# Scene Completion Using Millions of Photographs

By James Hays and Alexei A. Efros

### 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 labeling by the user. Unlike existing image completion methods, our algorithm can generate a diverse set of image completions and we allow users to select among them. We demonstrate the superiority of our algorithm over existing image completion approaches.

### **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.<sup>13</sup> Other times, there is simply missing data in some areas of the image: (a) an aged corner of an old photograph (b) a hole in an image-based 3D reconstruction due to occlusion, and (c) 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 cannot 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 (Figure 1), such as video (using various background stabilization techniques) or multiple photographs of the same scene.1,19

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<sup>4,6,24,25</sup> operate by extending adjacent textures and contours into the unknown region. These algorithms are similar to texture synthesis algorithms such as,8,7,14,15 sometimes with additional constraints to explicitly preserve Gestalt cues such as good con*tinuation*,<sup>23</sup> either automatically<sup>4</sup> or by hand.<sup>20</sup> 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<sup>6,25</sup> 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 does not 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 image content that appeared somewhere else in the same image. For situations in which the incomplete region is not bounded by texture regions, or when



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



Original image

Criminisi et al. 2003

Smart erase

Our algorithm

there is too little useful texture, existing algorithms have trouble completing scenes (Figure 2).

### 2. OVERVIEW

In this paper, we perform image completions 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<sup>6,8</sup> because of the slow texture search. Texture synthesis-based image completion methods are difficult to accelerate with traditional nearest-neighbor or approximate nearest-neighbor methods because of the high dimensionality of the features being searched and because the known dimensions of the feature being matched on 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 do not 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. More than other image completion methods, we need a robust seam-finding and blending method to make our image completions 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 one million images to a manageable number of best matching scenes (60 in our case), which are used for image completion. We use a low-dimensional scene descriptor<sup>16</sup> so it is relatively fast to find the nearest scenes, even in a large database. Our approach is purely data driven, requiring no labeling or supervision.

In order to seamlessly combine image regions we employ Poisson's blending. 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<sup>25</sup> or other heuristics to encourage a constant intensity offset for the blending boundary.<sup>11</sup> 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.25 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.<sup>5,12</sup> These methods create semantically valid images through explicit semantic constraints using image databases with semantically labeled regions. The database labeling process must be supervised<sup>5</sup> or semisupervised.<sup>12</sup>

### **3. SEMANTIC SCENE MATCHING**

Since we do not employ user-provided semantic constraints or a labeled 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 30 Flickr.com groups that focus on landscape, travel, or city photography. Typical group names are "lonely planet," "urban fragments," and "rural decay." Photographs in these groups are generally high quality. We also downloaded images based on keyword searches such as "travel," "landscape," and "river." We discarded all duplicate images and all images that did not have at least 800 pixels in their largest dimension and 500 pixels in their smallest dimension. All images were down-sampled, if necessary, such that their maximum dimension was 1024 pixels. Our database downloading, preprocessing, and scene matching are all

distributed among a cluster of 15 machines. In total we acquired about 2.3 million unique images totalling 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.<sup>5,12</sup>

We use the gist scene descriptor<sup>16,22</sup> which has been shown to perform well at grouping semantically similar scenes (e.g., city, tall buildings, office, fields, forest, and 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 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 one million yielded a qualitative leap in performance (see Figure 3 for a typical scene matching result). Independently, Torralba et al.<sup>21</sup> have observed a similar effect with a dataset of up to 70 million tiny  $(32\times32)$  images.

The gist descriptor aggregates oriented edge responses at multiple scales into very coarse spatial bins. We found a gist descriptor built from six oriented edge responses at five scales aggregated to a  $4 \times 4$  spatial resolution to be the most effective. We augment the scene descriptor with the color information of the query image down-sampled 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 one 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 compute the  $L_1$  distance between the gist of the query image and every gist in the database, weighted by the mask. The color distance is computed in the  $L^*a^*b^*$  color space. These distances are combined and weighted such that the gist contributes roughly twice as much as the color information to the final distance.

Because we match scenes using arbitrary dimensions of the descriptor (depending on which regions of a query image are missing), we cannot use principal components analysis (PCA) dimensionality reduction as suggested in<sup>16</sup> For the same reason, we do not use any approximate nearestneighbor 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. The local context is 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 three scales (0.81, 0.90, 1) using pixel-wise SSD error in  $L^*a^*b^*$  color space. Only placements (translations and scales) for which the context is fully

Figure 3: The first 164 nearest neighbor scenes for the incomplete image in the center. Most of the scenes are semantically and structurally similar; many are even from the same city (London).



contained in the matching scene are considered. Since we expect our matching scenes to already be roughly aligned with the incomplete image, we discourage spurious, distant matches by multiplying the error at each placement by the magnitude of the offset.

We composite each matching scene into the incomplete image at its best scoring placement using a form of graph cut seam finding<sup>15</sup> and standard Poisson's blending.<sup>17</sup> 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<sup>4</sup> have treated the remaining valid pixels in an image as hard constraints which are not changed. We relax this, as in<sup>25</sup> 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.

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.11 uses iterative dynamic programming optimizations to find a seam which leaves 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. Inspired by this, as well as the fact that our scene matches tend to have colors similar to our query image, we propose a seam-finding heuristic which ignores intensity differences entirely and instead minimizes the difference in image gradients. This heuristic finds seams that avoid cutting high-frequency image content for which Poisson's blending would cause obvious artifacts.

An advantage of our heuristic is that we can quickly find the optimal seam using a graph cut procedure.<sup>3</sup> The details of our graph cut setup can be found in our original SIGGRAPH publication.<sup>9</sup> A very similar metric was mentioned but not demonstrated in<sup>1</sup> As in,<sup>1</sup> we allow the Poisson blending to operate on the entire image domain instead of correcting one of the images. We use the Poisson solver of.<sup>2</sup>

Finally, we assign each composite a score which is the sum of the scene matching distance, the local context matching distance, the local texture energy 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**

We test our algorithm on a set of 51 images with missing regions. All test images come from either the LabelMe database<sup>18</sup> or our own personal photo collections, none of which are contained in our downloaded database. Images in the test set, about a half megapixel each, are higher resolution than the images used in most previous works<sup>4,6</sup> and likewise the missing regions are quite large (56,000 pixels on average). The regions we removed from the photos all had semantic meaning such as unwanted objects, store-fronts, walls with graffiti, roads, etc. The test set is made freely available on the authors' Web page.

Image completion is an inherently underconstrained 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 viable, 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 (Figure 4). After compositing, the bestmatching 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 completion(s) which they find most compelling. In other cases, only a few or none of the results were acceptable. The quality of our results benefits from this very simple user interaction and it





Original

Input

Alternative completions

is difficult for conventional image completion methods to offer an analogous selection of results.

Some of our image completions are shown in Figure 5. The bottom result 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 an increasingly large database such fortuitous events do occur.

In all of the successful cases, the completion is semantically valid but there might be slight low-level artifacts such as resolution mismatch between the image and patch, blurring from Poisson's blending, or fine-scale texture differences between the image and patch. For failure cases these lowlevel artifacts are often much more pronounced (Figure 6, top). Another source of failure is a lack of good scene matches which happens more often for atypical scenes (Figure 6, middle). Semantic violations (e.g., half-objects) account for another set of failures. 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 7, 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.<sup>4</sup> have explicit structure propagation which helps in some scenes (Figure 7, bottom).

Our hole filling algorithm requires about 5 min to process an input image. The scene matching, local context matching, and compositing would take about 50, 20, and 4 min respectively on a single central processing unit (CPU) but we parallelize all of these across 15 CPUs. Our algorithm is implemented in MATLAB and all of the timings are for Pentium 4 processors.

### 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 a perceptual study to see how well naive viewers could distinguish our results, as well as those of a previous approach,<sup>4</sup> 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. We were careful not to include any potentially recognizable scenes or introduce bias that would favor a particular algorithm. We generated three versions of each image-the real photograph from which the image completion test cases were constructed, 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 real or manipulated. Of the 51 images each participant examined, 17 were randomly chosen from each source, but such that they do not see multiple versions of the same image. The order of presentation was also randomized. The participants were told that some of the images would be real, but they were not told the ratio of real versus 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 the participants classified our algorithm's outputs as real 37% of the time compared with 10% for Criminisi et al.<sup>4</sup> Note that participants identified real images as such only 87% of the time. Participants scrutinized the images so carefully that they frequently convinced themselves real images were fake.

It is interesting to examine the responses of participants over time. In Figure 8 we measure the proportion of images from each algorithm that have been marked as fake with an increasing limit on the amount of time allowed. We claim that if a participant who has been specifically tasked with finding fake images cannot be sure that an image is fake within 10 s, it is unlikely that an unsuspecting, casual observer would notice anything wrong with the image. After 10 s 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 10 s. For real photographs, only 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<sup>4,6,8,25</sup> suggest clever ways to reuse visual data within the source image, we demonstrate the benefits of utilizing semantically valid data from a large collection of unlabeled 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 10 days 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 complementary 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 one million images. But one million images is still a tiny fraction of the high-quality photographs available on sites like Picasa or Flickr (which has approximately 2 billion images). The number of photos on the entire Internet is surely orders of magnitude larger still. Therefore, our approach would be an attractive Web-based application. A user would Figure 5: Results. The input and the matching scenes are composited together to create the outputs. The matching scene used in 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 as seen in the bottom example.



Figure 6: Typical failure cases. Some results exhibit pronounced texture seams (top). Others are failures of scene matching (middle). The last failure mode (bottom), shared with traditional image completion algorithms, is a failure to adhere to high-level semantics (e.g., entire people).



Input

Scene match

Figure 7: Situations where existing image completion algorithms perform better than our algorithm.

Output



submit an incomplete photo and a remote service would search a massive database, in parallel, and return results.

### 7. TOWARD BRUTE-FORCE IMAGE UNDERSTANDING

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 our dynamic 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 Figure 8: The percentage of images marked as fake within any amount of viewing time by participants in our perceptual study.



Torralba et al.,<sup>21</sup> suggest the feasibility of sampling from the entire space of scenes as a way of exhaustively modeling our visual world. This, in turn, might allow us to "brute force" many currently unsolvable vision and graphics problems!

Further supporting this possibility, we recently used scene matching methods similar to those presented here to estimate the GPS location of an arbitrary image. In a project called IM2GPS,<sup>10</sup> we collect a database of 6 million geotagged photographs from Flickr and show that image matches for a query photo are often "similar enough" to be geographically informative even if we do not match to the exact, real-world location. We represent the estimated image location as a probability distribution over the Earth's surface (see Figure 9). We quantitatively evaluate our approach in several geolocation tasks and demonstrate encouraging performance (up to 30 times better than chance). We show that geolocation estimates can provide the basis for numerous other image understanding tasks such as population density estimation, land cover estimation or urban/rural classification (see<sup>10</sup> for details).

Figure 9: Geolocation estimates for photos of the Grand Canyon and a generic European alley. From left to right are the query photographs, the first 16 nearest scene matches, and the distribution of the top 120 nearest-neighbors across the Earth. Geographic clusters are marked by X's with size proportional to cluster cardinality. The ground truth locations of the queries are surrounded by concentric green circles.



### Acknowledgments

We would like to thank Antonio Torralba for the gist code and useful comments and Gabriel Brostow for sharing their hole-filling software. We also thank Tamara Berg, Alex Berg, and Rob Fergus for sharing their image downloading scripts. Finally we are grateful to Flickr for allowing us to download so many images. This work has been partially supported by NSF Graduate Research Fellowship to James Hays and by NSF grants CCF-0541230 and IIS-0546547.

### References

- Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. Interactive digital photomontage. ACM Trans. Graph., 23(3):294–302, 2004.
- Agrawal, A., Raskar, R., and Chellappa, R. What is the range of surface reconstructions from a gradient field? *European Conference* on *Computer Vision (ECCV)*, May 2006.
- Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Tntell.*, 23(11):1222–1239, 2001.
- Mach. Intell., 23(11):1222–1239, 2001.
  Criminisi, A., Perez, P., and Toyama, K. Object removal by exemplar-based inpainting. *CVPR*, 02:721, 2003.
- 5. Diakopoulos, N., Essa, I., and Jain, R. Content based image synthesis. *Conference on Image and Video*

- Retrieval (CIVR), 2004.
- Drori, I., Cohen-Or, D., and Yeshurun, H. Fragment-based image completion. ACM Trans. Graph., 22(3):303–312, 2003.
- Efros, A. A. and Freeman, W. T. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.
- Efros, A. A. and Leung, T. K. Texture synthesis by non-parametric sampling. *ICCV*, pages 1033–1038, Corfu, Greece, September 1999.
- Hays, J. and Efros, A. A. Scene completion using millions of photographs. ACM Trans. Graph. (SIGGRAPH 2007), 26(3): 4, 2007.
- Hays, J. and Efros, A. A. im2gps: estimating geographic information from a single image. CVPR, 2008.

- Jia, J., Sun, J., Tang, C.-K., and Shum, H.-Y. Drag-and-drop pasting. *ACM Trans. Graph.*, 25(3): 631–637, 2006.
- Johnson, M., Brostow, G. J., Shotton, J., Arandjelović, O., Kwatra, V., and Cipolla, R. Semantic photo synthesis. *Comput. Graph. Forum (Proc. Eurographics)*, 25(3):407–413, 2006.
- King, D. The Commissar Vanishes. Henry Holt and Company, London, 1997.
- Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. Texture optimization for example-based synthesis. ACM Trans Graph., 24: 795–802, 2005.
- Kwatra, V., Schodl, A., Essa, I., Turk, G., and Bobick, A. Graphcut textures: Image and video synthesis using graph cuts. ACM Trans. Graph., or Computer Science Science
- 22(3):277–286, 2003.
  Oliva, A. and Torralba, A. Building the gist of a scene: The role of global image features in recognition. Visual perception. *Progr. Brain Res.*, 155:23–36, 2006.
- Perez, P., Gangnet, M., and Blake, A. Poisson image editing. ACM Trans. Graph., 22(3):313–318, 2003.
- Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. LabelMe: A database and web-based tool for

image annotation. Technical Report, MIT, 2005, 2005.

- Snavely, N., Seitz, S. M., and Szeliski, R. Photo tourism: exploring photo collections in 3d. ACM Trans. Graph., 25(3):835–846, 2006.
- Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3):861–868, 2005.
- Torralba, A., Fergus, R., and Freeman, W. T. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, 2007.
- Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. Context-based vision system for place and object recognition. *ICCV*, 2003.
- Wertheimer, M. Laws of organization in perceptual forms (partial translation). In: W. Ellis, editor, A Sourcebook of Gestalt Psychology, pages 71–88. Harcourt Brace and Company, London, 1938.
- Wexler, Y., Shechtman, E., and Irani, M. Space-time video completion. *CVPR*, 01:120–127, 2004.
- Wilczkowiak, M., Brostow, G. J., Tordoff, B., and Cipolla, R. Hole filling through photomontage. *BMVC*, 492–501, July 2005.

James Hays and Alexei A. Efros (jhhays@cs.cmu.edu, efros@cs.cmu.edu) Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA

A previous version of this paper was published in *ACM Transactions on Graphics*, Vol. 26, Issue 3 (July 2007).

© 2008 ACM 0001-0782/08/1000 \$5.00

## ACM Transactions on Internet Technology



This quarterly publication encompasses many disciplines in computing—including computer software engineering, middleware, database management, security, knowledge discovery and data mining, networking and distributed systems, communications, and performance and scalability—all under one roof. TOIT brings a sharper focus on the results and roles of the individual disciplines and the relationship among them. Extensive multi-disciplinary coverage is placed on the new application technologies, social issues, and public policies shaping Internet development.

