API [1]. The wide baseline makes it difficult to find correspondences between three consecutive views.

Fruh et al [10] presented an automated method capable of producing textured 3D models of urban environments for photo-realistic walk-throughs. Their data acquisition method is similar to Google's in that a vehicle is equipped with a camera and driven through city streets under normal traffic conditions. In addition to the camera, the vehicle is also equipped with inexpensive laser scanners. This setup provides them not only with images, but also with 3D point clouds. They then apply histogram analysis of pixel depths to identify and remove pixels corresponding to foreground objects. Holes are filled in with various methods such as interpolation and cut-and-paste. Google has confirmed that 3D data is also being collected [7], but this is still in an experimental stage.

In [2], Avidan proposed a method for automatic image resizing based on a technique called *seam carving*. Seam carving works by selecting paths of low energy pixels (seams) and removing or inserting pixels in these locations. The magnitude of the gradient is used as the energy function. Object removal from a single image is presented as an application of this technique. This works by manually indicating the object to be removed, then seams that pass through the object are removed until the object is gone from the image. The object removal results are virtually imperceptible, though it has the effect of altering the contents of the image by removing and inserting pixels. The method we propose uses images from multiple views to remove the pedestrian as if it had never been there. The general content of the image remains unaltered.

3. Proposed method

As mentioned earlier, urban scenes often contain a dominant planar surface, which makes it possible to relate two views by a planar perspective transformation. The first step is to compute the homography relating neighboring views I_1 and I_2. To do this, we first extract SIFT [12] descriptors from both views and match them using the algorithm proposed by Lowe. Given the putative set of correspondences, RANSAC [8] is used to exclude outliers and compute the homography. In order to minimize the introduction of artifacts in subsequent steps, a second round of RANSAC with a tighter threshold is run to further refine the homography. Figure 3 shows the results of this step for a pair of images.

Once the homographies are computed, we run the pedestrian detection algorithm by Liebe [11] to extract bounding boxes B_1 and B_2, as well as probability maps M_1 and M_2 from each view, see figure 4 for an example. Leibe's pedestrian detection algorithm automatically performs multi-scale search. The parameters $minScale$ and $maxScale$ determine the recognition search scale range. Using the codebooks from [11], we set $minScale = .2$ and

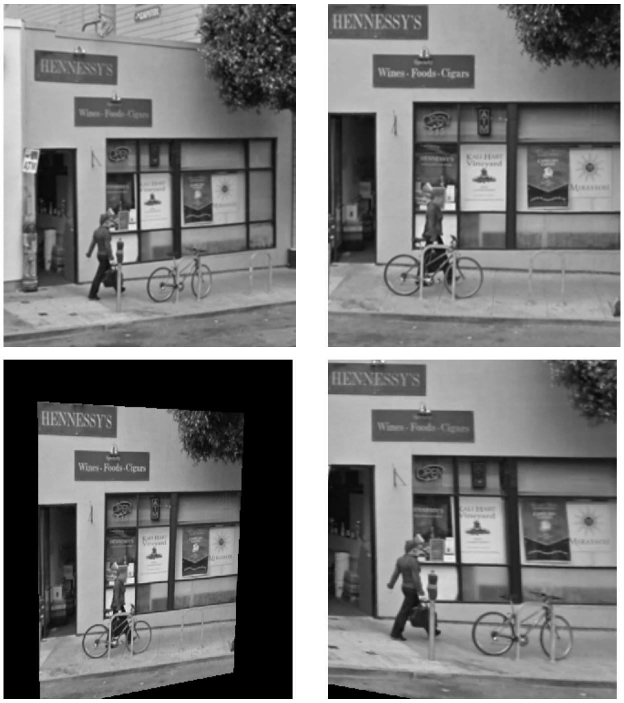


Figure 3. (top row) Two neighboring views of a scene. (bottom row) The other view warped by homography relating the two views.

$maxScale = 3$ to account for the wide range of distances between the camera and pedestrian in GSV images. Using the homography computed in the previous step, the bounding boxes are warped resulting in $\hat{B}_1$ (bounding box in I_1 warped into I_2) and $\hat{B}_2$, similarly for the probability maps.

In figure 4, the bounding box does not include the entire person, part of the foot is not contained in the bounding box. This happens frequently enough to require some attention. A simple solution is to enlarge the bounding box by a relative factor. In general, this produces acceptable results given the compositing method used in the following step.

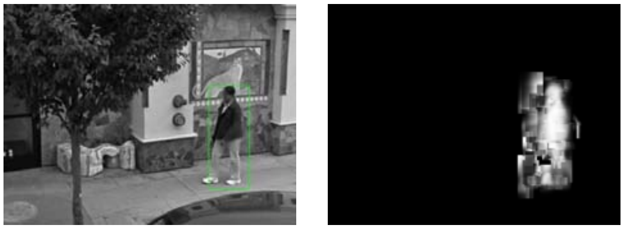


Figure 4. Pedestrian detection algorithm results: (left) Bounding box and (right)) per-pixel probability map

Assume we are removing the pedestrian from I_1. At this point, we could use the homography to replace pixels from I_1 inside bounding box B_1 with corresponding pixels from I_2. However, in certain situations, the warped bounding box $\hat{B}_1$ overlaps with B_2, this is illustrated in figure 5. In these situations, we would be replacing pedestrian pixels in I_1 with pedestrian pixels from I_2. This undesirable effect is mitigated by using a compositing method proposed by Davis [5] in the overlap region.

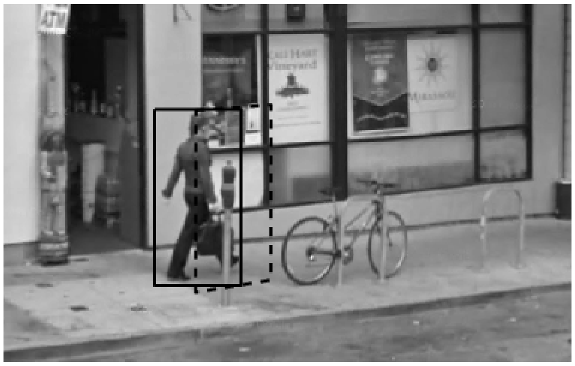


Figure 5. Illustrative example of a B_1 (solid line) overlapping with $\hat{B}_2$ (dashed line) caused by the pedestrian's walking velocity.

In [5], Davis proposed a compositing method to create image mosaics of scenes containing moving objects. A *relative difference image*, defined as $d = abs(I_1 - \hat{I}_2)/max(I_1 - \hat{I}_2)$, provides a measure of similarity on pixels in the overlapping region. A dividing boundary following a path of low intensity in the relative difference image is used to minimize the discontinuities in the final mosaic. A related method has been used for other purposes including texture synthesis from image patches [6] and image resizing [2]. For our purposes, this boundary has the desirable effect of minimizing discontinuities and stitching artifacts, as well as minimizing the number of pedestrian pixels in I_1 replaced with corresponding pedestrian pixels from I_2. See figure 6 for an illustrative example. As in [6], the shortest low intensity path is computed using dynamic programming. Assuming a vertical cut, suppose the overlap region d is of size n rows by m columns. We initialize $d_{1,j} = 0$ and then traverse $d(i = 2..n)$ and compute the minimum intensity path E for all paths by:

$$E_{i,j} = d_{i,j} + min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}). \tag{1}$$

The minimum value of the last row in E indicates the endpoint of the lowest intensity vertical path along d. We can then trace back to find this path.

Here it is still unclear which side of the boundary we should be taking pixels from. Depending on the direction and speed the pedestrian was moving, we may want to take pixels from the left or right side of the boundary. To resolve this ambiguity, we use the warped probability map $\hat{M}_1$ to decide which side of the boundary to take pixels from. The side maximizing the sum of the probability map, i.e. the side with most pedestrian pixels from I_1 (i.e., pixels we will be replacing), is chosen. See figure 7 for an example.

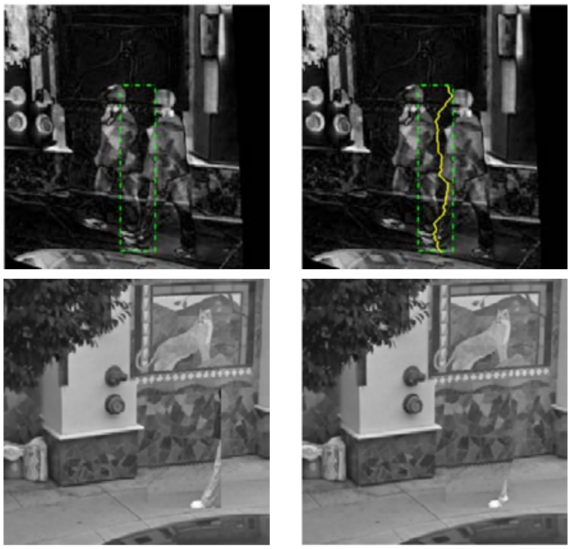


Figure 6. (top left) Relative difference image $d = abs(I_1 - \hat{I}_2)/max(I_1 - \hat{I}_2)$ with bounding box overlap. (top right) Minimum error boundary cut in overlap region. (bottom left) Pedestrian removed without using the minimum error boundary cut. (bottom right) Pedestrian removed using the minimum error boundary cut.

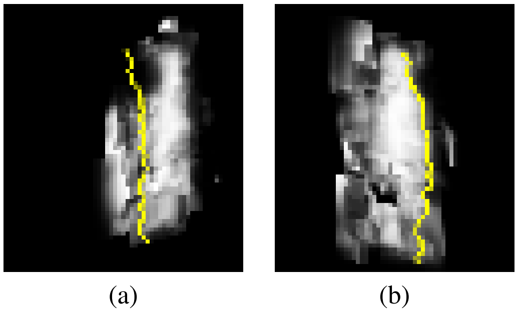


Figure 7. Warped probability maps used to decide which side of the boundary to take pixels from. Pixels are taken from the right in (a), from the left in (b)

A summary of the proposed method follows. Computation time is provided in parenthesis for each step on a 1.83 GHz Pentium CPU. With the exception of the pedestrian detection algorithm, all steps are implemented in Matlab.

1. Compute homographies between two views using SIFT and RANSAC ($3 - 5$s).

2. Extract bounding boxes and probability maps from both views using Leibe's pedestrian detection algorithm ($20 - 25$s).

3. Warp pedestrian bounding and heat maps using homography from step 1 ($200 - 500$ms).

4. Use compositing method proposed by Davis to obtain a dividing boundary in overlap region ($50 - 100$ms).

5. Warp probability map and use it to decide which side of the boundary to take pixels from ($200 - 500$ms).

6. Replace pixels inside the bounding box with corresponding pixels from the other view, using the boundary from step 4 ($300 - 500$ms).

4. Data

We manually identified and selected images from various cities including but not limited to San Francisco, New York, Berkeley, Boston, and London. Images are cropped to exclude overlays added by Google. We focus on images with a single pedestrian. We use this dataset for qualitative evaluation purposes only.

5. Results

See figure 8 for an example of results of our pedestrian removal system. Here, the pedestrian is completely removed and there are minimal stitching artifacts, though the glass door from the other view has a different shade.

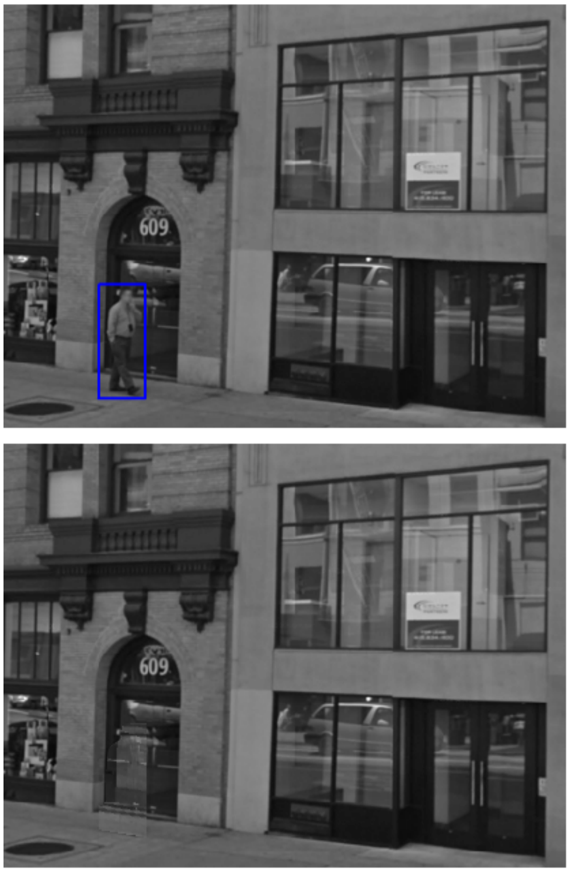


Figure 8. Pedestrian removal results.

Figure 9 contains a gallery of results. In figure 9b, a small portion of the pedestrian from the other view is brought in, but the majority of the pedestrian is gone. An artifact was introduced here near the pillar because it lies outside of the facade's planar surface. In figure 9c, there are multiple occluding objects in the scene, such as the bicycle. As a result, part of the bicycle is copied in place of the pedestrian. In figure 9d, the pedestrian is removed in a portion where the planar constraint is not satisfied. In spite of this, the results are still reasonable.

In figure 9g, the portion of the filled in corresponding to the building lines up well with the rest of the image, but the portion corresponding to the ground does not. Incorporating a ground plane could improve the results in this case. A situation where the method fails can be seen in figure 9l. Here the pedestrian is not moving and is very close to the facade. Figure 9k also shows the case where pixels from the car to the right are used in to replace the pedestrian.

6. Conclusion and future work

We have presented an automated method to remove pedestrians from GSV images. The proposed method works well in urban scenes where a dominant planar surface is typically present. Aside from removing the pedestrians from the image, the general structure and content of the scene remains unchanged. We presented promising qualitative results on a set of images from cities around the world. Pedestrians are removed from Street View images leaving an unobstructed view of the background. This is a step beyond the face blurring Google already does and may help to alleviate privacy concerns regarding GSV.

The proposed method may not work well in general outdoor scenes. Other situations where the proposed method may fail are: scenes containing many pedestrians, a stationary pedestrian too close to the building facade, the pedestrian moving in the same direction as the GSV vehicle and with the right speed.

The proposed method makes use of only two images. It may be possible to improve the results by using three images. In our experiments, establishing correspondences spanning three consecutive views was difficult due to the wide baseline. Other feature detectors or matching methods may make this possible. With three views, it would be possible to use a voting method similar to [4]. With more than two images, it may also be possible to use local image registration as in [14]. This is a subject for future research.

For additional future work, we will investigate ways to handle situations where the pedestrian is too close to the building facade, or when too many pedestrians are present. Possibilities include using texture synthesis [6], interpolation, inpainting [13], and copy-and-paste [10]. We will also investigate incorporating multiple planar surfaces (such as ground plane) to improve the results.

7. Acknowledgements

The authors would like to thank the Winter 2010 CSE 190-A class for the useful discussions and helpful feedback. This work was supported in part by DARPA Grant NBCH1080007 subaward Z931303 and NSF CAREER Grant #0448615.

References

[1] Google Maps API Reference. `http://code.google.com/apis/maps/documentation/reference.html`.

[2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, 26(3):10, 2007.

[3] S. Bodoni. Google street view may breach EU law, officials say. `http://www.bloomberg.com/apps/news?pid=20601085&sid=a2Tbh.fOrFB0`, Feb 2010.

[4] J. Bohm. Multi-image fusion for occlusion-free façade texturing. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(5):867–872, 2004.

[5] J. Davis. Mosaics of scenes with moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 354–360, 1998.

[6] A. Efros and W. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of SIGGRAPH 2001*, pages 341–346, 2001.

[7] D. Filip. Introducing smart navigation in street view: double-click to go (anywhere!). `http://google-latlong.blogspot.com/2009/06/introducing-smart-navigation-in-street.html`, June 2009.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[9] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent. Large-scale Privacy Protection in Google Street View. In *International Conference on Computer Vision*, 2009.

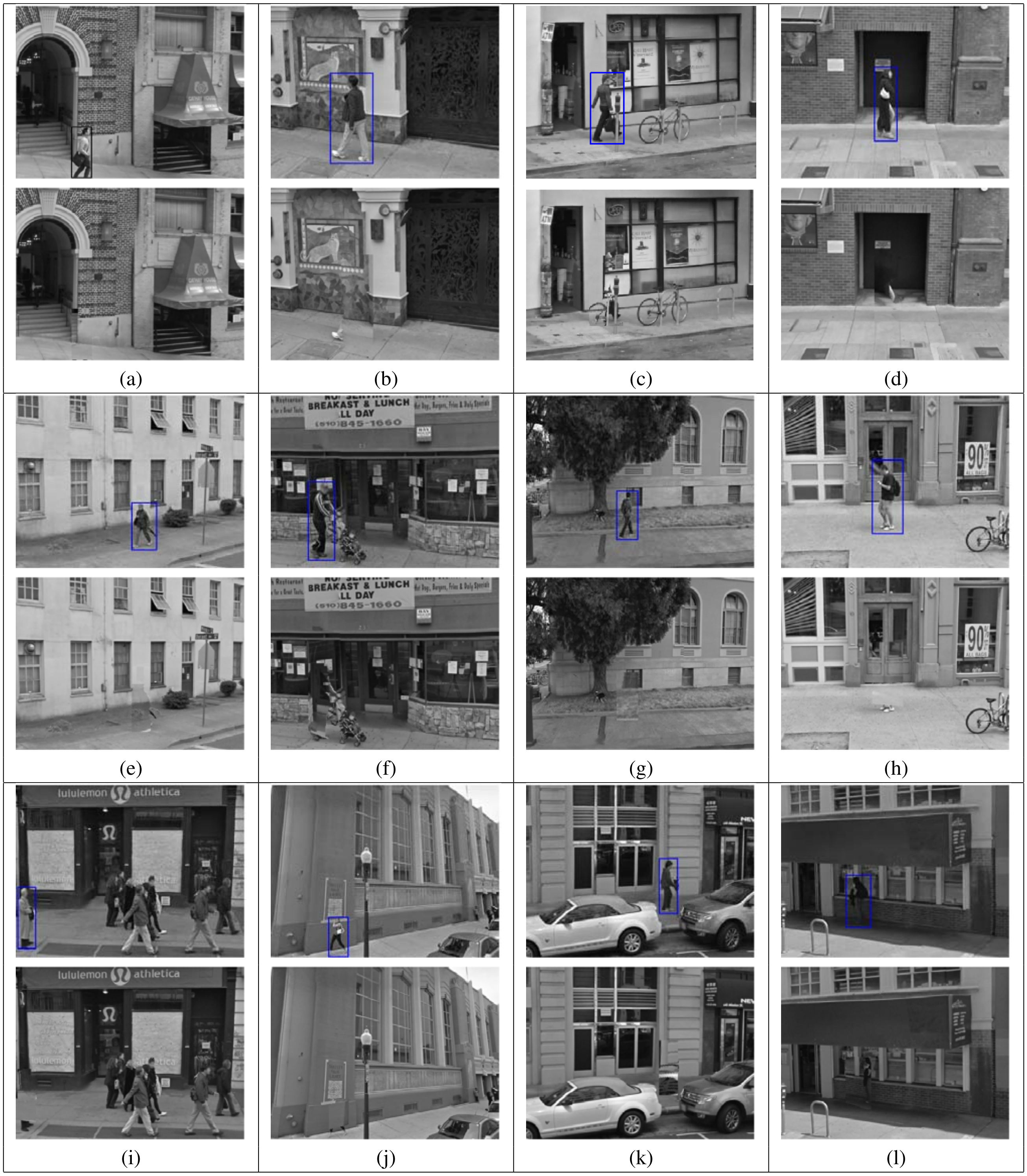
[10] C. Fruh and A. Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60(1):5–24, 2004.

[11] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1):259–289, 2008.

[12] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[13] J. Shen. Inpainting and the fundamental problem of image processing. *SIAM news*, 36(5):1–4, 2003.

[14] H. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings IEEE CVPR*, pages 953–958, 1998.


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 9. Gallery of results. Original images on top, pedestrians removed on bottom.