AUTOMATIC OCCLUSION REMOVAL FROM MINIMUM NUMBER OF IMAGES

Cormac Herley

Microsoft Research
One Microsoft Way
Redmond, WA 98052

ABSTRACT

We examine the problem of combining several images to remove occlusions that occur in one or more of a sequence of images. A sequence of digital camera images of a photo-worthy landmark may be occluded by passers-by walking between the photographer and the scene. As they move they will occlude different regions in each image of the sequence. It is obvious that if two or more images coincide at each location an unoccluded image can be formed. We show that this is not necessary: in fact so long as at least one image is unoccluded at each location an unoccluded image can be formed automatically. This greatly eases the conditions under which an unoccluded image can be formed. We detail the algorithm and show results of occlusion removal.

1. INTRODUCTION

In seeking to take a photograph of a famous monument or scenic spot many people will have experienced the difficulty that passers-by wander into the frame of the shot they wish to take. By the time one person has moved out another will often have moved in, making it difficult and time-consuming to get a picture of the desired scene without occlusion. An example is shown in Figure 1 where several photos of the same scene are each blocked by passers-by. Since the passers-by move, and occlude different parts of the scene in each of the photos it is natural to wonder whether an unoccluded photo could be formed by combining several occluded ones. That is, each of the photos in Figure 1 has some data that is "good" (*i.e.* the unobstructed view of the arch) and some that is "bad" (*i.e.* the passer(s)-by in that particular photo). Is it possible to combine the "good" data from several photos and get a single unobstructed view of the arch ? Thus to cover the obstruction in the first photo we might copy data from the second, since the obstructions in these two photos cover different parts of the scene.

Call the images $I_0(i,j), I_1(i,j), \cdots I_{N-1}(i,j)$. Assume for the moment that they are perfectly registered and there is no image to image noise, so that (apart from obstructions) they coincide. That is $I_m(i,j) = I_n(i,j) \forall m,n$ unless either I_m or I_n is occluded at that location. We make this unrealistic assumption only to simplify the analysis. We explore the necessary registration phase using techniques from the image stitching literature [4] in Section 3.2. Clearly, if two or more of the $I(i,j)$ have the same value at a particular location that value is background scene rather than an obstruction (assuming that the probability of two obstructions having the same value is low). So define

$$U(i,j) = \begin{cases} I_m(i,j) & \text{if } I_m(i,j) = I_n(i,j) \text{ some } m,n \\ 0 & \text{Otherwise.} \end{cases} \tag{1}$$

We call this the consensus image, since it acquires the value of any two images that agree at a location.

This allows a simple way to combine the images in a sequence to automatically remove occlusions, so long as at least two images coincide at each location. Image sequences such as shown in Figure 1 cannot be repaired in this way however. Though there is no overlap in the obstructed sections of the two photos (*i.e.* there is enough "good" data) we cannot form a single unobstructed view of the background.

Call the unobstructed image I_x, and define $R_m = Sgn(I_x - I_m)$, that is R_m is the image that has value 0 where I_m coincides with the unobstructed image, and is one everywhere else. A simple criterion to be able to reconstruct I_x is then to have that

$$\sum_{k=0}^{N-1} R_k(i,j) < N-2 \;\; \forall i,j. \tag{2}$$

In words: if at least two images agree at each location we can reconstruct. This is sufficient but by no means necessary. In fact we will show, that in almost all cases we can reconstruct I_x so long as

$$\sum_{k=0}^{N-1} R_k(i,j) < N-1 \;\; \forall i,j. \tag{3}$$

That is we can reconstruct so long as a single image is unoccluded at each location, subject only to some very minor constraints on the nature of the occlusions. The algorithm

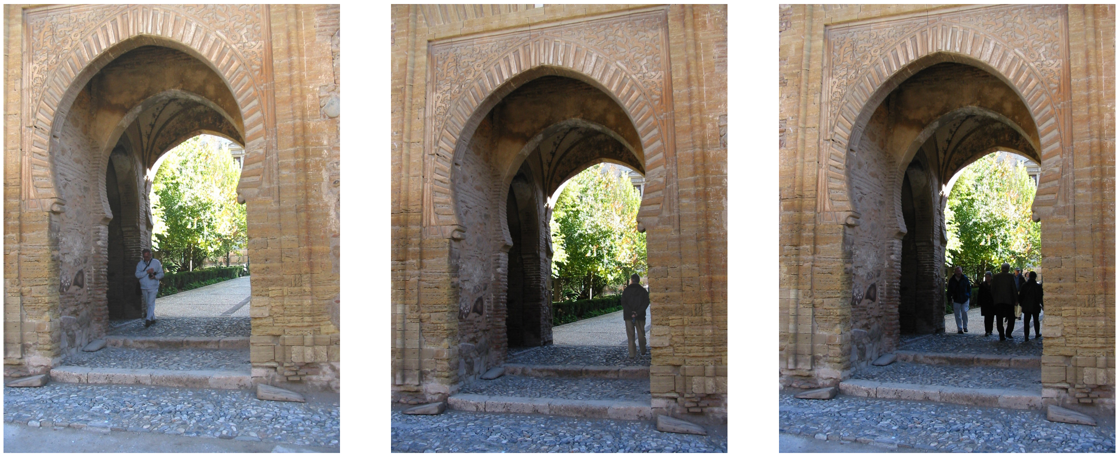

Fig. 1. Image sequence with occlusions. Several photos of the same desired shot are occluded by passers-by. In each photo a different section is occluded. We desire to form an unobstructed image of the scene from the minimum number of photos.

is fully automatic. In the next sections we prove the looser criterion, explain the algorithm and give examples.

The use of multiple copies of a signal to product an improved signal has a long and rich history. The particular case of use of image sequences for occlusion removal has also been addressed. See, for example [5, 3]. The novel contribution of this work is ability to remove occlusions using a minimum number of images.

2. NATURE OF OCCLUSIONS

As was the case in Figure 1 we assume that occlusions generally occur because of a person or object moving in front of the desired scene. Thus occlusions are connected closed sets. Without loss of generality assume that an image in the sequence I_m differs from the unknown target image I_x only over the union of several closed connected sets of pixels. An example is shown in Figure 2 (a) where two instances of the same scene are obstructed by an occlusion.

Clearly the consensus image shown in Figure 2 (b), U, equals I_x in locations where two or more of the images in the sequence agree and is zero elsewhere. The zero regions of U form closed connected sets, as shown in Figure 2 (b). At this stage U has several holes. Call these sets S_p for $p = 0, 1, \cdots, M-1$ assuming there are M of them (in our example $M = 2$). Clearly

$$\sum_{(i,j)\in S_p} U(i,j) = 0, \text{ for } p = 0, 1, \cdots, M-1.$$

We will also need to define the boundary of the S_p. Call B_p the set of all points in S_p that have at least one neighbor not in S_p, and call $B_p^{'}$ the set of all points not in S_p that have at least one neighbor in S_p. These can be though of as the sets of points just inside and just outside the connected set S_p respectively. The connected sets S_p are the ”holes” in the data of image U that need to be filled in. We try to identify which of the I_m has data most likely to perform the fill in.

From each of the I_m form a new image $I_m^{'}$ that agrees with the consensus image U except where U is zero. In the example of Figure 2 $I_m = I_m^{'}$, since the occlusions do not overlap, but for $M > 2$ this need not be the case. In general each of the $I_m^{'}$ will have an occlusion, over an area no larger than the I_m. Now consider the holes in U. Call the largest (the large black area in the right of Figure 2 (b)) S_0. First, the area covered by S_0 corresponds to an occlusion in $I_1^{'}$ but is unoccluded in $I_0^{'}$. Similarly the area covered by S_1 (the next largest hole) corresponds to an occlusion in both $I_0^{'}$ but is unoccluded in $I_1^{'}$. A user could manually copy the data from $I^{'}1$ to cover S_1. However in this simple example we deal only with two occlusions, in general we seek an automatic solution since large numbers of occlusions are likely. It remains to show how the best image to cover any particular hole can be determined *automatically.* Our approach is to observe that when an occlusion occurs there is generally a discontinuity all around the boundary of the occlusion. We will use this fact to determine which of the I_m is most likely to have “good” data.

For each connected set S_p we define

$$b_{mp} = \frac{1}{\#(B_p)} \sum_{(i,j)\in B_p} I_m(i,j) - \frac{1}{\#(B_p^{\prime})} \sum_{(i,j)\in B_p^{\prime}} I_m(i,j),$$

where $\#(B_p)$ is the number of pixels in B_p. This can be used as a crude measure of the discontinuity across the boundary of the set S_p in the image I_m. Assuming that (3) holds data from one or more of the images I_m agrees with I_x over the set S_p. We will show below that in fact data from a single one of the images will cover S_p rather than data from several.

	p=0	p=1
m=0	9	73
m=1	84	7

Table 1. Values of b_{mp} for the example shown in Figure 2. Observe that $b_{00} \ll b_{01}$ which indicates that data from $I_0^{'}$ can be used to cover S_1. Similarly $b_{11} \ll b_{10}$ which indicates that data from $I_1^{'}$ can be used to cover S_0.

Consider the simple case of Figure 2: there are two images and U will have $M = 2$ connected sets S_p. We calculate b_{mp} and tabulate as shown in Table 1. Observe that b_{00} is small relative to b_{01}. This indicates a far stronger discontinuity across the boundary of the set S_0 in the images $I_1^{'}$ than in image $I_0^{'}$. Similarly b_{11} is small relative to b_{10}. In general a small b_{mp} indicates that data from I_m can be used to cover the hole S_p.

3. AUTOMATIC OCCLUSION REMOVAL ALGORITHM

Having demonstrated an example where only two images are involved we now give the general algorithm. We propose the following algorithm:

1. Calculate $U_0(i,j) = U(i,j)$ as defined in (1).
2. Calculate
$$I_m^{'}(i,j) = \begin{cases} U(i,j) & \text{for } U(i,j) \neq 0 \\ I_m(i,j) & \text{Otherwise.} \end{cases}$$
3. Find the connected components $S_p, p = 0, 1, \cdots M-1$ having value zero in $U(i,j)$. Find the inner and outer boundaries of these sets B_p and $B_p^{'}$.
4. For each connected component S_p and for each image $I_m^{'}$ calculate
$$b_{mp} = \frac{1}{\#(B_{mp})} \sum_{(i,j)\in B_p} I_m(i,j) - \frac{1}{\#(B'_{mp})} \sum_{(i,j)\in B'_p} I_m(i,j).$$
5. Find the image q such that b_{qp} is minimum and form
$$U_p(i,j) = U_{p-1}(i,j) + \sum_{(i,j)\in S_p} I_q(i,j).$$

Finding connected sets of a particular value in an image can be carried out very efficiently [1]. There are various approaches to efficiently determining the boundary of a connected set [2].

3.1. Completeness of the Algorithm

We now show that if an image sequence satisfies (3) the above algorithm produces $U_{M-1} = I_x$. Recall that by assumption (3) holds. Hence each connected set of value zero in U can be filled with data from one or more of the $I_m^{'}$. Our algorithm assumes that each connected set could be filled with data from a *single* connected set, and hence the problem simplified to determining which image was best. Suppose this is not the case: a set S_p in U must be patched partly from $I_m^{'}$ and partly from $I_n^{'}$. That is $I_m^{'}$ is occluded over part of S_p, call this subset A, and unoccluded over the rest, call this subset B. It then must be that $I_n^{'}$ is occluded over B and unoccluded over A (they cannot both be unoccluded since then they would agree and those points would not be in the set S_p). Hence the boundary between the two subsets A and B forms a portion of the boundary of the occlusions *in both* $I_m^{'}$ and $I_n^{'}$. But this violates the assumption that our occlusion are independent objects. Hence $U_{M-1} = I_x$.

3.2. Registration and Blending

The algorithm above assumed that all of the images in the sequence were perfectly registered. Even when a tripod is used there will be enough image to image variation to make this unrealistic. When the camera is hand held it is to be expected that rotation, tilt, framing and white balance will all differ between the images. Observe, for example, that the three images in Figure 1 each present slightly different framings of the arch (even ignoring the occlusions). Thus prior to processing we register all of the images relative to one. Fortunately the problem of registering like images has been addressed by those working on the problem of stitching images for panoramas and mosaics [4].

Blending of that data from one of the images to cover and occluded region is also very important. It is important that this be carried out with care, but space does not allow a detailed treatment here. The interested reader is referred to [2, 4] for good approaches.

4. RESULTS AND CONCLUSION

We have implemented the occlusion removal algorithm and tested on a variety of image sequences. All of the occlusions were formed by people or objects obscuring the desired view. Image sequences were taken with a digital camera which was held by hand. Registration was performed before processing. Figure 3 is representative of the results.

A key advantage of our algorithm is the ability to remove occlusions with a minimum number of images. For example the images of Figure 2 could not be improved using previous methods. Cases where our method can remove occlusions while the consensus method cannot are common. In a set of 13 image sequences, each comprising from 3 to 7

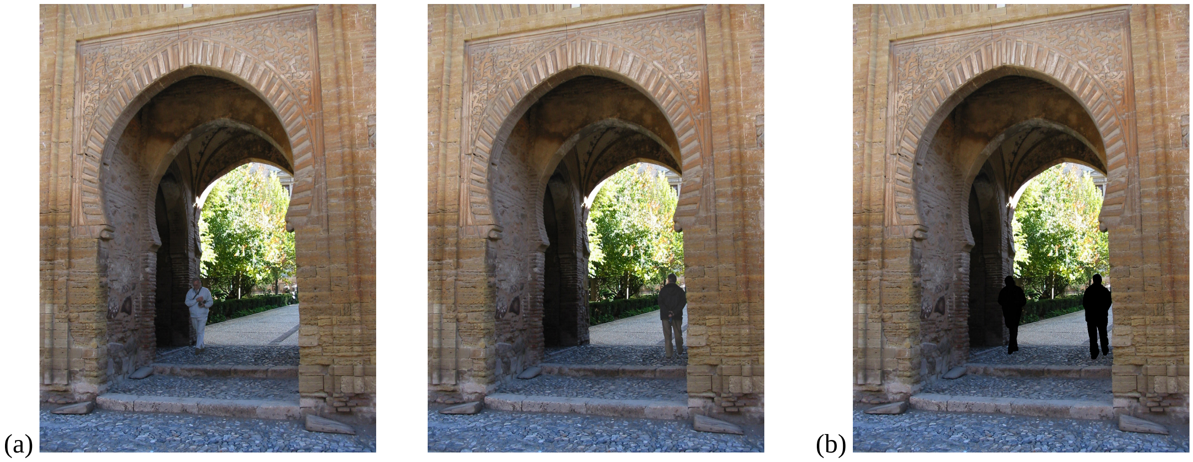

Fig. 2. Image sequence with occlusions. (a) Two images, I_0, I_1 each with different occlusions. (Note that in contrast to the images in Figure 1 these are registered). (b) The consensus image U, as defined in (1), which is non-zero at locations where any two or more images agree and zero elsewhere. The zero locations of U (the black regions) form connected sets, which are the "holes" that must be filled in. Here S_0 is on the right and S_1 on the left.

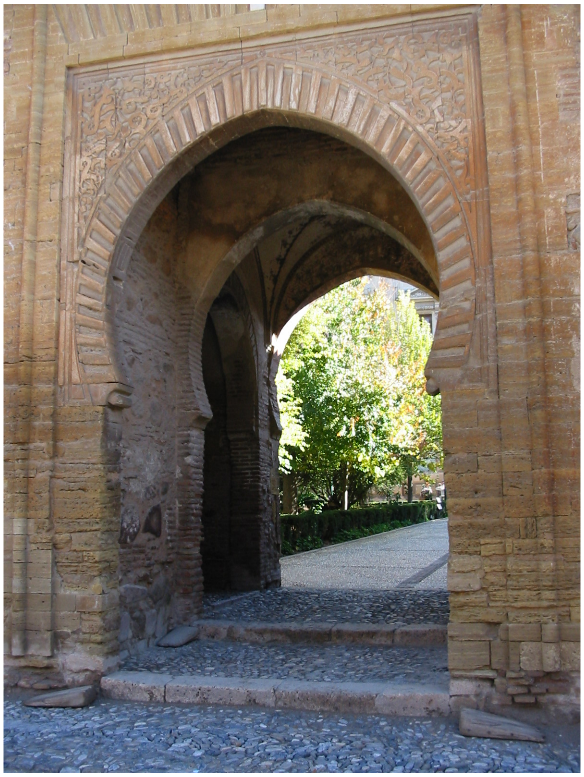

Fig. 3. Unoccluded image: the result of processing the two images from Figure 2 (a).

images our method produced an unoccluded image in all 13 cases, while the consensus method did so in only 4. Space obviously will not allow an exhaustive presentation of results since each sequence requires showing the M images of the sequence and the result.

5. REFERENCES

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw Hill, 1990.

[2] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[3] A. Kokaram, B. Collis, and S. Robinson. A bayesian framework for recursive object removal in movie post-production. *ICIP 2003*.

[4] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics*, 31:251–258, 1997.

[5] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *The IEEE Transactions on Image Processing*, 3(5):625–638, Sept. 1994.