

# The patch transform and its applications to image editing

## Supplemental Material

Taeg Sang Cho\*, Moshe Butman†, Shai Avidan‡, William T. Freeman\*

\* Massachusetts Institute of Technology

† Bar-Ilan University

‡ Adobe Systems Inc.

taegsang@mit.edu, butmoshe@cs.biu.ac.il, avidan@adobe.com, billf@mit.edu

In the CVPR supplemental material, we provide more image editing examples along with a video that shows the image reconstruction process. In section 1, we experimentally show that the propose framework is quite insensitive to the exact location of bounding boxes. Then, in section 2 and section 3, we show a larger version of images in the paper, and more image reconstruction results with different images, respectively.

### 1. The bounding box location sensitivity analysis

Examples shown in the paper don't convey how sensitive the algorithm is to the changes of bounding box locations. Here, we experimentally analyze the sensitivity of the algorithm with respect to the changing location of bounding boxes. We use the images in Figure 1 to generate a collage.

We first specify our initial condition as shown in Figure 2. Then, we will fix the man on the right, and move the mountain top vertically and laterally. The sequence in which we will move the mountain bounding box is shown in Figure 2(b). The bounding box for the mountain tip in Figure 2 corresponds to location 1, the bounding box for the mountain tip in the second example corresponds to location 2 (one image node to the left of location 1), and so on. Such a sequence of different user constraints will generate different images to satisfy them.

The reconstruction result starts from Figure 3, and proceeds in the order specified by Figure 2(b). In each figure, we show a pair of images. One on the left is the reconstructed image with the corresponding user constraint, and the image on the right shows the origin of the patch used at a given position in the resulting image. The yellow region denotes that patches from Figure 1(a) are used, and the green region denotes that patches from Figure 1(b) are used. As the examples show, the algorithm is quite robust to the changes of the location of bounding boxes. With different user constraints, different, but still visually pleasing, images can be generated. Therefore, the user does not have to go through too many trial-and-error to get a pleasing output.



(a)

(b)

Figure 1. The original images from which a photomontage will be generated.



(a)

5	4	6
2	1	3
8	7	9

(b)

Figure 2. (a) The user constraint that corresponds to 1 in (b). (b) The sequence in which the examples will be shown. In this experiment, the man on the right is fixed, and only the mountain top is moved around. 1 corresponds to the mountain tip location as shown in (a). 2 corresponds to moving the mountain tip one node to the left of 1. 3 corresponds to moving the mountain tip one node to the right of 1. 4 corresponds to moving the mountain tip one node to the top of 1 etc...



Figure 3. Mountain tip location 1



Figure 4. Mountain tip location 2



Figure 5. Mountain tip location 3



Figure 6. Mountain tip location 4



Figure 7. Mountain tip location 5



Figure 8. Mountain tip location 6



Figure 9. Mountain tip location 7

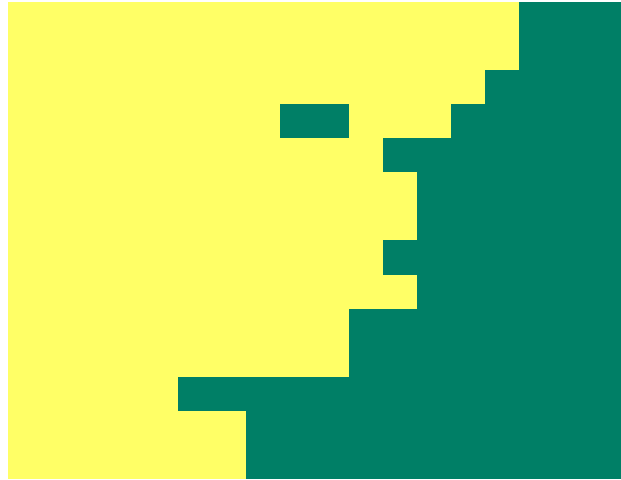


Figure 10. Mountain tip location 8

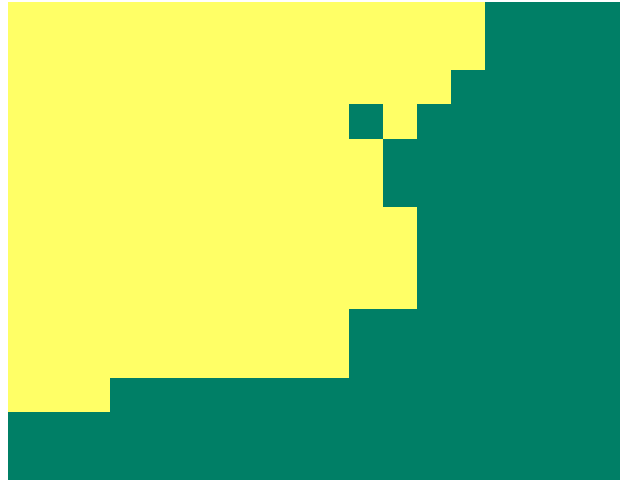


Figure 11. Mountain tip location 9

## 2. Blown-up version of images in the paper

In this section, we show a larger version of images included in the paper.

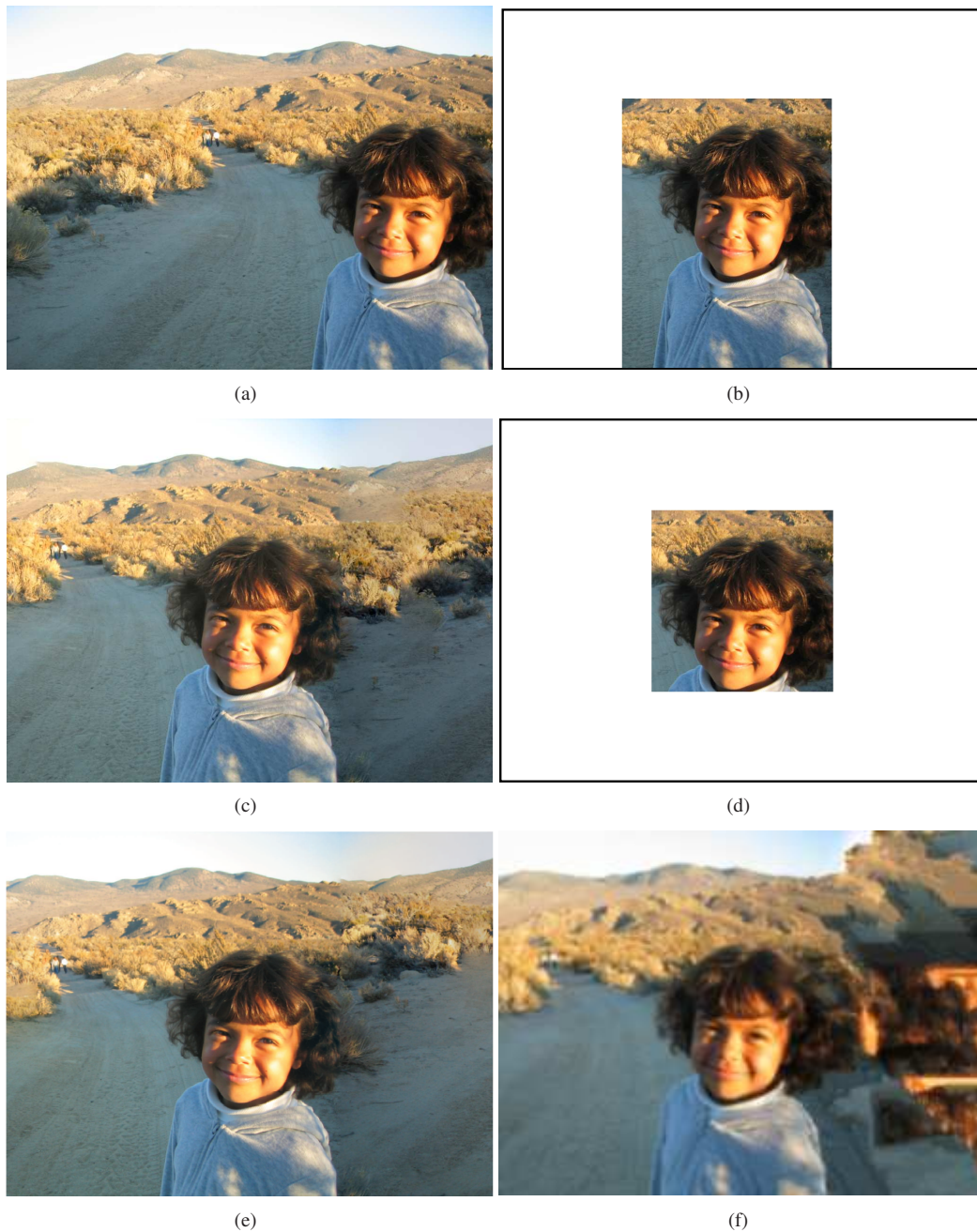


Figure 12. This example illustrates how the patch transform framework can be used to recenter a region / object of interest. (a) The original image. (b)(c) The inverse patch transform result. Notice that the overall structure and context is preserved. (d)(e) Another inverse patch transform result. This figure shows that the proposed framework is insensitive to the size of the bounding box. (f) Using the same constraint as that of (b), a texture synthesis method by Efros and Leung is used to generate a new image. Notice that the hair from the girl propagates into the bush, creating artifacts.

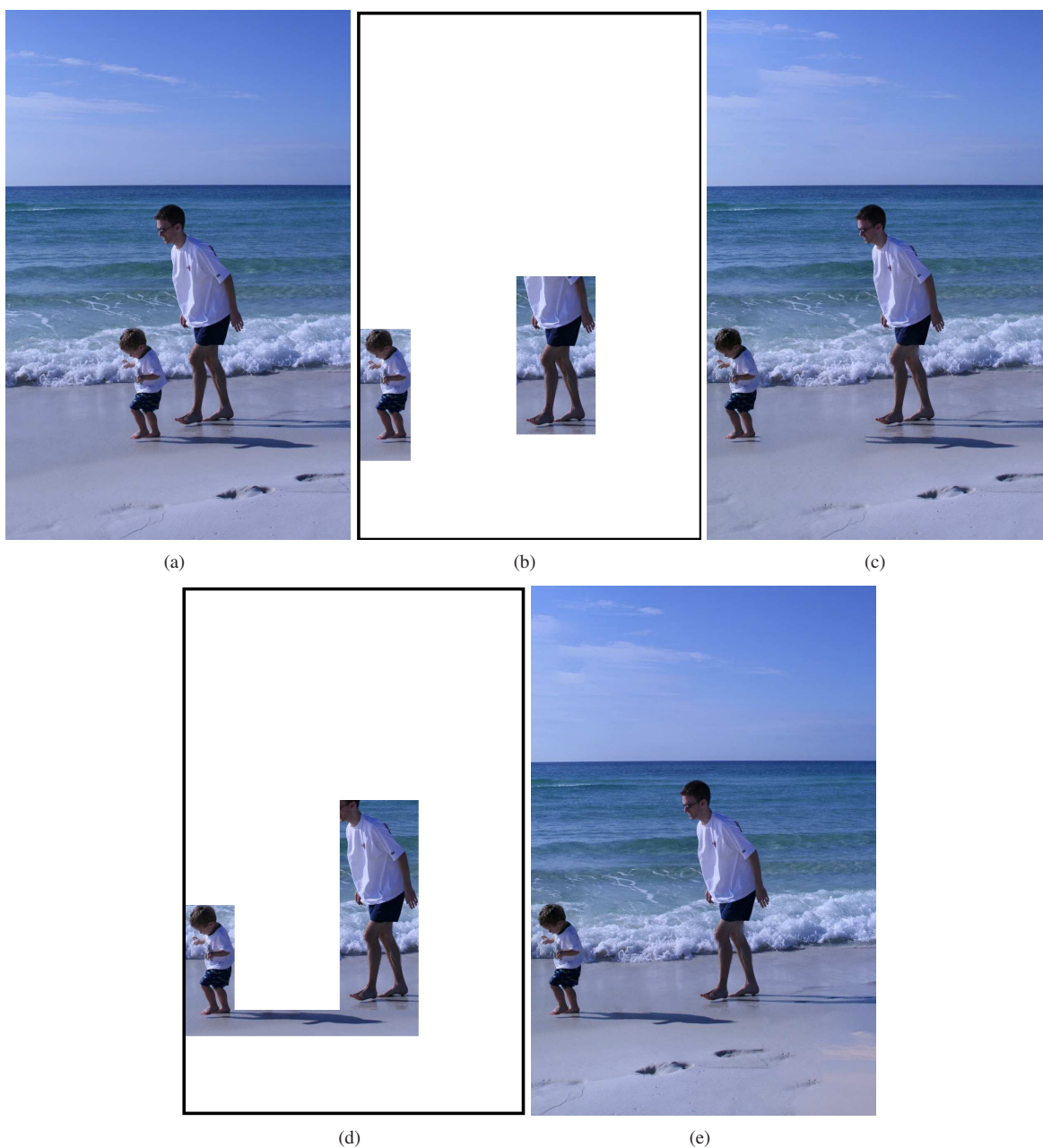
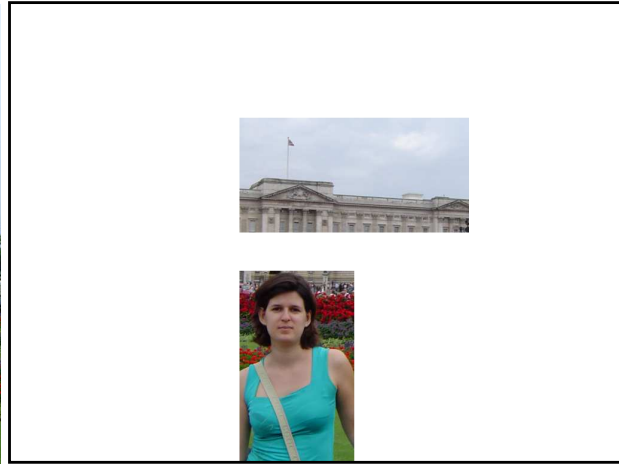


Figure 13. This example shows that the proposed framework can be used to change the relative position of two, or multiple, objects in the image. (a) The original image. (b)(c) A user specified constraint that the child should be place further away from his father. The patches are reorganized such that the overall scene context is preserved while meeting the user constraint. Notice, however, that the shadow didn't follow the kid since the compatibility the shadow is more tightly bonded to the father with larger boundary being shared. (d)(e) A user specified constraint that the child and the shadow should be place further away from his father. This time, the algorithm rearranges the patches such that the child's shadow is properly connected.



(a)



(b)



(c)



(d)

Figure 14. This example verifies that the proposed framework can still work well in the presence of complex background. Again, we want to recenter the woman such that the center of the building aligns with the woman. (a) The original image. (b) The user specified that the woman and the building should be aligned. (c) The patch transform result without Poisson blending. While the algorithm did not move the building, the algorithm reshuffled the patches in the garden to accommodate the changes in the woman's position. (d) The patch transform result with Poisson blending. We can see some color bleeding from the flowers into the building.





(a)

(b)



(c)

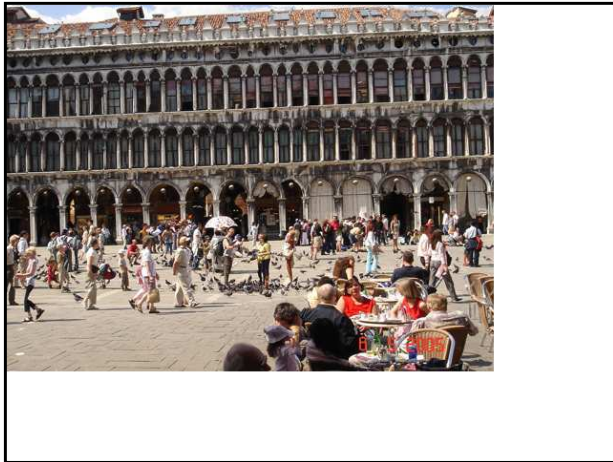
Figure 15. This example shows how the proposed framework can be used to manipulate the patch statistics of an image. In the reconstructed image, the tree is specified to stand on the right side of the image, as shown by the red bounding box. (a) is the original image. (b) is the patch transform result with a constraint to use less sky patches. (c) is the patch transform result with a constraint to use less cloud patches.



(a)



(b)



(c)



(d)

Figure 16. In this example, the original image shown in (a) is resized such that the width and height of the output image is 80% of the original image. (b) The reconstructed image from the patch transform framework. (c) The retargeting result using *Seam Carving*. While *Seam Carving* preserves locally salient structures well, our work preserves the global context of the image through local evidence. (d) A mere crop of the original image to meet the user request.



Figure 17. In this example, we collage two images shown in (a) and (b). This example can arise in real world when it's hard to capture both the background and the person at the same time. (c) The user specified that part of the Mount McKinley be placed next to a person shown on the right. (d) The patch transform result without Poisson blending. (e) The patch transform result with Poisson blending. Although we see some color bleeding due to Poisson blending, the overall structure looks visually pleasing. (f) This figure shows from which image the algorithm took the patches. The green region denotes patches from (a) and the yellow region denotes patches from (b).



Figure 18. These examples illustrate typical failure cases. In the top row, we move the sun to the right. Although the objects on the beach reorganize themselves to accommodate the user constraint, the sky patches propagate into the sea losing the overall structure of the image. In the middle row, we move the island in the sea to the right and let the patch transform reconstruct the image. The patch transform *thinks* that the optimal way to meet that constraint is to break the cliff into two. It could be possible that breaking the cliff at that seam is the optimal in terms of energy, but is still visually unnatural. In the bottom row, we move the person and a part of the temple to the right, and fixed the tourists at the left bottom corner. It's very hard to find a visually pleasing image with these constraints because some structures cannot be reorganized to generate natural looking structures.

### 3. More examples that could not be added to the paper

#### 3.1. Moving the position of an object in the image

In this section, we show that the patch transform can be used to change the position of a certain object/region in the image.



Figure 19. This example illustrates how the patch transform framework can be used to recenter a region / object of interest. (a) The original image. (b) A user specified constraint that the girl should be recentered. (c) The patch transform result without Poisson blending. Some blocky artifacts are visible. Notice that the overall structure and context is preserved. Notice that the bottom part of the reconstructed is a mere circular shift of the original image. This actually makes sense if we look at the boundary (the seam) where the right side and the left side of the original image meets. Interestingly, the branches are correctly connected, and the ground is also connected as well. (d) The patch transform result after Poisson Blending.



(a)



(b)



(c)



(d)

Figure 20. This example illustrates how the patch transform framework can be used to move a region / object of interest. (a) The original image. (b) A user specified constraint that the mountain peak should be moved to the right. (c) The patch transform result without Poisson blending. Some blocky artifacts are visible. Notice that the overall structure and context is preserved. (d) The patch transform result after Poisson Blending.

### 3.2. Changing the relative position of two objects/regions

The next 10 examples show how the patch transform framework can be used to change the relative position of two, possibly multiple, objects in the image. The examples are assorted to cover many imaging scenarios with different background.

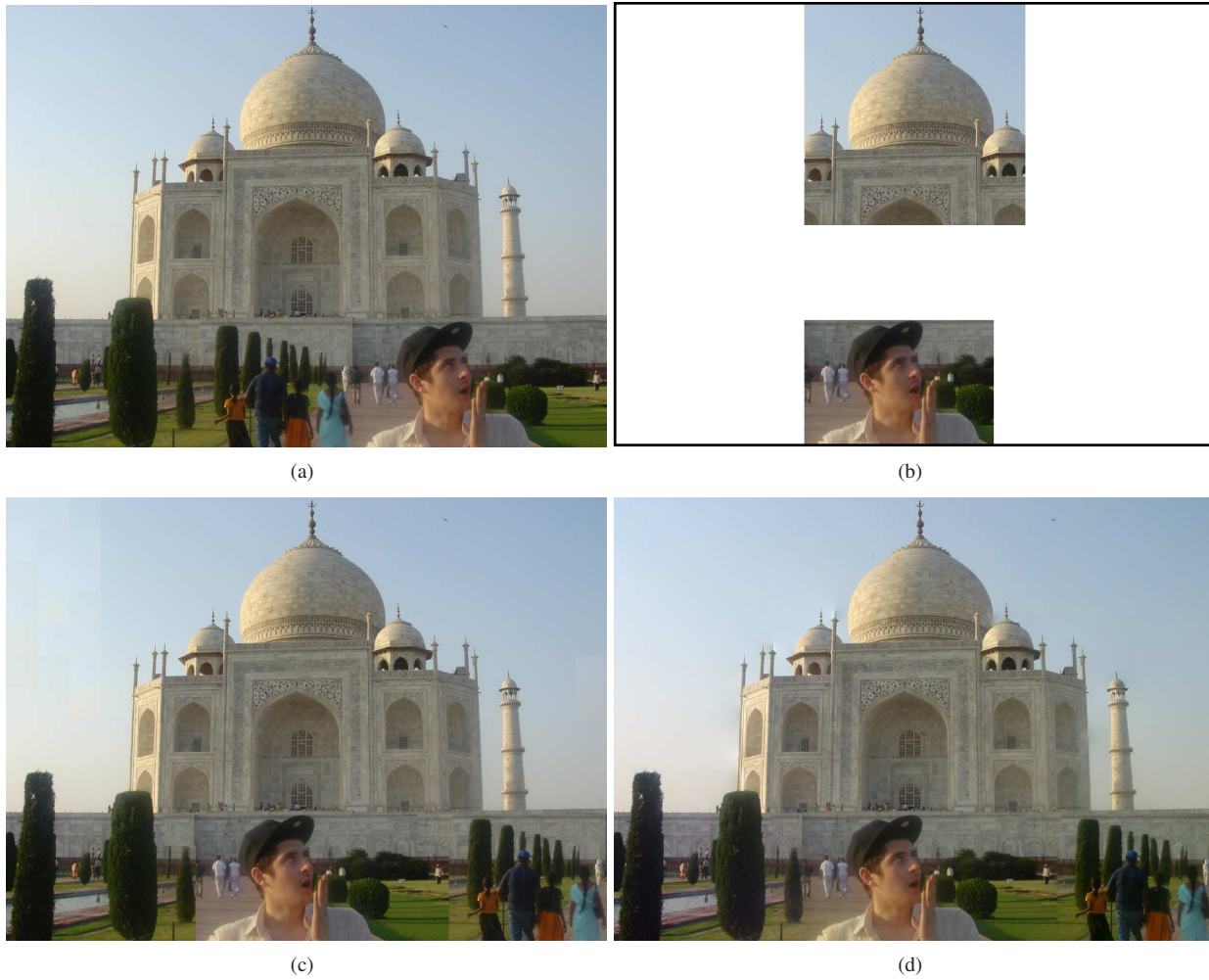
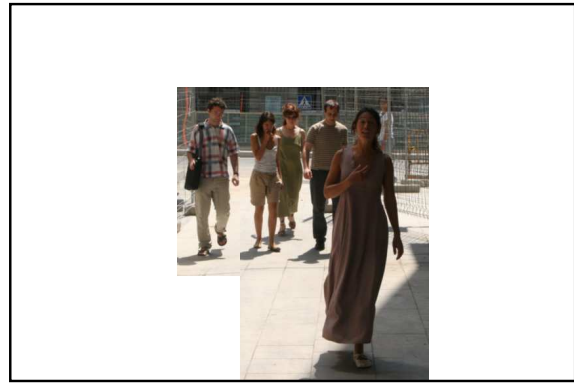


Figure 21. This example verifies that the proposed framework can still work well in the presence of complex background. We are interested in centering the man on the right. (a) The original image. (b) The user specified that the man and Taj Mahal should be more or less aligned. (c) The patch transform result without Poisson blending. While the algorithm did not move the building, the algorithm reshuffled the garden and the tourist to accommodate the changes in the man's position. (d) The patch transform result with Poisson blending.



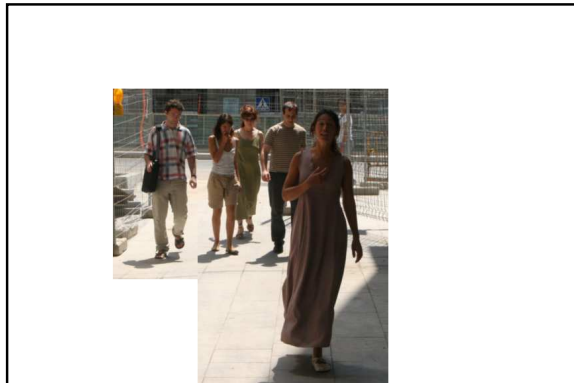
(a)



(b)



(c)



(d)



(e)

Figure 22. This example shows the process of manipulating the relative positions of multiple objects in the image. (a) The original image. (b) The man on the left is specified to move closer to the rest of the group. (c) The patch transform result. To accommodate the constraint (b), the background has been reconfigured: the advertisement panel became shorter, and the redundant window panes have been moved to the top left corner. Also, some tiles from the pavement moved to the left corner, which now looks like a wooden board attached to the fence. (d) The group on the right is specified to move closer to the man on the left. (e) The patch transform result. The algorithm reconstructed a different background: while the building structure is modified similarly, the right corner of the corridor has been extended toward the center with tiles from the pavement.





(a)



(b)



(c)

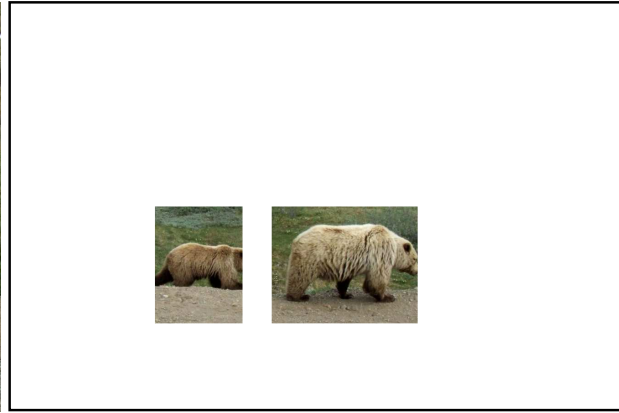


(d)

Figure 23. In the original image (a), a bear and two cubs are sleeping on the road. Because the cub on the left is isolated from the rest, we want to move the other two such that they don't look to separated. (b) User specified input that the three should stay close to each other. (c) The patch transform result without Poisson blending. (d) Reconstructed result with Poisson blending.



(a)



(b)



(c)



(d)

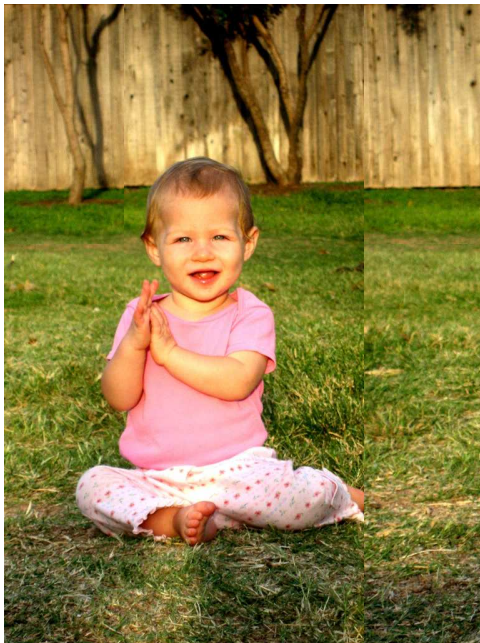
Figure 24. In the original image (a), the cub is falling behind the mother bear, so we want to bring the mother bear closer to the cub. (b) The user input place the cub and the mother bear closer. (c) Reconstructed result without Poisson blending (d) Reconstructed result with Poisson blending.



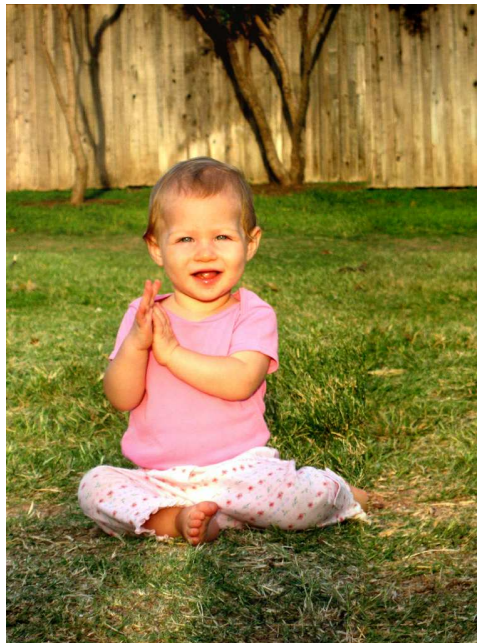
(a)



(b)



(c)



(d)

Figure 25. This example illustrates how we can modify the background by manipulating the relative position between the tree and the baby. (a) Original Image. (b) User Input (c) Reconstructed result without Poisson blending (c) Reconstructed result with Poisson blending.



(a)



(b)



(c)



(d)

Figure 26. This is an easy example of moving the lady to left side of the image. Because the background has gridded structure, the patch transform framework works well. (a) Original Image. (b) The user wants to fix on side of the image and move the lady to the other side. (c) Reconstructed result without Poisson blending. While the local structure of the image is visually similar, the 3D interpretation of the new image can be different from the original image. (d) Reconstructed result with Poisson blending.



(a)



(b)



(c)

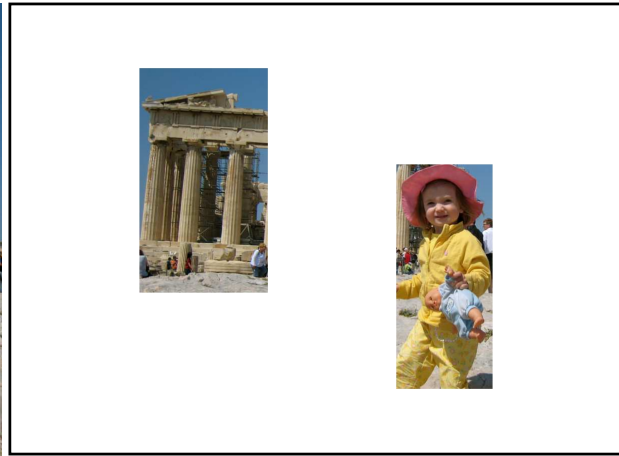


(d)

Figure 27. In this example, we are interested in moving the girl's shadow such that we get an image that cannot be taken. (a) The original image. (b) The user specified that the shadow move to the other side of the girl. (c) The patch transform result without Poisson blending. (d) The patch transform result with Poisson blending. This type of image cannot be taken with an ordinary lighting setup, but can easily be generated with the patch transform framework.



(a)



(b)



(c)



(d)

Figure 28. Because the Parthenon is placed too much to the left, we are interested in bringing the girl and the Parthenon back to the middle of the image. (a) The original Image. (b) The user specified that the girl and the parthenon should move to the right. (c) Reconstructed result without poisson blending (c) Reconstructed result with poisson blending.



(a)



(b)



(c)



(d)

Figure 29. This is an easy example of changing the relative position of a group of people. We are interested in moving the man on the right closer to the rest of the group. (a) The original Image. (b) The user wants to move the man closer to the group (c) Reconstructed result without poisson blending (d) Reconstructed result with poisson blending. This is a particularly easy example since the background is very textured.

### 3.3. Adding images in the patch domain

In the next four examples, we show that the patch transform can be used to collage two, possibly multiple, images.



Figure 30. This example shows how two images can be added in a patch domain to generate a collage. (a, b) The original images, (c) A user specified constraint that people from two different images should stand next to each other. (d)The inverse patch transform result. Interestingly, the algorithm automatically placed the tree from (a) in front of the trees from (b) to generate a contextually correct image.





(a)



(b)



(c)



(d)

Figure 31. This example shows how two images can be added in a patch domain to generate a collage. (a, b) The original images, (c) A user specified constraint that people from two different images should stand next to each other. (d)The inverse patch transform result. Notice that patches from two images are blended nicely to generate a natural looking image.



(a)



(b)



(c)



(d)

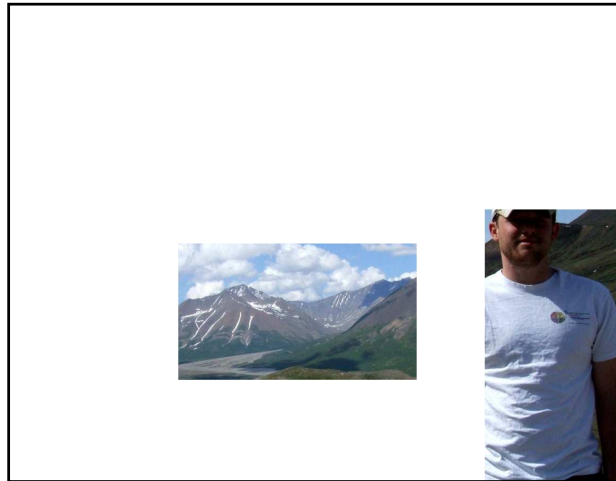
Figure 32. We are interested in hybridizing the two images shown in (a) and (b). (c) The user specified that a part of the image from (a) will serve as a background of a part of the tree from (b). (d) The patch transform result. Interestingly, the branches from two different trees are hybridized to synthesize a new tree. Notice that the branches are well-aligned, and the sky-line follows the style of two images simultaneously. Some misplaced patches, such as that of the electrical wires, can be corrected with a user intervention, but is not exploited in this work. The tone difference of the two trees has been compensated by poisson blending



(a)



(b)



(c)



(d)



(e)



(f)

Figure 33. In this example, we collage two images shown in (a) and (b). We want to copy a part of the image from (b) onto (a) to generate a new background. (c) The user specified that part of the valley from (b) be placed next to a person shown on the right. (d) The patch transform result without Poisson blending. (e) The patch transform result with Poisson blending. Although we see some color bleeding due to Poisson blending, the overall structure looks visually pleasing. (f) This figure shows from which image the algorithm took the patches. The green region denotes patches from (a) and the yellow region denotes patches from (b).