MAS.131/MAS.531
Separating Transparent Layers in Images

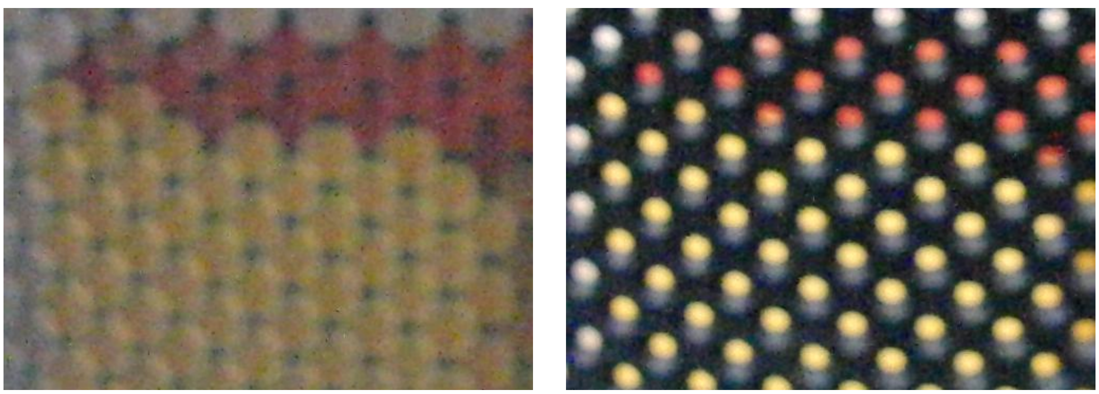
Anonymous MIT student

Abstract

This paper presents an algorithm to separate different transparent layers present in images of a 3-D scene. The criterion of minimum mutual information is presented and evaluated. We find that high frequency details are faithfully recovered while the recovery of low frequency details needs improvement.

INTRODUCTION

Typically, when we image a 3-D scene using a camera, different 2-D layers of the scene that are at different distances from the camera are superimposed. For example, consider the images shown in Figure 1. The two images belong to the same scene containing a chair and a file cabinet. However, the images are captured by focussing at different depths. The surface that is in focus is sharp and the surface that is out of focus is blurred.



(a) The file cabinet is in focus (b) The chair is in focus

Fig. 1. **An example illustrating superposition of transparent layers in an Image.**

Such images are often encountered in microscopy while studying the anatomy of various organisms. Different slices of cells or tissues are captured by focussing at different depths. In order to study any particular 2-D slice from such images, it is important to separate the effect of other 2-D layers that are super-imposed in the images.

THE MODEL

Let a 2-D layer $f_1(x, y)$ interfere with a 2-D layer $f_2(x, y)$. We consider the slices $g_a(x, y)$ and $g_b(x, y)$, in which either layer $f_1(x, y)$ or layer $f_2(x, y)$ respectively, is in focus. The other layer is blurred. Modeling the effect of blur as a convolution with blur kernals, we have,

$$\begin{aligned} g_a(x, y) &= \big(f_2(x, y) \otimes h_{2a}(x, y)\big).f_1(x, y) \\ g_b(x, y) &= \big(f_1(x, y) \otimes h_{1b}(x, y)\big).f_2(x, y) \end{aligned} \tag{1}$$

where '$\otimes$' denotes convolution and '.' denotes multiplication. The above model does, in fact describes the images shown in Figure 1. For example, consider a region in Figure 1(b) where the chair has holes. In that case, we are able to 'see through' a blurred version of the file cabinet. The regions where there are no holes in the chair in Figure 1(b), we do not 'see through' any version of the file cabinet. And hence, the 'multiply after convolution' model. We assume that the blur-kernals are space-invariant for constant depth objects. Moreover, if the system is telecentric, then $h_{1b}(x,y) = h_{2a}(x,y) = h(x,y)$.

Prior Art

Separation of different layers in images has been studied before. In [1], [2], the authors propose the use of Independent Component Analysis [3], [4] to separate reflections from Images that are caputered with a polarizer. Using local features to perceive transparency in images has been proposed in [5], [6]. The closest to our work is [7], [8], where the authors propose separation of transparent layers using a different model for superposition. The novelty of our work lies in the formulation of the transparent layer interference as a mutliplication of images as opposed to the standard approach of summation.

Our approach

The two equations in eq (1) have four unknowns, namely $f_1(x,y), f_2(x,y), h_{2a}(x,y)$ and $h_{1b}(x,y)$. And hence, this is a highly under-determined system. To solve the above system of equations, we make the assumption that $h_{1b}(x,y) = h_{2a}(x,y) = h(x,y)$ (This is a reasonable assumption especially for telecentric images [7], [8]). Under this assumption, the resulting model is given by,

$$\begin{aligned} g_a(x,y) &= \big(f_2(x,y) \otimes h(x,y)\big).f_1(x,y) \\ g_b(x,y) &= \big(f_1(x,y) \otimes h(x,y)\big).f_2(x,y) \end{aligned} \tag{2}$$

We are still left with three unknowns and two equations. Here, we leverage the idea of minimum mutual information that was proposed in [7], [8]. The underlying intuition is that if $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$ are the solutions to the equations (2), then $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$ have the least mutual information $\mathcal{I}(\hat{f}_1(x,y), \hat{f}_2(x,y))$ among all possible solutions to equations (2). We use this criterion to find the optimal $h(x,y)$ that yields minimum mutual information between the recovered images $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$. Even given $h(x,y)$, finding the solution to the equations (2) is challenging. Our algorithm for separating $f_1(x,y)$ and $f_2(x,y)$ is given below:

1) Assume a blur-kernal $h(x,y)$ and solve for $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$ using equations (2). This is not straightforward from the equations themselves. We find $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$ iteratively from equations (2) as follows:
   a) Initialize $\hat{f}_1(x,y) = 1$ and $\hat{f}_2(x,y) = 1$.
   b) Iteratively update $\hat{f}_1(x,y)$ and $\hat{f}_1(x,y)$ as

$$\begin{aligned} \hat{f}_1(x,y) &= \frac{g_a(x,y)}{\big(\hat{f}_2(x,y) \otimes h(x,y)\big)} \\ \hat{f}_2(x,y) &= \frac{g_b(x,y)}{\big(\hat{f}_1(x,y) \otimes h(x,y)\big)} \end{aligned}$$

   c) Return $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$ after ten iterations.
2) Calculate the normalized minimum mutual information

$$\mathcal{I}_n\big(\hat{f}_1(x,y), \hat{f}_2(x,y)\big) = \frac{\mathcal{I}\big(\hat{f}_1(x,y), \hat{f}_2(x,y)\big)}{[\mathcal{H}\big(\hat{f}_1(x,y)\big) + \mathcal{H}\big(\hat{f}_2(x,y)\big)]/2}$$

where

$$\mathcal{H}\big(\hat{f}_1(x,y)\big) = -\sum_{\hat{f}_1(x,y)} P\big(\hat{f}_1(x,y)\big)\log P\big(\hat{f}_1(x,y)\big),$$

$$\mathcal{I}\big(\hat{f}_1(x,y),\hat{f}_2(x,y)\big) = -\sum_{\substack{\hat{f}_1(x,y),\\ \hat{f}_2(x,y)}} P\big(\hat{f}_1(x,y),\hat{f}_2(x,y)\big)\log \frac{P\big(\hat{f}_1(x,y),\hat{f}_2(x,y)\big)}{P\big(\hat{f}_1(x,y)\big)P\big(\hat{f}_2(x,y)\big)}$$

3) Change the blur kernal $h(x,y)$ and return to step 1.

In our problem, we assume that the blur-kernels are Gaussian and are parameterized by their standard deviation σ. We iterate over the steps $1-3$ mentioned above by changing σ. The final estimates of the separated layers are the ones that result in the least normalized mutual-information across all the runs. For more details, refer to the MATLAB code in the appendix.

EXPERIMENTAL EVALUATION

We generated multi-layered images according to equations (2) by combining two sharp images as shown in Figure 4. We used $\sigma = 0.7$ for the images. The calculated normalized mutual information is shown in the Figure 2. We observe that the algorithm correctly estimates σ to be 0.7. The recovered images are shown in Figure 4. We find that the image recovery preserves the high-frequency information but performes poorly in the low-frequency components. Shown in Figure 3 are the x-gradients for a particular row for both, the original image and the recovered image. We find that the gradients are very accurate, confirming that the high frequency components are well preserved. The loss in the low-frequencies is retained even if the iterations in the algorithm mentioned above were initialized with better estimates for $\hat{f}_1(x,y)$ and $\hat{f}_2(x,y)$. From these observations, it looks like the loss in low-frequency information can be avoided only by using a different approach to solving equations (2) given the blur-kernel. Since our algorithm estimates the blur-kernel very accurately, we can use it to estimate σ and then use an enhanced solver to solve for equations (2) using the σ estimated by our algorithm.

We now proceed to test and see if the image separation works when both the transparent layers are similar (for example if the 3-D scence that we are imaging is comprised of 2-D scenes that have similar patterns). Since the criterion we are using to find the optimal blur-kernel is based on minimizing the normalized mutual information between the recovered images, one might suspect that this approach might not work well if the images are similar. However, it turns out that when the images are similar, their mutual information is already high to begin with and hence the minimum mutual information criterion continues to work. Shown in Figure 5 are the results when overlapping portions of the house picture in Figure 4 are used as the transparent layers.

CONCLUSION

We find that the minimum mutual-information criterion works very well in separating transparent layers present in images. An iterative algorithm was presented that uses this criterion for separating the layers. There is scope for improvement in the separation quality for the low-frequencies.

REFERENCES

[1] Hany Farid and Edward H. Adelson, "Separating reflections and lighting using independent components analysis," in *Proceedings, IEEE Computer Society conference on Computer Vision and Pattern Recognition*, 1999, vol. 1, pp. 262–267.

[2] Hany Farid and Edward H. Adelson, "Separating reflections from images using independent component analysis," *Optical Society of America*, vol. 16, no. 9, pp. 2136–2145, 1999.

[3] Jean-Francois Cardoso, "Source separation using higher order moments," in *Proceedings, International Conference on Acoustics, Speech and Signal Processing*, 1989, pp. 2109–2112.

[4] J.L. Lacoume and P. Ruiz, "Separation of independent sources from correlated inputs," *IEEE Transactions on signal processing*, vol. 40, no. 12, December 1992.

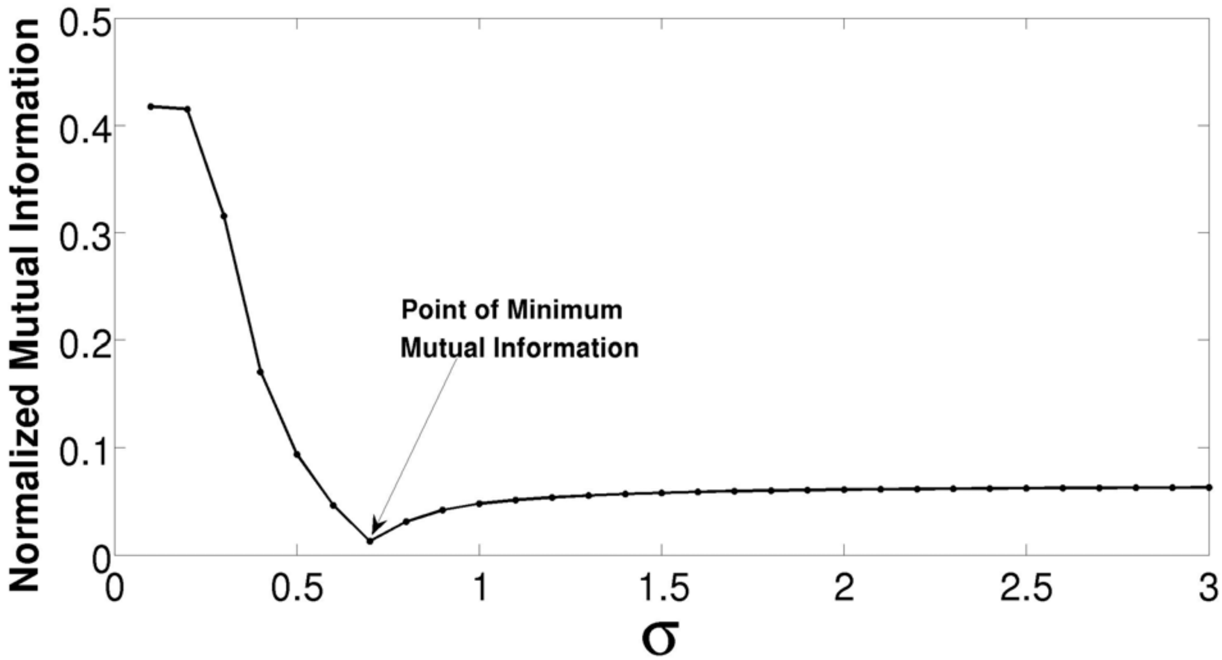


Fig. 2. Calculated normalized mutual information between the recovered images. We find that the normalized mutual information is in fact minimized at the original $\sigma = 0.7$.

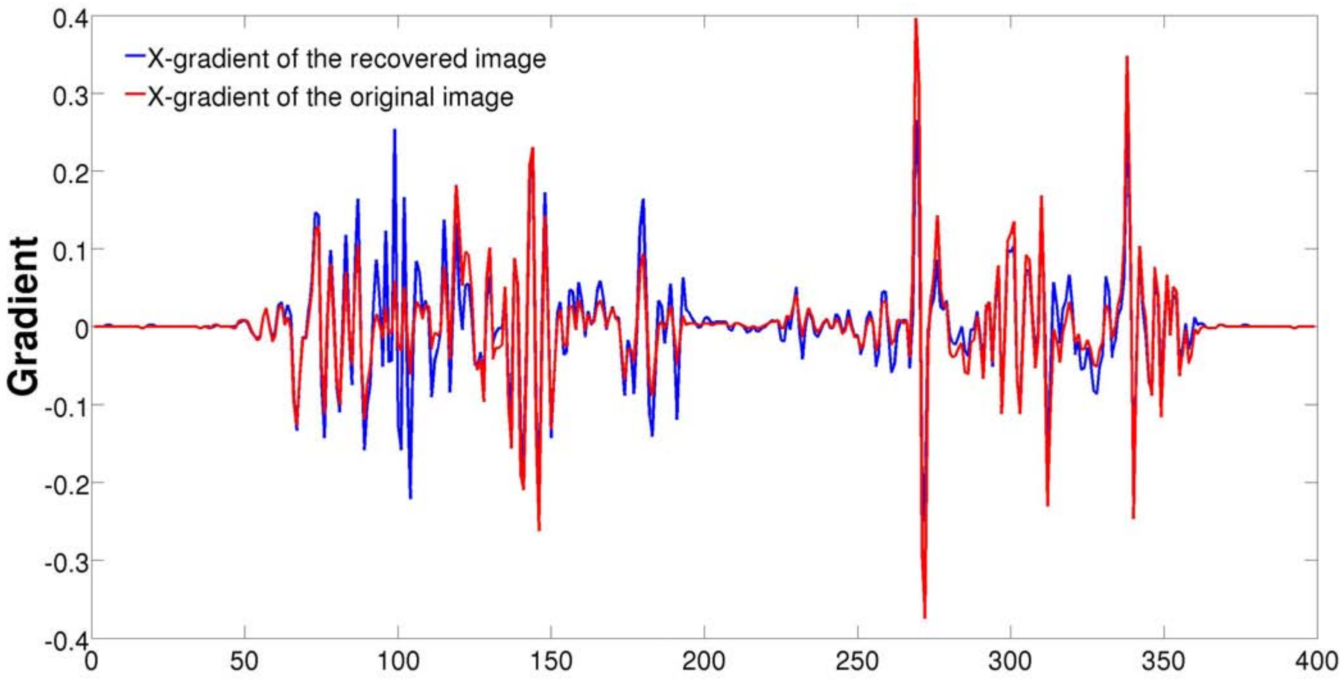
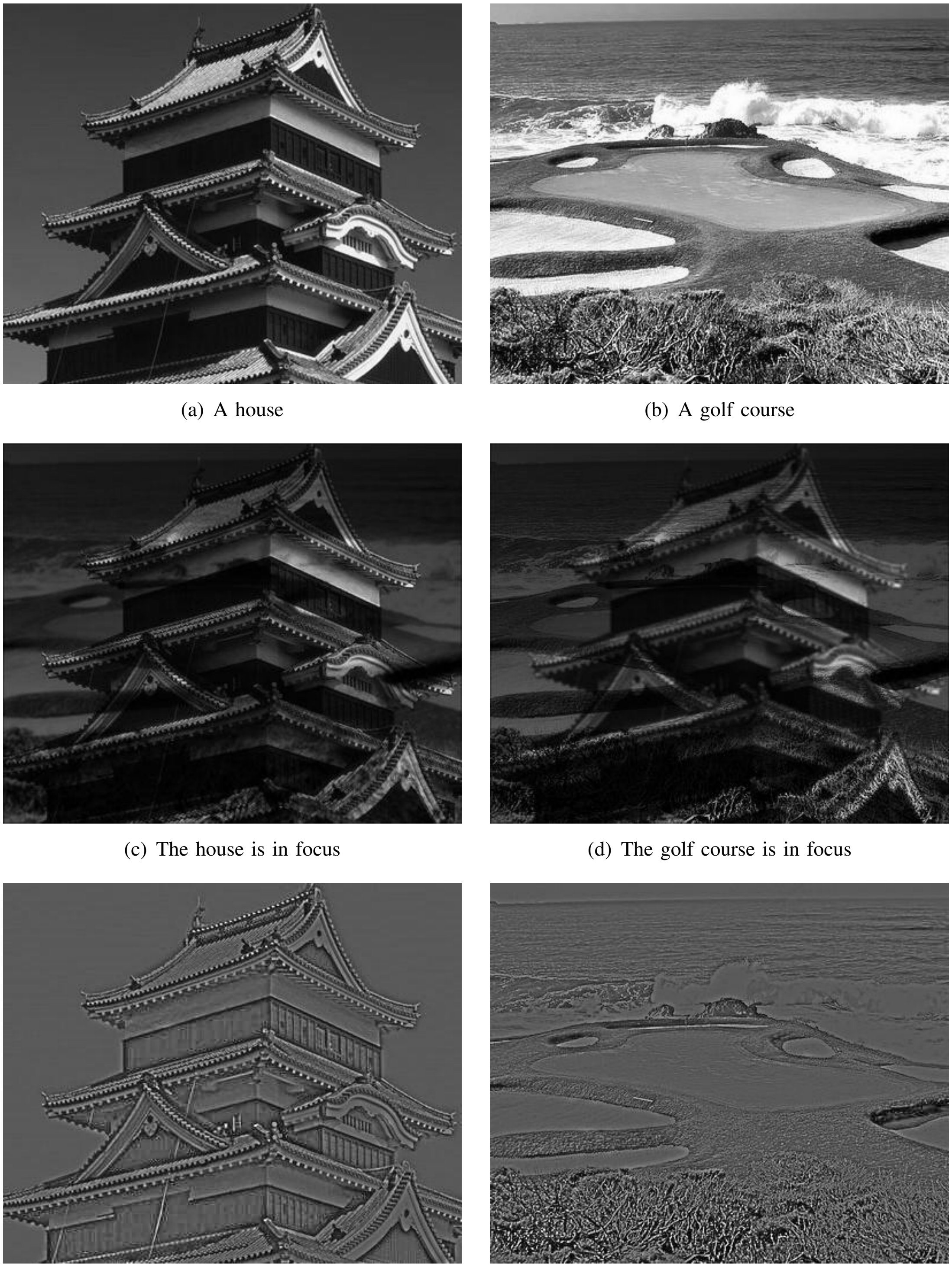


Fig. 3. x-gradients of the recovered image compared with that of the original image.

[5] Anat Levin, Assaf Zomet, and Yair Weiss, "Separating reflections from a single image using local features," in *IEEE Computer Society conference on Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 306–313.
[6] Anat Levin, Assaf Zomet, and Yair Weiss, "Learning to perceive transparency from the statistics of natural scenes," in *In NIPS-15; The 2002 Conference on Advances in Neural Information Processing Systems*, 2002.
[7] Yoav Y. Schechner, Nahum Kiryati, and Joseph Shamir, "Multi-valued images and their separation," in *Proceedings of the 10th International Workshop on Theoretical Foundations of Computer Vision: Multi-Image Analysis*. 2000, pp. 129–141, Springer-Verlag.

[8] Yoav Y. Schechner, Nahum Kiryati, and Joseph Shamir, "Separation of transparent layers using focus," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 25–39, August 2000.



(a) A house

(b) A golf course

(c) The house is in focus

(d) The golf course is in focus

(e) The recoverd house

(f) The recovered golf course

Fig. 4. **The separation of transparent layers in an Image.**

APPENDIX

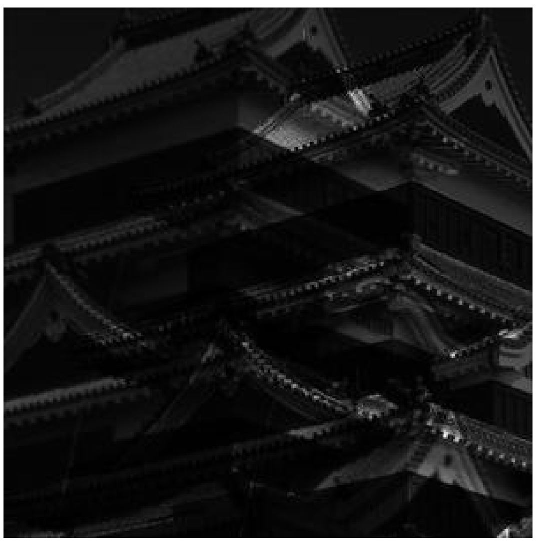
MATLAB *code*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
%   This code aims at solving the following set of equations:                  %
%                                                                              %
%               y1 = (g @ f1) . f2                                             %
%               y2 = f1 . (g @ f2)                                             %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f1_color = imread('../../images/chinese_house.jpg');
f2_color = imread('../../images/golf_course.jpg');

f1_orig = rgb2gray(f1_color);
f2_orig = rgb2gray(f2_color);

f1_orig = double(f1_orig)/256;
f2_orig = double(f2_orig)/256;

g_orig = fspecial('gaussian', [3, 3], 0.7);

y1 = imfilter(f1_orig, g_orig).*f2_orig;
y2 = imfilter(f2_orig, g_orig).*f1_orig;

num_iterations = 10;

%%%%%%%% We now search for g with different standard deviations %%%%%%%%%%

sig_range = 0.1:0.1:3;

cross_info = zeros(size(sig_range));

count = 1;

min_mutual_info = 100;

for sig = sig_range

    %%%% Begin iterations by initializing f2 = y1 and f1 = y2

    g_loop = fspecial('gaussian', [3,3], sig);

    G_loop = fft(g_loop);

    f1 = 0.5*ones(size(f1_orig));
    f2 = 0.5*ones(size(f2_orig));
```

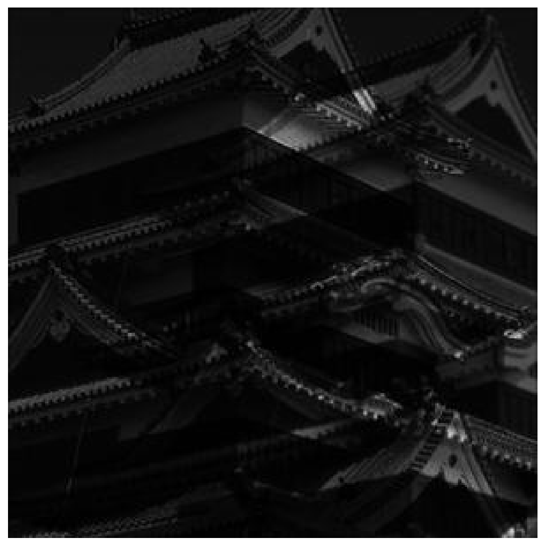
```
    for idx = 1:num_iterations

        %den2 = ifft(G_loop.*fft(f1));
        %den1 = ifft(G_loop.*fft(f2));

        den2 = imfilter(f1, g_loop);
        den1 = imfilter(f2, g_loop);

        %f2 = 0.5*(y1.*(1+(1-den2))).*(1+sign(y1.*(1+(1-den2))));
        %f1 = 0.5*(y2.*(1+(1-den1))).*(1+sign(y2.*(1+(1-den1))));

        f2 = (y1./den2);
        f1 = (y2./den1);

    end

    f1_image = uint8(128*f1);
    f2_image = uint8(128*f2);

    entropy1 = entropy(f1_image(:));
    entropy2 = entropy(f2_image(:));

    cross_info(1,count) = mutualinfo(f1_image,f2_image)/(entropy1+entropy2);

    if (min_mutual_info > cross_info(1,count))
        min_mutual_info = cross_info(1,count);
        f1_recovered = f1;
        f2_recovered = f2;
    end

    count = count + 1;

end

f1_image = double(uint8(256*0.7*f1_recovered))/256;
f2_image = double(uint8(256*0.7*f2_recovered))/256;
```

(a) A portion of the house

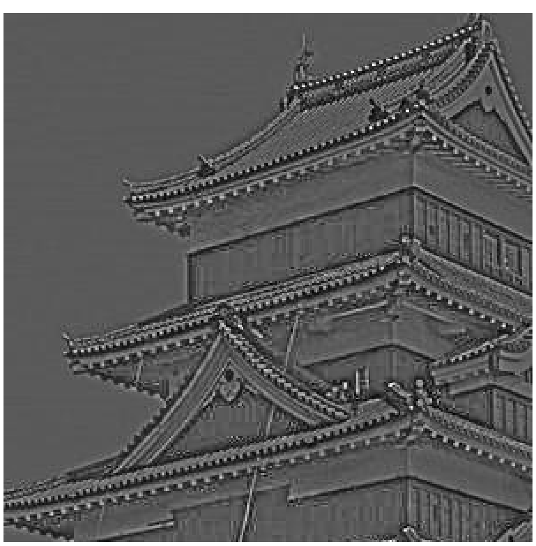
(b) Another portion of the house



(c) The first portion is in focus



(d) The second portion is in focus



(e) The recovered first portion of the house

(f) The recovered second portion of the house

Fig. 5. **The separation of similar transparent layers in an Image.**