Fragment-Based Image Completion

Iddo Drori Daniel Cohen-Or Hezy Yeshurun

School of Computer Science *
Tel Aviv University

Figure 1: From left to right: the input image, and inverse matte that defines the removal of an element, the result of our completion, and the content of the completed region.

Abstract

We present a new method for completing missing parts caused by the removal of foreground or background elements from an image. Our goal is to synthesize a complete, visually plausible and coherent image. The visible parts of the image serve as a training set to infer the unknown parts. Our method iteratively approximates the unknown regions and composites adaptive image fragments into the image. Values of an inverse matte are used to compute a confidence map and a level set that direct an incremental traversal within the unknown area from high to low confidence. In each step, guided by a fast smooth approximation, an image fragment is selected from the most similar and frequent examples. As the selected fragments are composited, their likelihood increases along with the mean confidence of the image, until reaching a complete image. We demonstrate our method by completion of photographs and paintings.

CR Categories: I.3.3 [Computer Graphics]: Picture/image generation—; I.4.1,3,5 [Image Processing and Computer Vision]: Sampling—,Enhancement,Reconstruction

Keywords: image completion, example-based synthesis, digital matting, compositing

*e-mail: {idrori | dcor | hezy}@tau.ac.il

1 Introduction

The removal of portions of an image is an important tool in photo-editing and film post-production. The unknown regions can be filled in by various interactive procedures such as clone brush strokes, and compositing processes. Such interactive tools require meticulous work driven by a professional skilled artist to complete the image seamlessly. Inpainting techniques restore and fix small-scale flaws in an image, like scratches or stains [Hirani and Totsuka 1996; Bertalmio et al. 2000]. Texture synthesis techniques can be used to fill in regions with stationary or structured textures [Efros and Leung 1999; Wei and Levoy 2000; Efros and Freeman 2001]. Reconstruction methods can be used to fill in large-scale missing regions by interpolation. Traditionally, in the absence of prior knowledge, reconstruction techniques rely on certain smoothness assumptions to estimate a function from samples. Completing large-scale regions with intermediate scale image fragments remains a challenge.

Visual perceptual completion is the ability of the visual system to "fill in" missing areas [Noe et al. 1998] (partially occluded, coinciding with the blind spot, or disrupted by retinal damage). While the exact mechanisms behind this phenomenon are still unknown, it is commonly accepted that they follow some Visual Gestalt [Koffka 1935, 1967] principles, namely, completion by frequently encountered shapes that result in the simplest perceived figure [Palmer 1999]. Motivated by these general guidelines, we iteratively approximate the missing regions using a simple smooth interpolation, and then add details according to the most frequent and similar examples.

Problem statement: Given an image and an inverse matte, our goal is to complete the unknown regions based on the known regions, as shown in the figure above.

In this paper, we present an iterative process that interleaves smooth reconstruction with the synthesis of image fragments by example. The process iteratively generates smooth reconstructions to guide the completion process which is based on a training set derived from the given image context.

The completion process consists of compositing image frag-

ments and can be regarded as a "push-background" process, in contrast to the "pull-foreground" processes associated with image matting. Our completion approach requires a relevant training set and a degree of self-similarity within the input image. It is an image-based 2D method that does not incorporate high-level information and can therefore produce unnatural looking completions.

This paper is organized as follows. After a survey of related work, we present an overview of the entire completion process (Section 2), and then describe each component of the method in detail. We introduce a fast approximation method (Section 3), and present a confidence map and traversal order (Section 4). Next, we describe the search for similar fragments and their adaptive neighborhood size (Section 5). In Section 6 we present the compositing of image fragments. Completion is performed from coarse-to-fine, as described in Section 7. Finally we show the results of our method on various photographs and paintings (Section 8), and discuss its limitations (Section 9).

1.1 Related work

Many operations ranging from low-level vision tasks to high-end graphics applications have been efficiently performed based on examples:

Hertzmann et al. [2001] study a mapping of spatially local filters from image pairs in correspondence, one a filtered version of the other. A new target image is filtered by example. Pixels are assigned values by comparing their neighborhood, and those of a coarser level in a multi-resolution pyramid, to neighborhoods of a training image pair. Considering the original image as output and taking its segmentation map as input allows texture to be painted by numbers [Haeberli 1990]. A new image that is composed of the various textures is then synthesized by painting a new segmentation map. Swapping the output segmentation map with the original input image results in example-based segmentation [Borenstein and Ullman 2002].

Freeman et al. [2000; 2002] derive a model used for performing super-resolution by example. The technique is based on examining many pairs of high-resolution and low-resolution versions of image patches from several training images. Baker and Kanade [2000] apply this technique restrictively to the class of face images. Given a new low-resolution face image, its corresponding high-resolution image is inferred by re-using the existing mapping between individual low-resolution and high-resolution face patches.

The example-based image synthesis methods described above use a supervised training set of corresponding pairs. In our work, the examples provide the likelihood of a given context appearing in the given image.

There is considerable work on texture synthesis, of which the most notable are based on Markov Random Fields [Efros and Leung 1999]. A new texture is incrementally synthesized by considering similar neighborhoods in the example texture. These techniques synthesize texture which is both stationary and local [Wei and Levoy 2000]. Igehy and Periera [1997] replace image regions with synthesized texture [Heeger and Bergen 1995] according to a given mask. Texture transfer [Ashikhmin 2001] adds the constraint that the synthesized texture match an example image. This yields the effect of rendering a given image with the texture appearance of a training texture. This technique is extended to color transfer [Welsh et al. 2002], a special case of image analogies, by matching local image statistics. Efros and Freeman [2001] introduce a simple and effective texture synthesis technique that synthesizes a new texture by stitching together blocks of existing example texture. The results depend on the size of a block which is a parameter tuned by the user that varies according to the texture properties. In this work, we synthesize both local and global structures. To capture structures of various sizes, similarly to hierarchical pattern matching [Soler et al. 2002], we take an adaptive approach, where fragments have different sizes based on the underlying structure.

Image inpainting [Bertalmio et al. 2000; Chan and Shen 2001] fills in missing regions of an image by smoothly propagating information from the boundaries inwards, simulating techniques used by professional restorators. However, the goals of image inpainting techniques and image completion are different. Image inpainting is suitable for relatively small, smooth, and non-textured regions. In our case the missing regions are large, and consist of textures, large-scale structures, and smooth areas. Recently, Bertalmio et al. [2003] combine image inpainting with texture synthesis by decomposing an image into the sum of two components. Inpainting [Bertalmio et al. 2000] is applied to the component representing the underlying image structure, whereas texture synthesis [Efros and Leung 1999] is separately applied to the component representing image detail, and the two components are then added back together.

There are a number of approaches related to image completion in the Computer Vision literature, but most of them focus on the edge and contour completion aspect of the problem. Edge completion methods find the most likely smooth curves that connect edge elements, usually by minimizing a function based on curvature [Guy and Medioni 1996]. Given a grid of points and orientations as elements, completion methods consider the space of all curves between pairs of elements, and the pairwise interaction between elements. The likelihood that any two elements are connected defines a field for each element, and completion is performed by a summation over all fields [Williams and Jacobs 1997]. Sharon et al. [2000] take into account edge elements at various scales, and use a multi-grid method to accelerate computations.

Our work is also related to photo-editing techniques. Oh et al.[2001] incorporate some depth information into photo-editing, whereas our method is a 2D image-based technique, which has no notion of the underlying scene. Brooks and Dodgson [2002] present an image editing technique that is based on texture self-similarity, editing similar neighborhoods in different positions. Our method finds self-similarities in the image under a combination of transformations: translation, scale, rotation and reflection.

2 Image completion

We assume that foreground elements or background regions are roughly marked with an image editing tool, or a more accurate α channel is extracted using a matting tool. This defines an *inverse matte* $\bar{\alpha}$ that partitions the image into three regions: the known region, where $\bar{\alpha}_i = 1$; unknown region, where $\bar{\alpha}_i = 0$; and, optionally, a gray region, where $0 < \bar{\alpha}_i < 1$ for each pixel i, and "inverts" the common definition of trimaps that are generated in the process of pulling a matte and foreground elements from an image [Chuang et al. 2002]. We require a *conservative* inverse matte that, at least, contains the entire extracted region. As in digital image matting, the regions of the inverse matte are not necessarily connected. The inverse matte defines a confidence value for each pixel. Initially, the confidence in the unknown area is low. However, the confidence of the pixels in the vicinity of the known region is higher. The confidence values increase as the completion process progresses.

Our approach to image completion follows the principles of *figural simplicity* and *figural familiarity*. Thus, an *approximation* is generated by applying a *simple* smoothing process in the low confidence areas. The approximation is a rough classification of the pixels to some underlying structure that agrees with the parts of the image for which we have high confidence. Then the approximated region is augmented with *familiar* details taken by example from a region with higher confidence.

All of these processes are realized at the image fragment level. A fragment is defined in a circular neighborhood around a pixel. The size of the neighborhood is defined adaptively, reflecting the scale

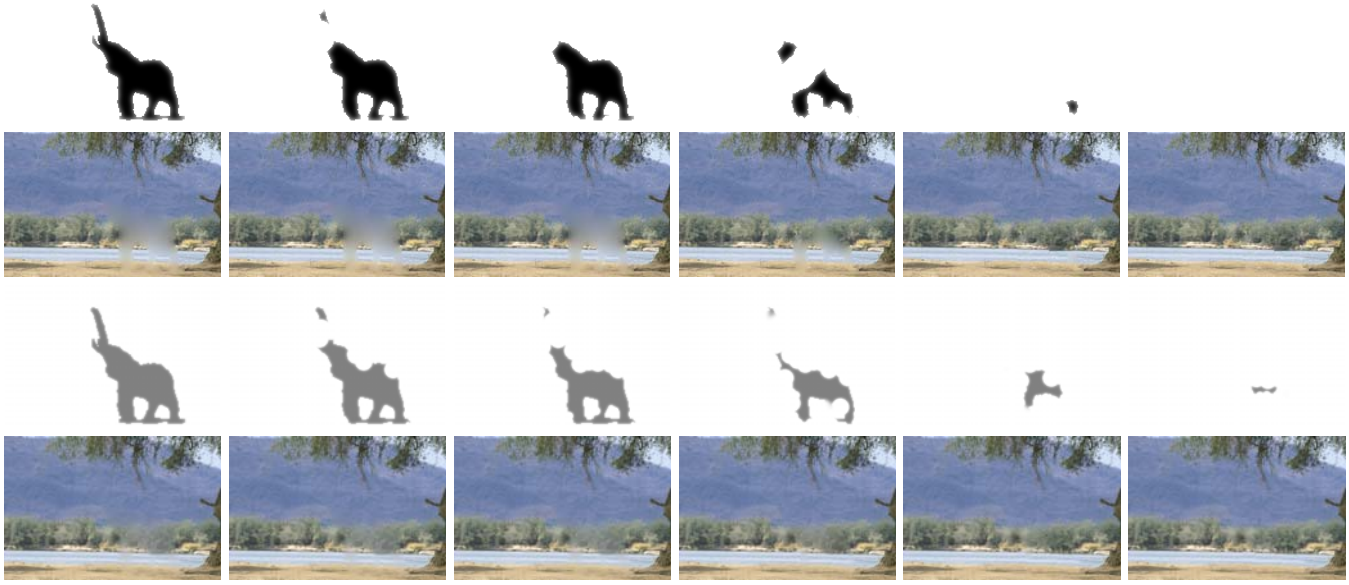


Figure 2: Completion process: confidence and color of coarse level (top row) and fine level (bottom row) at different time steps. The output of the coarse level serves as an estimate in the approximation of the fine level.

of the underlying structure. Image completion proceeds in a multi-scale fashion from coarse to fine, where first, a low resolution image is generated and then the results serve as a coarse approximation to the finer level. For every scale we consider neighborhoods in level sets from high to low confidence. Figure 2 shows the confidence and color values at different time steps in each scale.

At each step, a *target* fragment is completed by adding more detail to it from a *source* fragment with higher confidence. Typically, the target fragment consists of pixels with both low and high confidence. The pixel values which are based on the approximation generally have low confidence, while the rest of the fragment has higher confidence. For each target fragment we search for a suitable matching source fragment, as described in Section 5, to form a coherent region with parts of the image which already have high confidence. The search is performed under combinations of spatial transformations to extend the training set and make use of the symmetries inherent in images. The source and target fragments are composited into the image as described in Section 6. The algorithm updates the approximation after each fragment composition. As fragments are added, the mean confidence of the image converges to one, completing the image.

A high-level description of our approach appears in Figure 3. In the pseudocode, the following terms are emphasized: (i) approximation, (ii) confidence map, (iii) level set, (iv) adaptive neighborhood, (v) search, and (vi) composite. These are the building blocks of our technique. In the following sections we elaborate on each in detail.

Input: image C, inverse matte $\bar{\alpha}$ ($\exists$ pixel with $\bar{\alpha} < 1$)

Output: completed image, $\bar{\alpha} = \mathbf{1}$

Algorithm:

```
for each scale from coarse to fine
    approximate image from color and coarser scale
    compute confidence map from ᾱ and coarser scale
    compute level set from confidence map
    while mean confidence < 1 − ε
        for next target position p
            compute adaptive neighborhood N(p)
            search for most similar and frequent source match N(q)
            composite N(p) and N(q) at p, updating color and ᾱ
            compute approximation, confidence map and update level set
```

Figure 3: Image completion pseudocode.

3 Fast approximation

A fast estimate of the colors of the hidden region is generated by a simple iterative filtering process based on the known values. The estimated colors guide the search for similar neighborhoods, as described in Section 5. Our completion approach adds detail by example to the smooth result of the fast approximation. The process of approximating a given domain with values that "agree" with some known values is known as scattered data interpolation. Typically, the assumption is that the unknown data is smooth, and various methods aim at generating a smooth function that passes through or close to the sample points. In image space methods, the approximation can be based on simple discrete kernel methods that estimate a function f over a domain by fitting a simple model at each point such that the resulting estimated function $\hat{f}$ is smooth in the domain. Localization is achieved either by applying a kernel K that affects a neighborhood ε, or by a more elaborate weighting function. A simple iterative filtering method known as push-pull [Gortler et al. 1996] is to down-sample and up-sample the image hierarchically using a local kernel at multiple resolutions. In the coarser levels, the kernel filter affects larger regions and the values of the samples percolate, yielding smoother data. When applied to finer resolutions, the effect is localized and higher frequencies are approximated.

The above simple process is accelerated and refined by employing a multi-grid method. The image C is pre-multiplied by the in-

verse matte $\bar{C} = C\bar{\alpha}$, and the approximation begins with $\bar{C} + \alpha$. Let $l = L, \ldots, 1$ denote the number of levels in a pyramid constructed from the image. Starting from an image $Y^{l=L}_{t=0}$ of ones, the following operations are performed iteratively for $t = 1, \ldots, T(l)$:

$$Y^{l}_{t+1} = (Y^{l}_{t}\alpha + \bar{C})(*K_\varepsilon \downarrow)^{l}(\uparrow *K_\varepsilon)^{l}. \quad (1)$$

Equation (1) consists of re-introducing the known values $\bar{C}$, then l times down-sampling $\downarrow$ with a kernel K_ε, and l times up-sampling $\uparrow$ with a kernel K_ε. It is applied repeatedly (subscript t is incremented) until convergence $Y^{l}_{t+1} = Y^{l}_{t}$.

Applying this process for $l = L$ scales results in a first approximation $Y^{L}_{t=T(L)}$. Then (superscript l is decremented) the known values are re-introduced, and the process is repeated for $l - 1$ scales,

$$Y^{l-1}_{t=0} = Y^{l}_{t=T(l)}\alpha + \bar{C}. \quad (2)$$

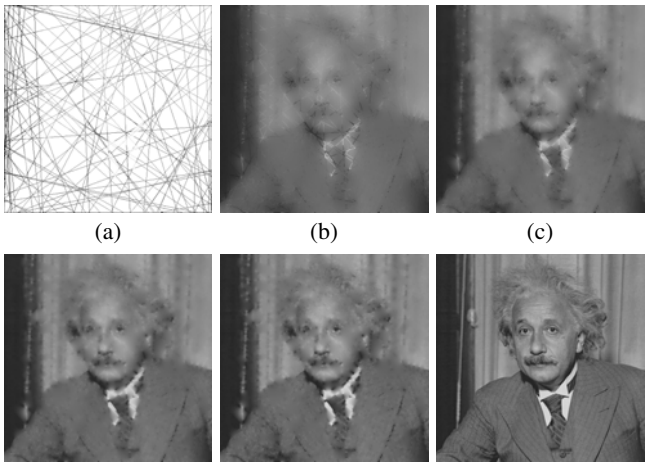
For $l = 1$, the approximation is $Y^{l=1}_{t=0} = Y^{l=2}_{t=T(2)}$, and the following iterations are performed:

$$Y^{1}_{t+1} = (Y^{1}_{t}\alpha + \bar{C}) * K_\varepsilon. \quad (3)$$

The final output is $C = Y^{1}_{T(1)}$.

As shown in Figure 4 the iterations applied to many levels (b-c), for large l, approximate the lower frequencies, while the iterations applied to few levels (d) or a single scale (e) handle higher frequencies. Table 1 summarizes the number of iterations, error, and computation time for each set of levels. The running times are for a 512 by 512 image, shown in Figure 4. Note that in the completion process, the bounding box of the unknown regions is typically much smaller, and decreases with each completion step.

Figure 5(a) shows an input image where part of the image is hidden by text, while the rest of the image is only visible through the text. The result of the approximation is shown in (b). The global root mean square error of luminance values in [0,1] is 0.049. Our completion method is based on image fragments, and therefore we are interested in the local error in a neighborhood around each pixel. This is illustrated in (c), which shows the inverse of the RMSE of luminance neighborhoods of radius 4 around each pixel (37 samples). The source image is shown in (d).



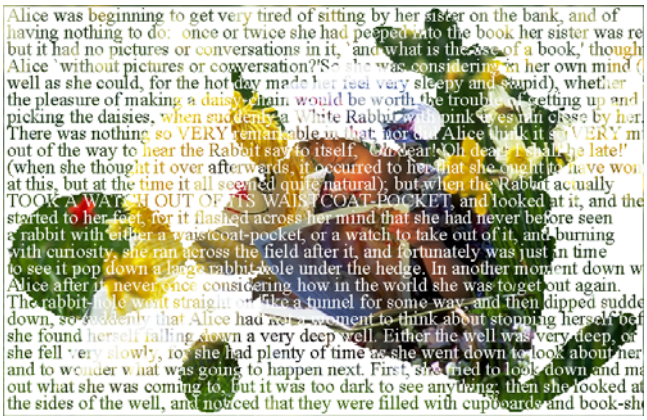
(a) (b) (c)

(d) (e) (f)

Figure 4: (a) Input, $\bar{C} + \alpha$: 100 lines are randomly sampled from a 512 by 512 image and superimposed on a white background. (b-e) $Y^{l}_{T(l)}$ for $l = 4, \ldots, 1$. (e) the result of our approximation. (f) source image.

# of levels	# of iterations	RMSE	Time (sec.)
4	197	0.0690	4.2
3	260	0.0582	5.3
2	739	0.0497	13.7
1	1269	0.0470	22.9

Table 1: Statistics and running times for the approximation in Figure 4. The levels correspond to Figure 4 (b-e), and the RMSE is of luminance in values [0,1]. Total computation time is 46.1 seconds for a 512 by 512 image.



(a)

(b)

(c) (d)

Figure 5: The input image in (a) is partly covered by a white text, while the rest of the image is only visible through the text. The result of our approximation is in (b). The RMSE of local neighborhoods in radius 4 is in (c) and the ground truth is in (d).

4 Confidence map and traversal order

In our setting we assume an inverse alpha matte that determines the areas to be completed. The matte assigns each pixel a value in $[0, 1]$.

We define a confidence map β by assigning a value to each pixel i:

$$\beta_i = \begin{cases} 1 & \text{if } \bar{\alpha}_i = 1 \\ \sum_{j \in N(i)} g_j \bar{\alpha}_j^2 & \text{otherwise,} \end{cases} \quad (4)$$

where N is a neighborhood around a pixel, g is a Gaussian falloff, and $\bar{\alpha}$ is the inverse matte. The map determines how much confidence to place in image information in each pixel as new information is generated during the course of the algorithm's progress, and is used for comparing and selecting fragments (Section 5).

In each scale, the image is traversed in an order that proceeds from high to low confidence, while maintaining coherence with previously synthesized regions. To compute the next target position from the confidence map, we set pixels that are greater than the mean confidence $\mu(\beta)$ to zero and add random uniform noise between 0 and the standard deviation $\sigma(\beta)$:

$$v_i = \begin{cases} 0 & \text{if } \beta_i > \mu(\beta) \\ \beta_i + \rho[0, \sigma(\beta)] & \text{otherwise.} \end{cases} \quad (5)$$

This defines a level set of candidate positions. At each iteration we retrieve the image coordinates of the pixel with maximal value in Eq. 5. This determines the position of the next search and composite. After compositing (Section 6), the set of candidate positions is updated by discarding positions in the set that are within the fragment. Once the set of candidate positions is empty it is recomputed from the confidence map. As the algorithm proceeds, the image is completed, $\mu(\beta) \to 1$ and $\sigma(\beta) \to 0$. The width of the zone from which the candidates for the next target position are selected, and the tolerance, decrease as the image is completed.

Figure 6 shows, from left to right, the inverse matte, confidence map, and level set, at two different time steps. The confidence map β is visualized on a logarithmic scale $h(\beta) = \lg(\beta + 1)$, with $h = 1$ shown in green, $h \leq 0.01$ in purple, and $0.01 < h < 1$ in shades of blue.

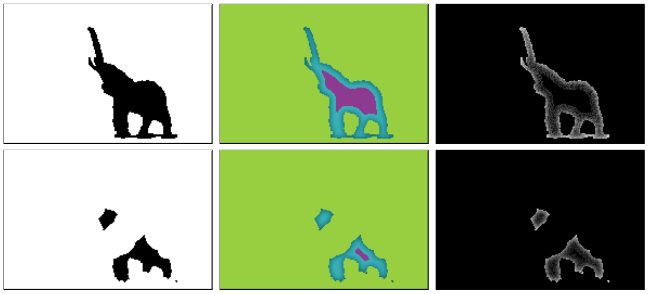


Figure 6: From left to right: inverse matte, visualization of confidence values on a logarithmic scale, and level set, at two different time steps.

5 Search

This section describes a 2D pattern matching method that considers the confidence of each pixel, in addition to the similarity of features used in traditional template matching. For each target fragment T, we search for the best source match S over all positions x, y, five scales l, and eight orientations θ: denoted as the parameter $r = (l, x, y, \theta)$. The algorithm adds detail to the approximated pixels ($\beta_i < 1$) by example without modifying the known pixels ($\beta_i = 1$). The confidence map is used for comparing pairs of fragments by considering corresponding pixels $s = S(i)$ and $t = T(i)$ in both target and source fragment neighborhoods N. For each pair of corresponding pixels, let β_s, β_t denote their confidence, and $d(s,t)$ denote the similarity between their features. The measure is L_1 and the features are color and luminance for all pixels, and gradients (1st order derivative in the horizontal, vertical, and diagonal directions) in the known pixels. We find the position, scale and orientation of the source fragment that minimizes the function:

$$r^* = \arg\min_r \sum_{s=S_r(i), t=T(i), i \in N} (d(s,t)\beta_s\beta_t + (\beta_t - \beta_s)\beta_t). \quad (6)$$

The first term of this function, $d(s,t)\beta_s\beta_t$, penalizes different values in corresponding pixels with high confidence in both the target and source fragments. The second term, $(\beta_t - \beta_s)\beta_t$, rewards pixels with a higher confidence in the source than in the target, while penalizing pixels with lower confidence in the source than in the target, normalized by the target confidence. The goal of these two terms is to select an image fragment that is both coherent with the regions of high confidence and contributes to the low confidence regions.

Source fragments are first trivially rejected based on luminance mean and variance. We consider a small set of $k = 5$ nearest neighbors according to Eq. 6 and take the most frequent, based on the luminance statistics.

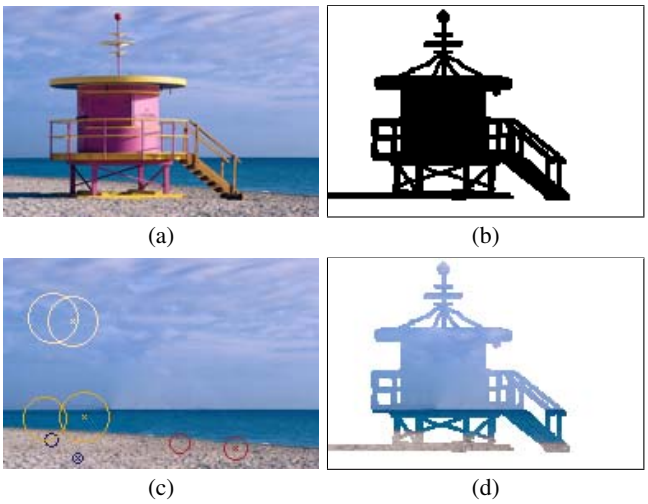


Figure 7: (a-b) Input color and inverse matte. (c-d) The result of our completion, several matching neighborhoods outlined with circles of the same color, the target center marked by a cross. Our approach completes the smooth areas with large fragments, the textured regions with smaller fragments, and the shoreline by searching in different *scales*.

Figures 7, 8 and 9 show pairs of matching neighborhoods outlined by the same color. The center of each target neighborhood is marked with a cross. A smaller or larger source neighborhood than target means a match across different scales. Note that in the completed region in Figure 7 (d) there are artifacts in the lower left portion, along the coast-line, where the dark and bright colors of opposite sides of the large missing region meet.

5.1 Adaptive neighborhood

An adaptive scheme to determine the size of the neighborhood is important for capturing features of various scales. Our algorithm uses existing image regions whose size is inversely proportional to the spatial frequency. We therefore tested several criteria to determine the adaptive size of a neighborhood [Gonzalez and Woods

Figure 8: Our approach completes structured texture in perspective by searching in different *scales*.

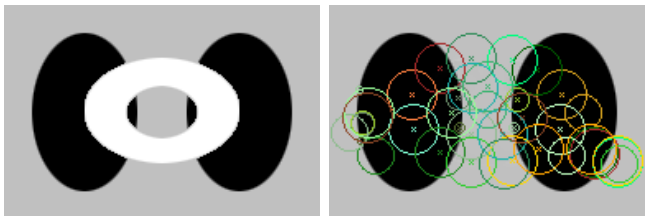
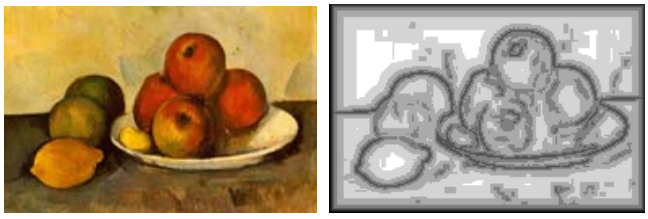


Figure 9: Our approach completes symmetric shapes by searching under *rotations and reflections*. Completion consists of a total of 21 fragments, marked on the output image, with mean radius of 14.

2002]. The first is the set of statistical moments of the luminance histogram of an element. If x is a random variable denoting luminance values and $p(x_i)$ the corresponding histogram for k distinct values, then the n-th moment of x about the mean m is $\mu_n(x) = \sum_{i=0}^{k}(x_i - m)^n p(x_i)$. The second is the measure of uniformity of an element $u = \sum_{i=0}^{k} p^2(x_i)$. The third is the average entropy $e = -\sum_{i=0}^{k} p(x_i) \lg p(x_i)$. Considering only a small number of radii 2^j for $j = 1,\ldots,l = 5$ is a quantization of the neighborhood size map. We have found that a simple contrast criterion, the absolute of difference between extreme values across channels, yields results which are nearly as good as the other, more elaborate measures that we have tried.

Allowing nearby matches of large smooth neighborhoods, while discarding nearby matches of smaller textured neighborhoods avoids smearing artifacts. Searching in scale factors above 1 only for large neighborhoods allows interpolation of smooth areas while avoiding blurring of detailed regions.



(a) (b)

Figure 10: An image (a) and its corresponding neighborhood size map (b), where brighter regions mark larger neighborhoods.

Figure 10 shows an image and its corresponding neighborhood size map. These values are updated every completion iteration. Each pixel in (b) reflects an estimate of the frequencies in a neighborhood around the corresponding pixel in (a). Regions with high frequencies are completed with small fragments whereas those with low frequencies are completed with larger fragments.

6 Compositing fragments

We would like to superimpose the selected source fragment S over the target fragment T such that the regions with high confidence seamlessly merge with the target regions with low confidence. Since we give priority to the target, we need to compose the source fragment "under" the target fragment. Taking the alpha values into consideration, we apply the compositing operator: T OVER S. To create a seamless transition between T and S we apply this operator in frequency bands.

The Laplacian pyramid [Burt and Adelson 1985] can be used to smoothly merge images according to binary masks. The color components C of each image A and B are decomposed into Laplacian pyramids L, and the binary masks M into Gaussian pyramids G. The Gaussian and Laplacian pyramids are separately multiplied for each image at corresponding scales k, and then added, to form a Laplacian pyramid of the merged image $L(C_{merge})$, which is then reconstructed,

$$L_k(C_{merge}) = L_k(C_A)G_k(M_A) + L_k(C_B)G_k(M_B). \quad (7)$$

An alpha value in $[0,1]$ is traditionally considered as either having partial coverage or as semi-transparent. The alpha channel, commonly represented by 8 bits, is used for capturing fractional occlusion to combine anti-aliased images. The classical associative operator for compositing a foreground element over a background element, F OVER B, is:

$$\begin{aligned} C_{out} &= C_F\alpha_F + C_B\alpha_B(1-\alpha_F) \quad (8)\\ \alpha_{out} &= \alpha_F + \alpha_B(1-\alpha_F). \end{aligned}$$

Porter and Duff [1984] pre-multiplied color by alpha, and the volume rendering ray casting integral is often discretely approximated by a sequence of OVER operations.

We represent color and alpha values as matrices and plug Eq. 8 into Eq. 7 to get the Laplacian OVER operator:

$$L_k(C_{out}) = L_k(C_F)G_k(\alpha_F) + L_k(C_B)G_k(\alpha_B)G_k(\mathbf{1}-\alpha_F). \quad (9)$$

The alpha values α_{out} are obtained by setting $C_F = C_B = \mathbf{1}$ in Eq. 9.

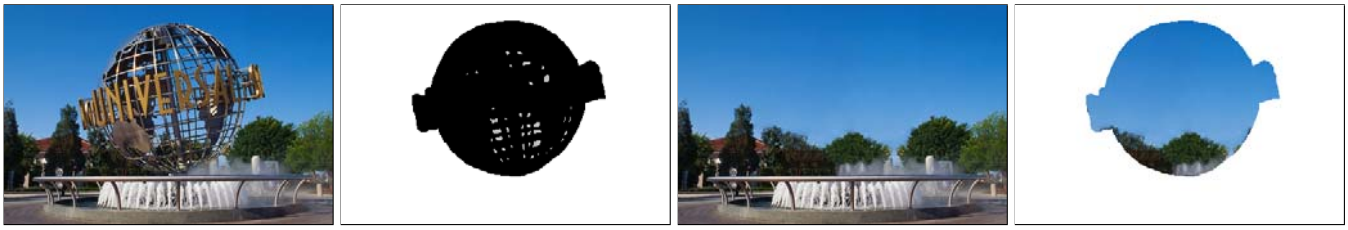
The Laplacian OVER operator uses a wide overlapping region for the low frequencies and a narrow overlap for the high frequencies. The OVER operator is traditionally used for compositing a figure over a background, while maintaining sharp transitions according to alpha values. We use the Laplacian OVER operator for compositing two similar fragments, one over the other, to create a smooth composite.

The output is a fragment R that is premultiplied, and is masked by a circular neighborhood. To insert this result into the image, we multiply the previous approximated image by its inverse matte $\bar{C} = C\bar{\alpha}$, and then put the fragment values $R(\bar{C})$ into $\bar{C}$ and $R(\bar{\alpha})$ into $\bar{\alpha}$. In the following iteration the approximation begins with these premultiplied values.

Fragment compositing is illustrated in Figure 12. The top row shows the matching neighborhoods (left) and the inverse matte (right). The alpha values, $G_k(\alpha_F)$ and $G_k(\alpha_B)$, are shown in the center row. The reconstructed color values of each intermediate term in Eq. 9 are shown on the bottom left. The premultiplied color and alpha output are shown on the bottom right. The output is finally masked by a circular neighborhood.

7 Implementation

Our implementation completes the image from coarse to fine. Aside from computational efficiency, coarse-to-fine completion is important for capturing features at several scales. Starting with a coarse scale corresponds to using a range of relatively large target



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

Figure 11: (a) The *Universal Studios* globe. (b) Inverse matte. (c) Completed image. (d) Content of the completed region.

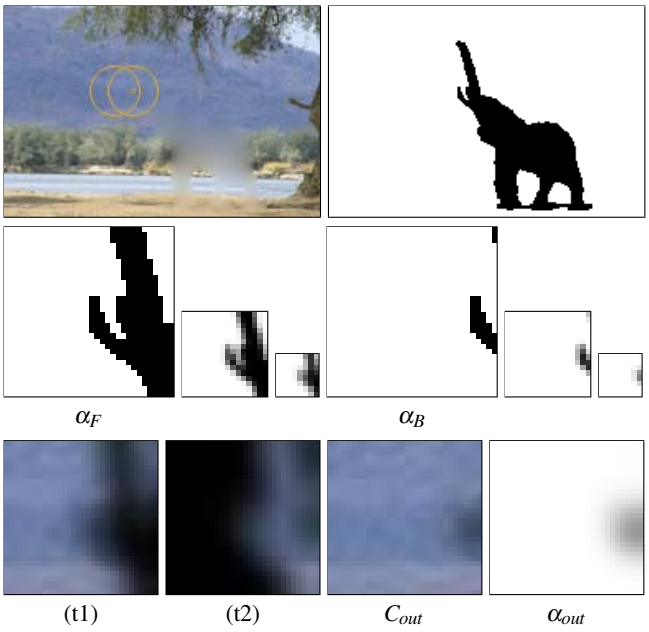


Figure 12: Fragment compositing: matching fragments (top left), inverse matte (top right), and Gaussian pyramids of the fragments alpha (center row). (t1) First term of Eq. 9. (t2) Second term of Eq. 9. The output color and alpha of compositing (bottom right).

fragments, and then using target fragments that become gradually smaller at every scale, finer and finer details are added to the completed image. The result of the coarse level C^l is up-sampled by bicubic interpolation and serves as an estimate for the approximation of the next level:

$$\hat{f}^{l+1} = \lambda \hat{f}^{l+1} + (1-\lambda)(C^l \uparrow *),$$

where $\lambda = 0.5$. The confidence values of the next level are also updated:

$$\beta^{l+1} = \lambda \beta^{l+1} + (1-\lambda)(\beta^l \uparrow),$$

where the confidence values of the coarse level are $\beta^l = \mathbf{1}$ upon completion.

The following are implementation details. We perform completion using two scales, with 192 by 128 and 384 by 256 resolution. To create a conservative inverse matte for the fine level, the inverse matte of the coarse level is up-sampled by nearest neighbors and dilated. The approximation described in Section 3 is a local process, and therefore performed in the bounding box of the unknown regions. We use three scales $L = 3$ in Eq. 2, a Gaussian kernel with standard deviation 1 in Eq. 1, and 0.85 for the final pass in Eq. 3. It is wasteful to run the approximation until convergence since the error diminishes non-linearly. First, we choose a representative subset of $\sqrt{N}$ random positions in the unknown regions, where N is the number of pixels. At each set of levels in Eq. 1 the approximation stops when the $\frac{\sqrt{N}}{L-l+1}$ representative values converge. The search based on Eq. 6 is performed at five scales with factors spaced equally between 0.75 and 1.25, at each x,y position, and eight orientations (four rotations in increments of 90 degrees, and their four reflections). Compositing in Eq. 10 is performed using at most three levels k in the Gaussian and Laplacian pyramids for the largest fragments. The completion process terminates when $\mu(\beta) \geq 1-\varepsilon$, as described in the pseudocode in Figure 3, and we set $\varepsilon = 0.05$.

8 Results

We have experimented with the completion of various photographs and paintings with an initial mean confidence $\mu(\beta) > 0.7$. The computation times range between 120 and 419 seconds for 192 by 128 images, and between 83 and 158 minutes for 384 by 256 images, on a 2.4 GHz PC processor. Slightly over 90 percent of the total computation time is spent on the search for matching fragments. Note that computation time is quadratic in the number of pixels.

Figure 1 shows a 384 by 256 image with $\mu(\beta) = 0.876$. The image consists of various layers of smooth and textured regions. Our method synthesizes textures of different scale and completes the shorelines. Completion consists of 137 synthesized fragments (of them 26 for the coarse level), and total computation time is 83 minutes (220 seconds for the coarse level).

Figure 11 shows a 384 by 256 image of the *Universal Studios* globe with $\mu(\beta) = 0.727$. In the center of the globe there are pixels which are used for completion. However, the color is contaminated by the neighboring pixels of the globe. Marking these pixels as completely known in the matte creates artifacts in the completed regions. An alternative is to assign values that are less than 1 to the corresponding matte positions, as shown in (b). This way, these pixels are not totally discarded, and are taken as strong hints to the location of the smooth area and the textured regions. Completion consists of 240 synthesized fragments (of them 57 for the coarse level), and total computation time is 158 minutes (408 seconds for the coarse level).

Figure 13 shows more results for 192 by 128 images with $\mu(\beta) > 0.87$. Our approach completes shapes, smooth and textured regions, and various layers of both, as well as transparency. Completion requires between 20 and 53 synthesized fragments, with average neighborhood radii between 6 and 14. Statistics for each image appear in Table 2.

Image	# of fragments	$\mu(\beta)$	$\sigma(\beta)$	Time (sec.)
Golfer	50	0.931	0.221	414
Pyramids	20	0.951	0.204	129
Whale	26	0.934	0.227	184
Hollywood	34	0.968	0.146	120
The Raw Nerve by Magritte	26	0.950	0.181	149
Still Life by Cezanne	53	0.874	0.321	419

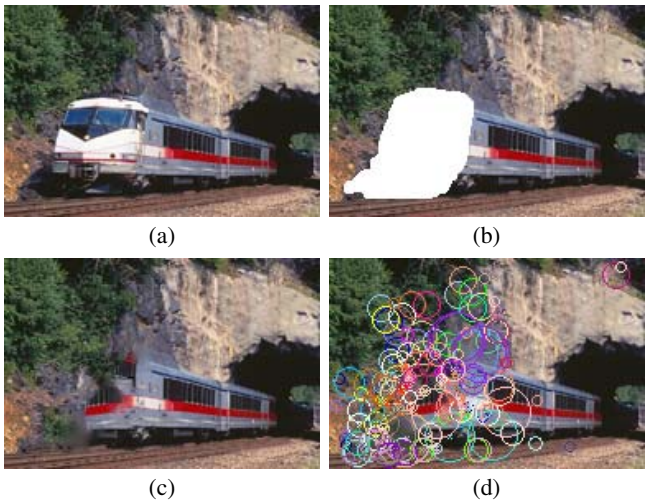
Table 2: Statistics and running times for completion of the images in Figure 13.

9 Limitations

Our approach to image completion is example-based. As such, its performance is directly dependent on the richness of the available fragments. In all the examples presented in this paper, the training set is the known regions in a single image, which is rather limited. The building blocks for synthesizing the unknown region are image fragments of the known region. To utilize the training set, new fragments are synthesized by combining irregular parts of fragments, applying combinations of transformations to fragments (scale, translation, rotation, and reflection), and compositing fragments together.

To evaluate the performance of our method, we can use examples where the unknown region is available, and quantify and measure the completion with respect to the ground truth. Specifically, we can use the ground truth data for the search and reconstruct it in the composite, calculating the best alpha for the blend that reconstructs the ground truth.

Our technique is an image-based 2D method. It has no knowledge of the underlying 3D structure in the image. For example, in Figure 14, the front end of the train is removed. Our method completes the image as shown in (c). The matching fragments of completion are shown in (d).

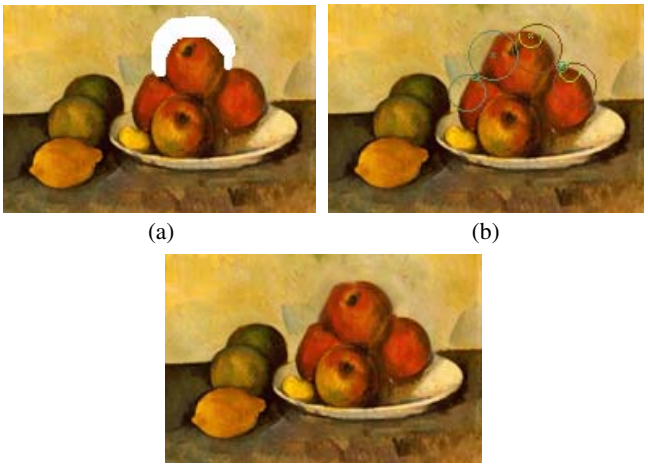


(a) (b) (c) (d)

Figure 14: Our approach is an image-based 2D technique and has no knowledge of the underlying 3D structures. (a) Input image. (b) The front of the train is removed. (c) The image as completed by our approach. (d) Matching fragments marked on output.

Note that even two fragments of the same part of an object may differ greatly under slight illumination changes and transformations. While this is obviously a notable limitation, it is also an advantage as it is based on a simple mechanism, that does not require building models from a single image. An alternative to an automatic image completion is to let the user manually build an image-based model and apply photo-editing operations with some 3D knowledge as in [Oh et al. 2001].

Our image completion approach does not distinguish between figure and ground. This presents a limitation for completion when the inverse matte is on the boundary of a figure, since both the figure and background can be synthesized by example. For example, in Figure 15(a), the unknown region meets the boundary of the figure of the apple. Our approach does not handle these cases. However, note that the known regions of this painting contain similar patterns, and the search finds matches under combinations of scale and orientation in (b), and the completion results in (c). Note the ghost artifact on the right portion of the completed figure.



(a) (b) (c)

Figure 15: Our approach does not handle cases in which the unknown region is on the boundary of a figure as in (a). However, the known regions of this painting contain similar patterns, and the search finds matches under combinations of scale and orientation in (b), and the completion results in (c).

Our approach does not handle ambiguities in which the missing area covers the intersection of two perpendicular regions as shown in Figure 16.

(a) (b) (c)

Figure 16: Our approach does not handle ambiguities such as shown in (a), in which the missing area covers the intersection of two perpendicular regions as shown above. (b) The same ambiguity in a natural image. (c) The result of our completion.

Visual grouping and background-foreground separation are an open problem, usually addressed by a subset of the Gestalt principles (proximity, similarity, good continuation, closure, smallness, common fate, symmetry and pragnanz) [Koffka 1935, 1967]. These principles can be incorporated into a photo-editing tool, in which it is practical to give the user some degree of control over the completion process, provided it be done in a simple and intuitive fashion. By specifying a *point of interest* in the image or a direction, the user can optionally favor symmetry, proximity, or the horizontal or vertical axis. This requires adding bias to the search in Eq. 6 by a

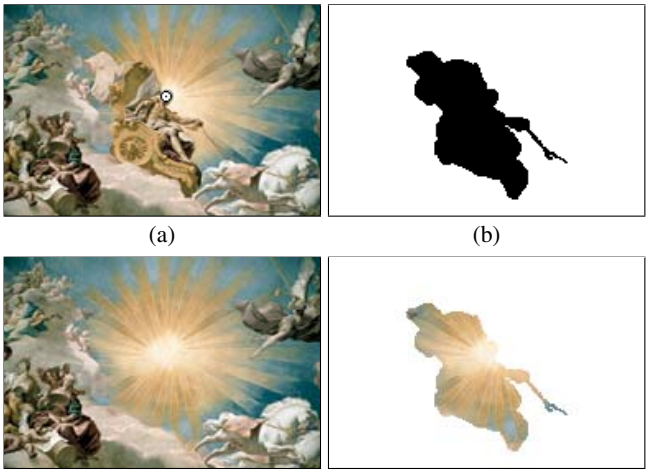


Figure 17: A *point of interest* is specified with a white circle near the center of the image in (a). Our result using the point of interest to complete the circularly symmetric shape is shown in (c).

function of the horizontal or vertical distances between target and source fragments, or by their distance to the point of interest. Guiding the traversal order requires adding a wide Gaussian centered at a point or a soft directional gradient to Eq. 5. Figure 17 shows completion using a point of interest located near the center of the image.

10 Summary and future work

We have introduced a new method for image completion that interleaves a smooth approximation with detail completion by example fragments. Our method iteratively approximates the unknown region and fills in the image by adaptive fragments. Our approach completes the image by a composition of fragments under combinations of spatial transformations.

To improve completion, future work will focus on the following: (i) Performing an anisotropic filtering pass in the smooth approximation, by computing an elliptical kernel, oriented and non-uniformly scaled at each point, based on a local neighborhood; (ii) Locally completing edges in the image based on elasticity, and then using the *completed* edge map in the search; and (iii) Direct completion of image gradients, reconstructing the divergence by the full multi-grid method.

In addition, we would like to extend our input from a single image to classes of images. Finally, increasing dimensionality opens exciting extensions to video and surface completion.

Acknowledgments

This work was supported in part by a grant from the Israeli Ministry of Science, a grant from the Israeli Academy of Science (Center of Excellence in Geometry), and a grant from the AMN foundation.

References

ASHIKHMIN, M. 2001. Synthesizing natural textures. In *ACM Symposium on Interactive 3D Graphics*, 217–226.

BAKER, S., AND KANADE, T. 2000. Limits on super-resolution and how to break them. In *IEEE Conference on Computer Vision and Pattern Recognition*, 372–379.

BERTALMIO, M., SAPIRO, G., CASELLES, V., AND BALLESTER, C. 2000. Image inpainting. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press, 417–424.

BERTALMIO, M., VESE, L., SAPIRO, G., AND OSHER, S. 2003. Simultaneous structure and texture image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition, to appear*.

BORENSTEIN, E., AND ULLMAN, S. 2002. Class-specific, top-down segmentation. In *European Conference on Computer Vision*, 109–124.

BROOKS, S., AND DODGSON, N. 2002. Self-similarity based texture editing. *ACM Transactions on Graphics, 21*, 3, 653–656.

BURT, P. J., AND ADELSON, E. H. 1985. Merging images through pattern decomposition. *Applications of Digital Image Processing VIII 575*, 173–181.

CHAN, T., AND SHEN, J. 2001. Mathematical models for local nontexture inpainting. *SIAM Journal on Applied Mathematics 62*, 3, 1019–1043.

CHUANG, Y.-Y., AGARWALA, A., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2002. Video matting of complex scenes. *ACM Transactions on Graphics, 21*, 3, 243–248.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, 341–346.

EFROS, A., AND LEUNG, T. 1999. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, 1033–1038.

FREEMAN, W. T., PASZTOR, E. C., AND CARMICHAEL, O. T. 2000. Learning low-level vision. *International Journal of Computer Vision 40*, 1, 25–47.

FREEMAN, W. T., JONES, T. R., AND PASZTOR, E. 2002. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 56–65.

GONZALEZ, R. C., AND WOODS, R. E. 2002. *Digital Image Processing*. Prentice Hall.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The lumigraph. In *Proceedings of ACM SIGGRAPH 96*, ACM Press, 43–54.

GUY, G., AND MEDIONI, G. 1996. Inferring global perceptual contours from local features. *IEEE International Journal of Computer Vision*, 1–2, 113–133.

HAEBERLI, P. 1990. Paint by numbers: Abstract image representations. In *Computer Graphics (Proceedings of ACM SIGGRAPH 90)*, ACM Press, 207–214.

HEEGER, D. J., AND BERGEN, J. R. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of ACM SIGGRAPH 95*, ACM Press, 229–238.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, 327–340.

HIRANI, A. N., AND TOTSUKA, T. 1996. Combining frequency and spatial domain information for fast interactive image noise removal. In *Proceedings of ACM SIGGRAPH 96*, ACM Press, 269–276.

IGEHY, H., AND PEREIRA, L. 1997. Image replacement through texture synthesis. In *IEEE International conference on Image Processing*, vol. 3, 186–189.

KOFFKA, K. 1935, 1967. *Principles of Gestalt Psychology*. New York, Hartcourt, Brace and World.

NOE, A., PESSOA, L., AND THOMPSON, E. 1998. Finding out about filling-in: A guide to perceptual completion for visual science and the philosophy of perception. *Behavioral and Brain Sciences*, 6, 723–748, 796–802.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, 433–442.

PALMER, S. 1999. *Vision Science*. MIT Press.

PORTER, T., AND DUFF, T. 1984. Compositing digital images. In *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, 253–259.

SHARON, E., BRANDT, A., AND BASRI, R. 2000. Completion energies and scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 10, 1117–1131.

SOLER, C., CANI, M.-P., AND ANGELIDIS, A. 2002. Hierarchical pattern mapping. *ACM Transactions on Graphics, 21*, 3, 673–680.

WEI, L. Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press, 479–488.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. *ACM Transactions on Graphics, 21*, 3, 277–280.

WILLIAMS, L., AND JACOBS, D. W. 1997. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation 9*, 4, 837–858.

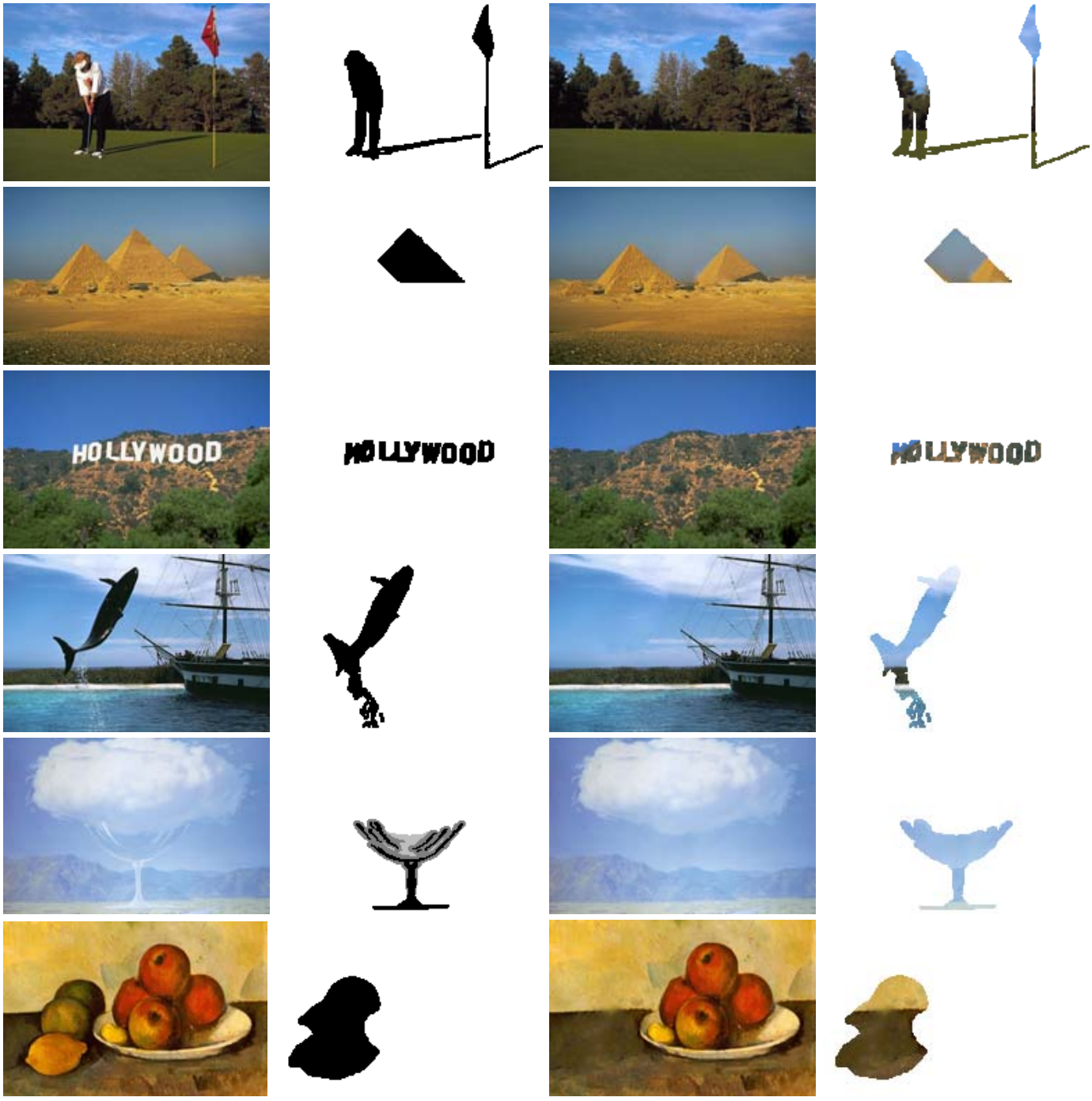


Figure 13: Completion results for some photographs and well-known paintings.