Scene Completion Using Millions of Photographs

James Hays   Alexei A. Efros
Carnegie Mellon University
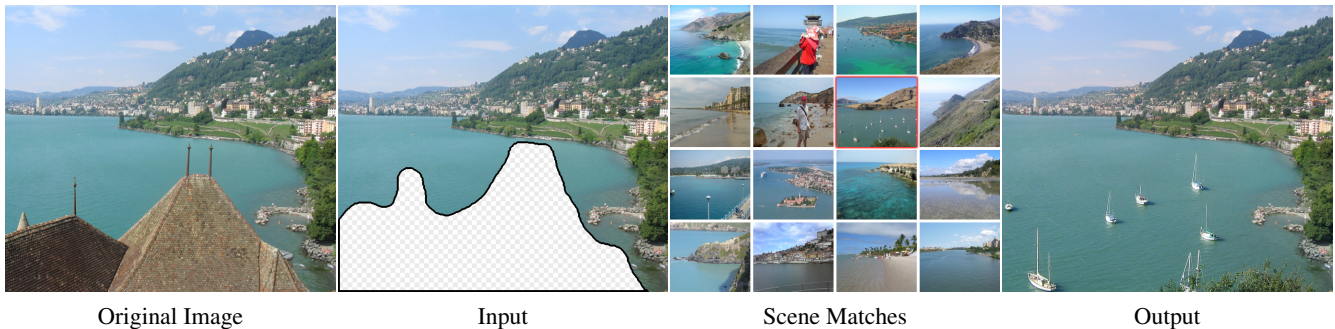
| Original Image | Input | Scene Matches | Output |

Figure 1: Given an input image with a missing region, we use matching scenes from a large collection of photographs to complete the image.

Abstract

What can you do with a million images? In this paper we present a new image completion algorithm powered by a huge database of photographs gathered from the Web. The algorithm patches up holes in images by finding similar image regions in the database that are not only seamless but also semantically valid. Our chief insight is that while the space of images is effectively infinite, the space of semantically differentiable scenes is actually not that large. For many image completion tasks we are able to find similar scenes which contain image fragments that will convincingly complete the image. Our algorithm is entirely data-driven, requiring no annotations or labelling by the user. Unlike existing image completion methods, our algorithm can generate a diverse set of results for each input image and we allow users to select among them. We demonstrate the superiority of our algorithm over existing image completion approaches.

Keywords: Image Completion, Image Database, Image Compositing, Inpainting, Hole Filling

1 Introduction

Every once in a while, we all wish we could erase something from our old photographs. A garbage truck right in the middle of a charming Italian piazza, an ex-boyfriend in a family photo, a political ally in a group portrait who has fallen out of favor [King 1997]. Other times, there is simply missing data in some areas of the image. An aged corner of an old photograph, a hole in an image-based 3D reconstruction due to occlusion, a dead bug on the camera lens. Image completion (also called inpainting or hole-filling) is the task of filling in or replacing an image region with new image data such that the modification can not be detected.

There are two fundamentally different strategies for image completion. The first aims to reconstruct, as accurately as possible, the data that *should have been* there, but somehow got occluded or corrupted. Methods attempting an accurate reconstruction have to use some other source of data in addition to the input image, such as video (using various background stabilization techniques, e.g. [Irani et al. 1995]) or multiple photographs of the same physical scene [Agarwala et al. 2004; Snavely et al. 2006].

The alternative is to try finding a plausible way to fill in the missing pixels, hallucinating data that *could have been* there. This is a much less easily quantifiable endeavor, relying instead on the studies of human visual perception. The most successful existing methods [Criminisi et al. 2003; Drori et al. 2003; Wexler et al. 2004; Wilczkowiak et al. 2005; Komodakis 2006] operate by extending adjacent textures and contours into the unknown region. This idea is derived from example-based texture synthesis [Efros and Leung 1999; Efros and Freeman 2001; Kwatra et al. 2003; Kwatra et al. 2005], sometimes with additional constraints to explicitly preserve Gestalt cues such as *good continuation* [Wertheimer 1938], either automatically [Criminisi et al. 2003] or by hand [Sun et al. 2005]. Importantly, all of the existing image completion methods operate by filling in the unknown region with content from the known parts of the input source image.

Searching the source image for usable texture makes a lot of sense. The source image often has textures at just the right scale, orientation, and illumination as needed to seamlessly fill in the unknown region. Some methods [Drori et al. 2003; Wilczkowiak et al. 2005] search additional scales and orientations to gain additional source texture samples. However, viewing image completion as constrained texture synthesis limits the type of completion tasks that can be tackled. The assumption present in all of these methods is that all the necessary image data to fill in an unknown region is located somewhere else in *that same image*. We believe this assumption is flawed and that the source image simply doesn't provide enough data except for trivial image completion tasks.

Typical demonstrations of previously published algorithms are object removal tasks such as removing people, signs, horses, or cars from relatively simple backgrounds. The results tend to be fairly sterile images because the algorithms are only reusing visual content that appeared somewhere else in the same image. For situations in which the incomplete region is not bounded by texture regions, or when there is too little useful texture, existing algorithms have trouble completing scenes (Figure 2).

| Original Image | Input | Criminisi et al. | MS *Smart Erase* | Wilczkowiak et al. | Our algorithm |

Figure 2: Results from various image completion algorithms including Microsoft Digital Image Pro *Smart Erase*.

2   Overview

In this paper we perform image completion by leveraging a massive database of images. There are two compelling reasons to expand the search for image content beyond the source image. 1) In many cases, a region will be impossible to fill plausibly using only image data from the source image. For instance, if the roof of a house is missing or the entire sky is masked out. 2) Even if there is suitable content in the source image, reusing that content would often leave obvious duplications in the final image, *e.g.* replacing a missing building with another building in the image. By performing hole filling with content from other images, entirely novel objects and textures can be inserted.

However, there are several challenges with drawing content from other images. The first challenge is computational. Even in the single image case, some existing methods report running times in the hours [Efros and Leung 1999; Drori et al. 2003] because of the slow texture search. Texture synthesis-based image completion methods are difficult to accelerate with exact or approximate nearest-neighbor methods because of the high dimensionality of the features being searched and because the valid dimensions of the feature being matched will change depending on the shape of the unknown region at each iteration.

The second challenge is that as the search space increases, there is higher chance of a synthesis algorithm finding locally matching but semantically invalid image fragments. Existing image completion methods might produce sterile images but they don't risk putting an elephant in someone's back yard or a submarine in a parking lot.

The third challenge is that content from other images is much less likely to have the right color and illumination to seamlessly fill an unknown region compared to content from the same image. Therefore, we need a robust seam-finding and blending method to make our image completions look plausible.

In this work we alleviate both the computational and semantic challenges with a two-stage search. We first try to find images depicting semantically similar scenes and then use only the best matching scenes to find patches which match the context surrounding the missing region. Scene matching reduces our search from a database of over two million images to a manageable number of best matching scenes (200 in our case) which are used for image completion. We use a low dimensional scene descriptor [Oliva and Torralba 2006] so it is relatively fast to find the nearest scenes, even in a large database. Our approach is purely data-driven, requiring no labelling or supervision.

In order to seamlessly combine image regions we employ Poisson blending [Perez et al. 2003]. To avoid blending artifacts, we first perform a graph cut segmentation to find the boundary for the Poisson blending that has the minimum image gradient magnitude. This is in contrast to minimizing the intensity domain difference along a boundary [Wilczkowiak et al. 2005] or other heuristics to encourage a constant intensity offset for the blending boundary [Jia

et al. 2006]. In section 4 we explain why minimizing the seam gradient gives the most perceptually convincing compositing results.

The image completion work most closely resembling our own, Wilczkowiak et al. [2005], also demonstrates the search of multiple images. However, in their case it was only a few images that were hand selected to offer potentially useful image regions. Also related are methods which synthesize semantically valid images either from text or image constraints using image databases with labelled regions [Diakopoulos et al. 2004; Johnson et al. 2006]. However, the database labelling process must be supervised [Diakopoulos et al. 2004] or semi-supervised [Johnson et al. 2006].

3   Semantic Scene Matching

Since we do not employ user-provided semantic constraints or a labelled database, we need to acquire our semantic knowledge from the data directly. This requires us to sample the set of visual images as broadly as possible. We constructed our image collection by downloading all of the photographs in thirty Flickr.com groups that focus on landscape, travel, or city photography. Typical group names are "lonelyplanet," "urban-fragments," and "ruraldecay." We also downloaded images based on keyword searches such as "outdoors," "vacation," and "river." We discarded duplicate images, as well as images that are too small. All images were down-sampled, if necessary, such that their maximum dimension was 1024 pixels. Our database downloading, pre-processing, and scene matching were all distributed among a cluster of 15 machines. In total, we acquired about 2.3 million unique images (396 gigabytes of JPEG compressed data).

In order to successfully complete images we need to find image regions in our database that are not just seamless and properly textured but also *semantically* valid. We do not want to complete hillsides with car roofs or have ducks swimming in city pavement which locally resembles a lake. To help avoid such situations we first look for *scenes* which are most likely to be semantically equivalent to the image requiring completion. The use of scene matching is the most important element of our image completion method. Without it, our search would not be computationally feasible and our image completion results would rarely be semantically valid. Our scene matching, in combination with our large database, allows us to do image completion without resorting to explicit semantic constraints as in previous photo synthesis work [Diakopoulos et al. 2004; Johnson et al. 2006].

We use the gist scene descriptor [Torralba et al. 2003; Oliva and Torralba 2006] which has been shown to perform well at grouping semantically similar scenes (eg. city, tall buildings, office, fields, forest, beach) and for place recognition. It must be noted that a low-level scene descriptor in and of itself cannot encode high-level semantic information. Scenes can only be trusted to be semantically similar if the distance between them is very small. The way we address this issue is by collecting a huge dataset of images, making it more likely that a very similar scene to the one being searched
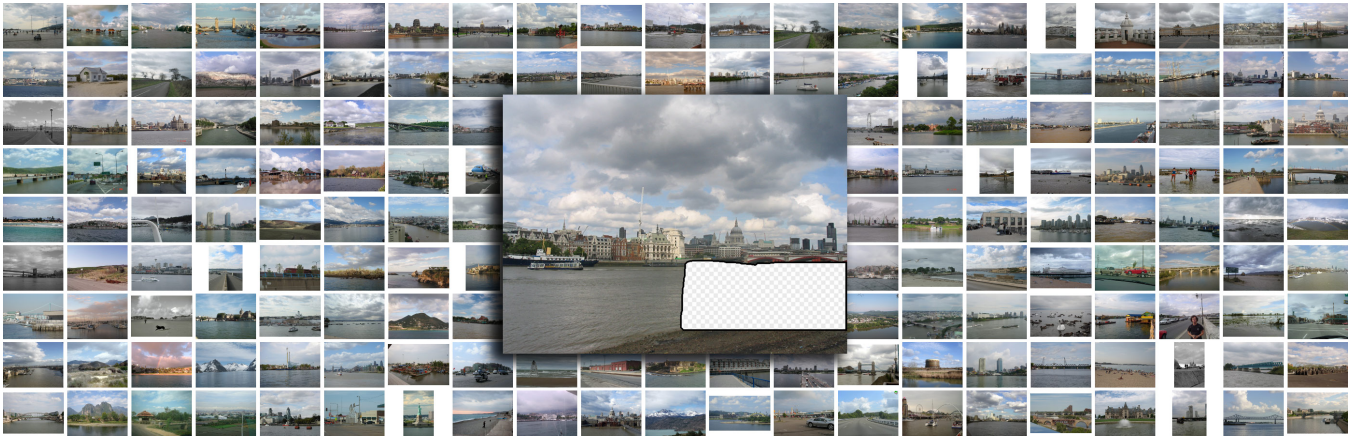
Figure 3: The 164 closest scenes to the incomplete image in the center. Most of the scenes are semantically and structurally similar; many are even from the same city (London).

is available in the dataset. Indeed, our initial experiments with the gist descriptor on a dataset of ten thousand images were very discouraging. However, increasing the image collection to two million yielded a qualitative leap in performance (see Figure 3 for a typical scene matching result). Independently, Torralba et al. [2007] have observed a similar effect with a dataset of up to 70 million tiny (32x32) images.

The gist descriptor aggregates oriented edge responses at multiple scales into very coarse spatial bins. We found a gist descriptor built from 6 oriented edge responses at 5 scales aggregated to a 4x4 spatial resolution to be most effective. We augment the scene descriptor with the color information of the query image downsampled to the spatial resolution of the gist.

Searching for similar scenes first rather than directly looking for similar patches speeds up our search dramatically. Instead of looking for image patches in all two million images at multiple offsets and scales we can instead eliminate 99.99% of the database quickly by finding the nearest neighbor scenes based on the relatively low-dimensional scene descriptor.

Given an input image to be hole-filled, we first compute its gist descriptor with the missing regions excluded. This is done by creating a mask which weights each spatial bin in the gist in proportion to how many valid pixels are in that bin. We calculate the SSD between the the gist of the query image and every gist in the database, weighted by the mask. The color difference is computed in the L*a*b* color space. The differences are weighted such that the gist contributes roughly twice as much as the color information to the final distance calculation.

Because we match scenes using arbitrary dimensions of the descriptor (depending on which regions of a query image are missing), we can not use PCA dimensionality reduction as suggested in [Oliva and Torralba 2006]. For the same reason we do not use any approximate nearest-neighbor scheme since they tend to incur large relative errors when matching on arbitrary dimensions of descriptors with hundreds of dimensions. However, the descriptors are small enough to exhaustively search in a few minutes using a cluster of 15 machines.

4   Local Context Matching

Having constrained our search to semantically similar scenes we can use traditional template matching to more precisely align those matching scenes to the local image context around the missing region. We define the local context to be all pixels within an 80 pixel radius of the hole's boundary. This context is compared against the 200 best matching scenes across all valid translations and 3 scales (.81, .90, 1) using pixel-wise SSD error in L*a*b* color space. Only placements (translations and scales) for which the context is

fully contained in the matching scene are considered. Since we expect our matching scenes to already be roughly aligned with the query image, we discourage distant matches by multiplying the error at each placement by the magnitude of its translational offset. The translation and scale with the minimum weighted SSD error is chosen as the best placement for each matching scene. In addition to the pixel-wise alignment score, we also compute a texture similarity score to measure coarse compatibility of the proposed fill-in region to the source image within the local context. A simple texture descriptor is computed as a 5x5 median filter of image gradient magnitude at each pixel, and the descriptors of the two images are compared via SSD.

We composite each matching scene into the incomplete image at its best placement using a form of graph cut seam finding [Kwatra et al. 2003] and standard poisson blending [Perez et al. 2003]. Using a seam finding operation to composite the images is arguably undesirable for hole filling since a user might want to preserve all of the image data in the input image. Past image completion algorithms [Criminisi et al. 2003] have treated the remaining valid pixels in an image as hard constraints which are not changed. We relax this, as in [Wilczkowiak et al. 2005], and allow the seam-finding operation to remove valid pixels from the query image but we discourage the cutting of too many pixels by adding a small cost for removing each pixel in the query image which increases with distance from the hole (Equation 2).

When performing a seam-finding operation and gradient domain fusion in sequence it makes sense to optimize the seam such that it will minimize the artifacts left behind by the gradient domain fusion. Jia et al. [2006] use dynamic programming to find a seam which has minimum intensity difference between the two images after allowing some constant intensity offset. The intuition is that humans are not sensitive to relatively large shifts in color and intensity as long as the shifts are seamless and low frequency.

We argue that it is better to minimize the gradient of the image difference along the seam instead of intensity differences. While the heuristic in [Jia et al. 2006] will indeed minimize the seam gradient if it succeeds in finding a constant color offset seam, that is impossible in many compositing situations (Figure 4). Minimizing intensity domain differences (even after optimizing for the best global color offset) can cause the seam to pass through many high frequency edges which will leave obvious blending artifacts after Poisson blending (Figure 4a). If we instead minimize the gradient of the image difference along the seam (Figure 4c), the seam tends to pass through regions of the images which either match (and thus have low absolute difference and thus low difference gradient) or are both constant colored (in which case poisson blending will hopefully do a good job of hiding the color difference at low

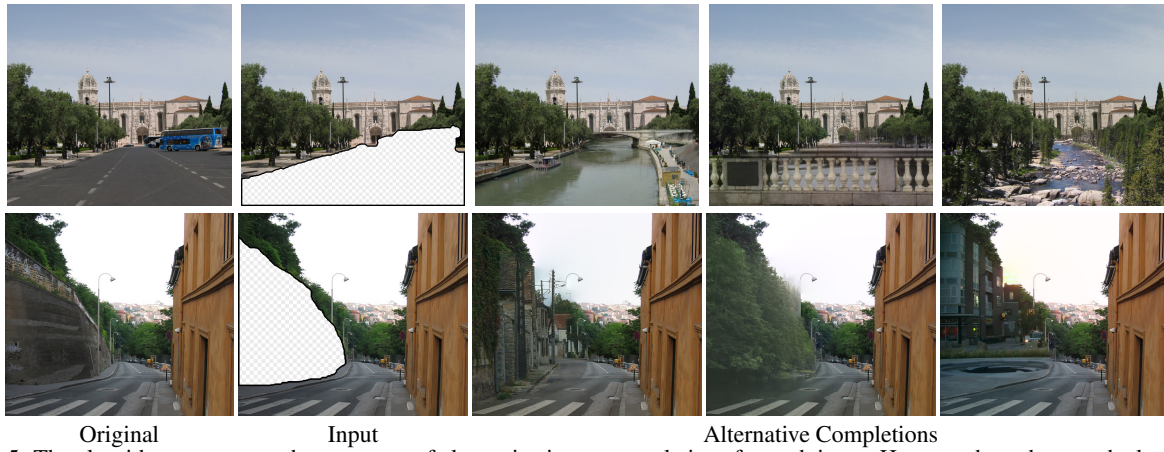Original   Input   Alternative Completions

Figure 5: The algorithm presents to the user a set of alternative image completions for each input. Here we show three such alternatives.
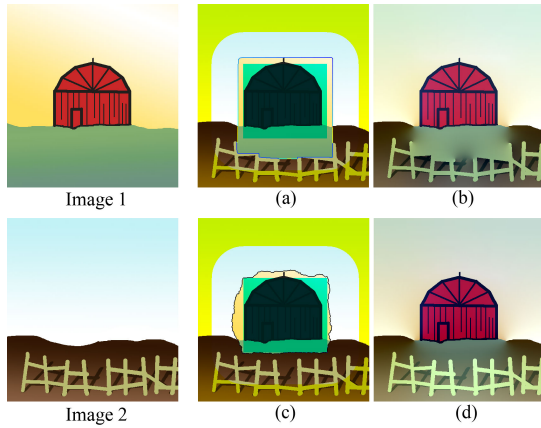

Figure 4: In this synthetic compositing example, we are placing the barn from Image 1 into the field of Image 2. (a) shows the optimal seam which minimizes intensity differences (even after some best, constant color shift). (c) shows the optimal seam which minimizes the gradient of the image difference. (b) and (d) are the Poisson blending results from (a) and (c) respectively.

frequencies). We find the seam by minimizing the following cost function:

$$C(L) = \sum_p C_d(p, L(p)) + \sum_{p,q} C_i(p, q, L(p), L(q)) \qquad (1)$$

Where $C_d(p, L(p))$ are the unary costs of assigning any pixel, p, to a specific label $L(p)$, and $C_i(p, q, L(p), L(q))$ is the higher-order cost for two pixels p and q taking labels $L(p)$ and $L(q)$ respectively. The label of each pixel, $L(p)$, has two values *patch* and *exist* corresponding to a pixel being taken from the scene match or the incomplete image. For missing regions of the existing image, $C_d(p, exist)$ is a very large number. For regions of the image not covered by the scene match, $C_d(p, patch)$ is very large. For all other pixels,

$$C_d(p, patch) = (k * Dist(p, hole))^3 \qquad (2)$$

This small penalty assigns a cost to each pixel taken from the scene match which increases with a pixel's distance from the hole, $Dist(p, hole)$. We use an empirically found $k = .002$. Wilczkowiak et al. [2005] use a similar penalty.

$C_i(p, q, L(p), L(q))$ is non-zero only for immediately adjacent, four-way connected pixels. When neighbors p and q share the same label, $L(p) = L(q)$, the cost is also zero. Along the seams, when $L(p) \neq L(q)$, $C_i(p, q, L(p), L(q)) = \nabla diff(p, q)$ where $\nabla diff(p, q)$

is the magnitude of the gradient of the SSD between the existing image and the scene match at pixels p and q.

Using this cost function has the added advantage that it can be globally minimized quickly using a graph cut [Boykov et al. 2001] unlike the heuristic in [Jia et al. 2006] which has to iterate until convergence while estimating the best color offset. A very similar metric was also mentioned but not demonstrated in [Agarwala et al. 2004]. Like them, we allow the Poisson blending to operate on the entire image domain instead of correcting just the composited region. We use the poisson solver of [Agrawal et al. 2006].

Finally we assign each composite a score which is the sum of the scene matching distance, the local context matching distance, the local texture similarity distance, and the cost of the graph cut. All four components of the score are scaled to contribute roughly equally. We present the user with the 20 composites with the lowest scores.

5 Results and Comparison

Image completion is an inherently under-constrained problem. There are many viable ways to fill a hole in an image. Previous approaches, which operate by reusing texture from the input image can offer relatively few alternative completions (perhaps by changing parameters such as the patch size). While some such results might look slightly better than others, the semantic interpretation of the image is unlikely to change. On the other hand, our algorithm can offer a variety of semantically valid image completions for a given query image (e.g. Figure 5). After compositing the best-matching patches we present a user with the 20 top image completions (roughly equivalent to one page of image search results). In some cases, many of these completions are of acceptable quality and the user can select the one(s) they find most compelling. In other cases, only a few or none of the results are acceptable. The quality of our results benefits from this very simple user interaction and it is difficult for conventional image completion methods to offer an analogous selection of results.

Some of our image completions are shown in Figure 6. The 4th result from the top is interesting because the scaffolding on the cathedral that was masked out has been replaced with another image patch of the same cathedral. The database happened to contain an image of the same cathedral from a similar view. It is not our goal to complete scenes and objects with their *true* selves in the database, but with the availability of increasingly large photo collections such fortuitous events do occur.

When our algorithm is successful, the completion is semantically valid, although there might be slight low-level artifacts such as resolution mismatch, blurring from Poisson blending, or fine-scale texture differences between the image and the inserted patch.

Original Image   Input   Output   Matching Scene

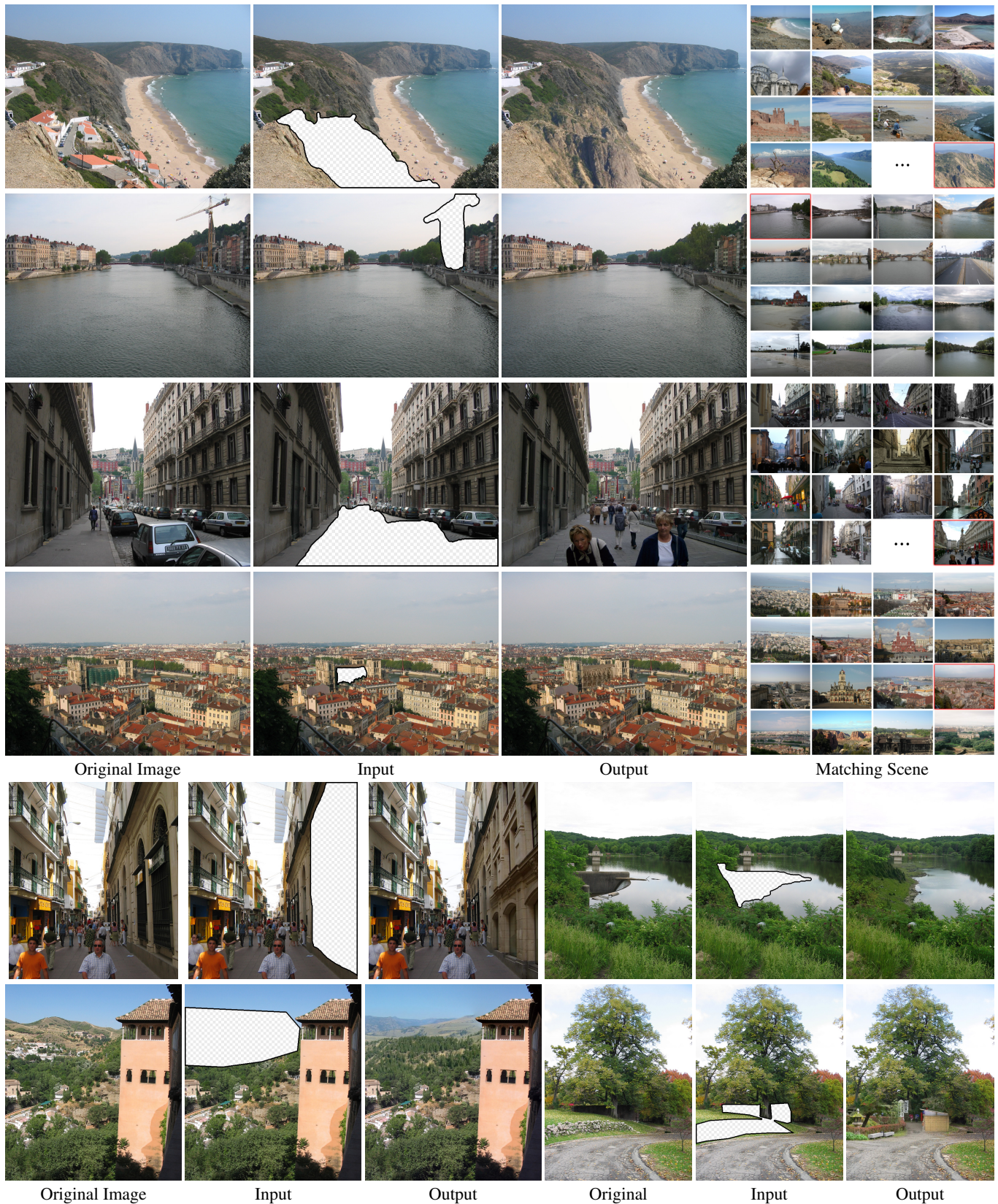Original Image   Input   Output   Original   Input   Output

Figure 6: Example results. The input and the matching scenes are composited together to create the outputs. The matching scene used to produce each output is highlighted in red. Note that the algorithm can handle a large range of scenes and missing regions. On rare occasions, the algorithm is lucky enough to find another image from the same physical location (4th result from the top).
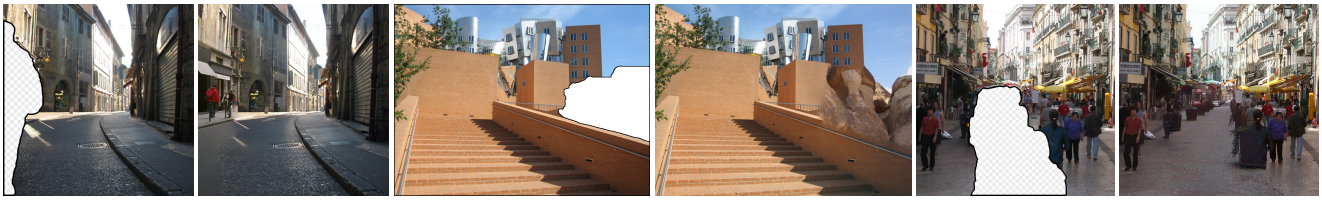
Figure 7: Typical failure cases. Some results exhibit pronounced texture seams (left). Others are failures of scene matching (center). The last failure mode (right), shared with traditional image completion algorithms, is a failure to adhere to high-level semantics (e.g. entire people).

Input   Criminisi et al.   Our Algorithm

Figure 8: Cases where existing image completion algorithms perform better than our algorithm.

For failure cases, these low-level artifacts are often much more pronounced (Figure 7, left). Another cause of failure is a lack of good scene matches which happens more often for atypical scenes (Figure 7, center). Semantic violations (e.g. half-objects) account for another set of failures (Figure 7, right). The latter is not surprising since the algorithm has no object recognition capabilities and thus no notion of object boundaries.

For uniformly textured backgrounds (Figure 8, top), existing image completion algorithms perform well. However, our algorithm struggles since our scene matching is unlikely to find the exact same texture in another photograph. Furthermore, image completion algorithms such as Criminisi et al. [2003] have explicit structure propagation which helps in some scenes (Figure 8, bottom).

Our algorithm requires over an hour to process one input image on a single CPU (50 min for scene matching, 20 min for local context matching, and 4 min for compositing). However, by parallelizing the processing across 15 CPUs, we are able to bring the running time down to about 5 minutes in our Matlab implementation.

5.1 Quantitative Evaluation

It is difficult to rigorously define success or failure for an image completion algorithm because so much of it depends on human perception. While previous approaches demonstrate performance qualitatively by displaying a few results, we believe that it is very important to also provide a quantitative measure of the algorithm's success. Therefore, to evaluate our method, we performed an IRB-approved perceptual study to see how well naive viewers could distinguish our results, as well as those of a previous approach [Criminisi et al. 2003], from real photographs.

The study was performed on a set of 51 test images that were defined *a priori* and spanning different types of completions. All test images came from either the LabelMe database [Russell et al. 2005] or our own personal photo collections, none of which are contained in the downloaded database. We were careful not to include any potentially recognizable scenes or introduce bias that would favor a particular algorithm. The regions we removed all had semantic meaning such as unwanted objects, storefronts, walls with graffiti, roads, etc. The test set is freely available on our website.

We generated three versions of each image – the original (unaltered) photograph, the result from Criminisi et al., and the result from our algorithm. Each of our 20 participants viewed a sequence of images and classified them as either real or manipulated. Of the 51 images each participant examined, 17 were randomly chosen from each source, but such that multiple versions of the same image did not appear. The order of presentation was also randomized. The participants were told that some of the images would be real, but not the ratio of real vs. manipulated images. We also told the participants that we were timing their responses for each image but that they should try to be accurate rather than fast. Overall the participants classified 80% of the images correctly. No effort was made to normalize for the differences in individual aptitude (which were small).

With unlimited viewing time, the participants classified our algorithm's outputs as real 37% of the time compared with 10% for Criminisi et al. Note that real images were identified as such only 87% of the time. Apparently, participants scrutinized the images so carefully that they frequently convinced themselves that real images were fake. It is interesting to examine the responses of participants as a function of maximum response time. In Figure 9 we show the proportion of images from each algorithm that have been marked as fake after *t* seconds of viewing. We claim that if a participant who has been specifically tasked with finding fake images cannot be sure that an image is fake within ten seconds, it is unlikely that an unsuspecting, casual observer would notice anything wrong with the image. After ten seconds of examination, participants have marked our algorithm's results as fake only 34% of the time (the other 66% are either undecided or have marked the image as real already). For Criminisi et al. participants have marked 69% of the images as fake by ten seconds. For real photographs, 3% have been marked as fake. All pairwise differences are statistically significant ($p < 0.001$).

6 Discussion

This paper approaches image completion from an entirely new direction – orthogonal and complementary to the existing work. While previous algorithms [Efros and Leung 1999; Criminisi et al. 2003; Drori et al. 2003; Wilczkowiak et al. 2005] suggest clever ways to reuse visual data within the source image, we demonstrate the benefits of utilizing semantically valid data from a large, unordered collection of unlabelled images. Our approach successfully fills in missing regions where prior methods, or even expert users with the Clone brush, would have no chance of succeeding because there is simply no appropriate image data in the source image to fill the hole. Likewise, expert users would have trouble leveraging such a large image collection – it would take a month just to view it with one second spent on each image. Additionally, this is the first paper in the field of image completion to undertake a full perceptual user study and report success rates on a large test set. While the results suggest substantial improvement over previous work, image completion is extremely difficult and is far from solved. Given the complimentary strengths of our method and single-image techniques, a hybrid approach is likely to be rewarding.

It takes a large amount of data for our method to succeed. We saw dramatic improvement when moving from ten thousand to two million images. But two million is still a tiny fraction of the high quality photographs available on sites like Picasa or Flickr (which has approximately 500 million photos). The number of photos on
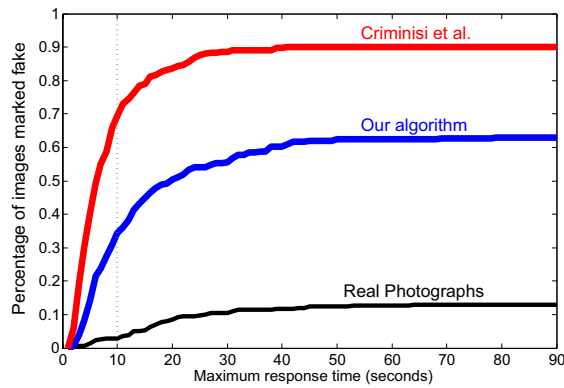
Figure 9: Cumulative percentage of images marked as fake by human subjects as function of response time.

the entire Internet is surely orders of magnitude larger still. Therefore, our approach would be an attractive web-based application. A user would submit an incomplete photo and a remote service would search a massive database, in parallel, and return results.

Beyond the particular graphics application, the deeper question for all appearance-based data-driven methods is this: *would it be possible to ever have enough data to represent the entire visual world?* Clearly, attempting to gather all possible images of the world is a futile task, but what about collecting the set of all *semantically differentiable scenes*? That is, given any input image can we find a scene that is "similar enough" under some metric? The truly exciting (and surprising!) result of our work is that not only does it seem possible, but the number of required images might not be astronomically large. This paper, along with work by Torralba et al. [2007], suggest the feasibility of sampling from the entire space of scenes as a way of exhaustively modelling our visual world. This, in turn, might allow us to "brute force" many currently unsolvable vision and graphics problems!

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Trans. Graph. 23*, 3, 294–302.

AGRAWAL, A., RASKAR, R., AND CHELLAPPA, R. 2006. What is the range of surface reconstructions from a gradient field? In *ECCV*.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11, 1222–1239.

CRIMINISI, A., PEREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. *CVPR 02*, 721.

DIAKOPOULOS, N., ESSA, I., AND JAIN, R. 2004. Content based image synthesis. In *Conference on Image and Video Retrieval*.

DRORI, I., COHEN-OR, D., AND YESHURUN, H. 2003. Fragment-based image completion. *ACM Trans. Graph. 22*, 3, 303–312.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001* (August), 341–346.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, 1033–1038.

IRANI, M., ANANDAN, P., AND HSU, S. 1995. Mosaic based representations of video sequences and their applications.

JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2006. Drag-and-drop pasting. *ACM Trans. Graph.*.

JOHNSON, M., BROSTOW, G. J., SHOTTON, J., ARANDJELOVIĆ, O., KWATRA, V., AND CIPOLLA, R. 2006. Semantic photo synthesis. *Computer Graphics Forum (Proc. Eurographics) 25*, 3 (September), 407–413.

KING, D. 1997. *The Commissar Vanishes*. Henry Holt and Co.

KOMODAKIS, N. 2006. Image completion using global optimization. In *CVPR*, 442–452.

KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (July), 277–286.

KWATRA, V., ESSA, I., BOBICK, A., AND KWATRA, N. 2005. Texture optimization for example-based synthesis. In *ACM Trans. Graph.*, 795–802.

OLIVA, A., AND TORRALBA, A. 2006. Building the gist of a scene: The role of global image features in recognition. In *Visual Perception, Progress in Brain Research*, vol. 155.

PEREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22*, 3, 313–318.

RUSSELL, B. C., TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2005. LabelMe: a database and web-based tool for image annotation. Tech. rep., MIT, 2005.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graph. 25*, 3, 835–846.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. *ACM Trans. Graph. 24*, 3, 861–868.

TORRALBA, A., MURPHY, K. P., FREEMAN, W. T., AND RUBIN, M. A. 2003. Context-based vision system for place and object recognition. In *ICCV*.

TORRALBA, A., FERGUS, R., AND FREEMAN, W. T. 2007. Tiny images. Tech. Rep. MIT-CSAIL-TR-2007-024.

WERTHEIMER, M. 1938. Laws of organization in perceptual forms (partial translation). In *A sourcebook of Gestalt Psychology*, W. Ellis, Ed. Harcourt Brace and Company, 71–88.

WEXLER, Y., SHECHTMAN, E., AND IRANI, M. 2004. Space-time video completion. *CVPR 01*, 120–127.

WILCZKOWIAK, M., BROSTOW, G. J., TORDOFF, B., AND CIPOLLA, R. 2005. Hole filling through photomontage. In *BMVC*, 492–501.