Human Computation
AAAI Technical Report WS-12-08

Hallucination: A Mixed-Initiative Approach for Efficient Document Reconstruction

Haoqi Zhang, John K. Lai, Moritz Bächer
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA
hq@eecs.harvard.edu, jklai@seas.harvard.edu, moritz_baecher@harvard.edu

Abstract

We introduce a mixed-initiative approach for document reconstruction that can significantly reduce the amount of time and effort required to reassemble a document from shredded pieces or an artifact from broken fragments. We focus in particular on the hardest subproblem, which is the problem of identifying a matching neighbor for any given piece. Our approach, called *hallucination*, combines human and machine intelligence by leveraging people's ability to draw what a neighboring piece may look like, and then using the drawing as a template based on which the computer computes likely matches. Experiments on a puzzle from the DARPA Shredder Challenge demonstrate that the hallucination approach significantly reduces the search space for identifying a match, outperforming humans and computers working in isolation.

Introduction

Over the last decade, there has been a rise in systems that leverage human and machine intelligence for solving complex problems that neither can solve alone. Such systems take advantage of human abilities—particularly in vision, natural language, and pattern recognition—to handle instances and aspects of problems that are difficult for computers. The ESP game (von Ahn and Dabbish 2008), FoldIt (Cooper et al. 2010), and reCAPTCHA (von Ahn et al. 2008) are a few examples of successful systems that draw on human contributors and machine computations to tackle problems in image labeling, protein folding, and text digitization.

An interesting problem that can benefit from harnessing a mix of human and machine intelligence is document reconstruction. Given a document or artifact that has been broken up into many smaller pieces, the goal is to reconstruct the document or artifact to its original by arranging the pieces into their correct relative positions and orientations. The problem has natural applications in archaeology and art as well as in forensics and investigation sciences.

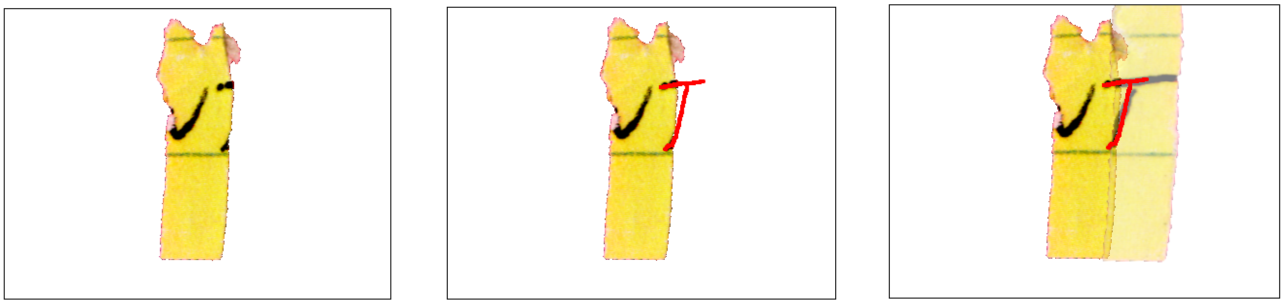
An important subproblem in document reconstruction is finding pairwise matches; that is, figuring out for a given piece what its neighboring pieces are. This subproblem is difficult for humans alone and for computers alone. For humans, there are two major challenges. One is scale: for each piece, finding a matching neighbor requires a linear scan across all unmatched pieces, which is expensive when there is a large number of pieces. Another is confirming a match: a match requires having the pieces in the correct orientation and relative positions, and figuring out how two pieces may or may not connect requires effort and is non-trivial.

The tedium of manual reassembly of fragmented pieces led to numerous efforts in automated reconstruction of documents and artifacts. Early work on apictorial jigsaw puzzles (Freeman and Garder 1964) focuses on using piece contours to perform pairwise matches. Since then other local matching algorithms have been suggested that use content-based features such as colors and textures (Ukovich et al. 2004; Toler-Franklin et al. 2010). While these approaches can help reduce the search space, their effectiveness depends largely on the quality of features. For example, in settings where contours are non-discriminative, automatic reconstruction is difficult if not impossible.

One advantage that humans have over machines is that humans are much more efficient at abstracting and matching visual cues across piece borders based on their content. For example, a person looking at a piece of a shredded document can recognize a letter that is only partially present, and an experienced archaeologist looking at a particular piece of a broken artifact can recognize unique patterns that extend beyond the fragment. Unfortunately, for a human to find a matching piece still requires scanning through the pieces, which is expensive when there are many pieces. Given that such knowledge is often domain-specific and hard to encode, it is also difficult to build automated systems that can fully take advantage of it.[1]

In this paper, we introduce a novel mixed-initiative approach for document reconstruction that we call *hallucination*. Hallucination integrates human and machine intelligence by having humans draw based on a single piece what they think a neighboring piece may look like, and then having the machine perform template matching using these drawings to find pairwise matches (see Figure 1). Hallucination takes advantage of people's ability to look at the con-

[1] An automated system would require as features information about content across piece borders; such a system thus needs to consider not only recognizing 'full content' (e.g., full handwritten characters) but also 'partial content' (e.g., handwritten characters that are cutoff at arbitrary positions).



(a) Shredded Piece (b) User-Drawn 'Hallucination' (in red) (c) Matched Neighbor

Figure 1: (a) A piece of a shredded document with cut off letters 'N' and 'T' written in black ink is displayed. We use human-drawn templates (in red, (b)) to find the best matching neighbor (c) using a standard computer vision algorithm. This approach, which we call *hallucination*, significantly outperforms humans and computers working in isolation.

tent on a piece and know something about the content on a neighboring piece, and to express this knowledge naturally by drawing. These drawings can then be treated as templates that the machine can use to tractably compute and identify the best matching pieces. Hallucination thus aims to address the challenges with both manual and automated reconstruction approaches: human knowledge about potential matches is passed on to the computer in a natural manner, and the computer uses this knowledge to expedite the search for the correct match.

To test the hallucination method, we implement a prototype system for identifying pairwise matches for reconstructing a shredded document from the DARPA Shredder Challenge. Experiments demonstrate that the hallucination method significantly outperforms humans or machines working on this problem in isolation, requiring up to 27% fewer pieces to be scanned for human verification over a machine algorithm and up to 54% fewer pieces than a person scanning the pieces at random. We suggest a number of areas for future work, and highlight how the hallucination method can be utilized in a variety of settings and integrated as part of a complete document reconstruction system.

Related Work

Recognizing the challenges in document reconstruction, DARPA issued the Shredder Challenge in Fall 2011.[2] The challenge contained a number of shredded documents with handwritten text and images that needed to be reconstructed to answer questions based on their content. There were a total of five puzzles of increasing difficulty, ranging from hundreds to thousands of pieces. Given the regularity of the shredded pieces and the imprecision of shape contours, it is difficult for a machine-based approach to take advantage of piece contours to identify matches. Teams competing in the challenge used a variety of approaches, with some using machine vision algorithms to identify likely matches, that were then passed on to human contributors for verification and manual matching. The hallucination method seeks to better integrate human and machine intelligence by involving humans in the process of identifying likely matches by using their drawings to reduce the search space.[3]

Our template-based hallucination approach complements feature-based approaches for identifying pairwise matches, and applies even when features such as contours are non-discriminative. Like other pairwise matching algorithms, the hallucination method can be integrated as part of a global assembly strategy (Wolfson et al. 1988; Castañeda et al. 2011) that utilizes results from pairwise matches to completely solve the puzzle while taking into account global constraints and being able to recover from errors.

Template matching is a standard computer vision technique for finding parts of an image that match or are similar to a template image (see, e.g., (Brunelli 2009) for an overview). Many vision and graphics applications such as object tracking (Mao et al. 2011), image retargeting (content-aware changes to the aspect ratio) (Simakov et al. 2008), and image completion (filling of unwanted regions with plausible content) (Criminisi, Perez, and Toyama 2003) make use of template matching. In such applications, templates are typically patches in the underlying image. For document reconstruction, templates are not readily available, and are instead created by contributors as part of the hallucination process.

For the hallucination method to be feasible, it is important to be able to check a template against a large set of pieces efficiently. In situations where matches occur at a sparse set of pixel locations (e.g., we can restrict checking of handwriting templates to pixels that are part of the ink regions of pieces) and pieces can be aligned up to a few orientations (e.g., sheets of paper may include lines that can be used to align snippets up to flips), matches can be computed at interactive rates by calculating pairwise cross-correlations in a straightforward manner. In more complex settings where templates either have to be scaled and rotated to best match with a neighbor and matches occur at a dense set of interest

[2] See archive.darpa.mil/shredderchallenge/

[3] Only one team successfully solved all the puzzles, doing so after spending 600 man-hours. As the winning team's approach still required a significant amount of human effort and used custom-tailored algorithms, it may not scale well to more difficult instances or apply to other document or artifact reconstruction tasks.

Figure 2: Puzzle 1 from the DARPA Shredder Challenge (left) and its solution (right). The puzzle contains 227 pieces.

points, techniques introduced in recent works such as PatchMatch (Barnes et al. 2009) and its generalization (Barnes et al. 2010) and extensions (Korman and Avidan 2011) can be used to reduce computation time and space complexity.

Our work is also related to the larger body of previous efforts in leveraging human computation for vision tasks. Many works focus on how to efficiently and accurately collect labeled annotations that can be used to improve the performance of machine vision algorithms (Vijayanarasimhan and Grauman 2011a; 2011b; Welinder et al. 2010). Other works take advantage of human abilities in new ways, e.g., to build a *crowd kernel* for judging object similarity (Tamuz et al. 2011), or to gather human understandable attributes that best help to discriminate between objects (Parikh and Grauman 2011). Directions have also led to interactive vision systems that involve humans in the loop, e.g., for object classification (Branson et al. 2010). To the best of our knowledge, we are the first to present a human-directed, template-based technique for document reconstruction.

Problem Definition

Consider a document or artifact D that has been broken into many pieces $K = \{p_1, \ldots, p_k\}$, some of which may be missing. The *document reconstruction problem* asks for an assignment of coordinates and orientations to available pieces, such that the relative positions and orientations of the pieces are correct with respect to the actual document D. For any given piece p, the *pairwise matching subproblem* asks for the neighbors of p, as well as their relative positions and orientations with respect to p. Figure 2 shows an example of a document reconstruction problem from the DARPA Shredder Challenge, which we will use as a running example and later in our experiments.

We assume the existence of visual cues on (some) pieces that provide a human with some information about the content that is likely to be on a neighboring piece, and that this information can be expressed via drawing. For example, a person looking at pieces of a shredded document may be able to recognize and complete letters that are only partially present. We make no assumptions about the quality of available features such as piece contours, which may be non-discriminative for reconstructing shredded documents.

The Hallucination Method

The hallucination method aims to discover pairwise matches by first having a human draw based on a given piece the content that is likely to be on a neighboring piece, and then using the drawing as a template that is matched against other pieces to identify the best matches (neighboring pieces and their relative positions). We can think of a piece as providing a noisy signal of a local region of the puzzle that gives some clues as to what piece might be next to it. Figure 1 shows this process for finding a neighboring piece based on a hallucination of the letter ‘T’ that is partially shown on the right side of the displayed piece.[4]

To determine how good a piece is as a matching neighbor, we fix the position of the template, and place the piece in positions at which the template and the piece overlap. For each of these positions, we compute a score based on the similarity of the template and the piece on the region of overlap. We take the highest score across all positions for each piece as that piece’s score. Potential matches include in them the relative positions at which the two pieces match, thus requiring no further manual positioning whenever the highest scoring position is the correct one.[5]

While the hallucination method is here described for performing pairwise matches, we can also employ this technique when a user is shown more context (e.g., all the pieces connected thus far), or for finding matches globally (e.g., where a user draws not only what the neighboring piece may look like but what an entire region may look like). At a high level, the core idea is to allow humans to express, via drawings, the likely completions of the document when provided with some portion of the document. These drawings then serve as soft ‘visual constraints’ that the machine can use to narrow down the solution space.

Application to the DARPA Shredder Challenge

We apply the hallucination method to the documents in the DARPA Shredder Challenge, focusing here on Puzzle 1 (see Figure 2). We describe below implementation details specific to this domain.

Preprocessing

We first segment the pieces from the image by extracting them from the background. Since almost all of the shredded pieces include lines (e.g., note the green line on the piece in Figure 3), we can automatically align most of the pieces using a simple Hough transform (Ballard 1987). For pieces where this process fails, we manually align them by drawing over the lines and recording these coordinates. The orientations are therefore reduced to two (pieces can still be flipped). To simplify our experiments, we further reduce the

[4] The interested reader can find a short video demonstrating the hallucination method for the DARPA Shredder Challenge at http://tinyurl.com/hallucinatevideo

[5] As described, our method assigns a single score to each piece based on the best scoring position. Our underlying assumption is that the best scoring position for a matching neighbor will typically be (near) the actual position of the match. The method can be easily modified to allow multiple matches for each piece.

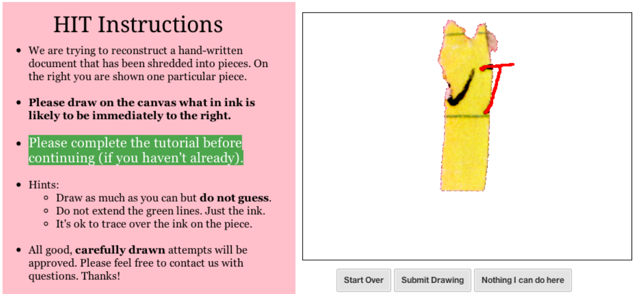


Figure 3: Screenshot of the hallucination interface presented to Mechanical Turk workers.

possible orientations to one (flipped or not flipped), by manually scanning the pieces and determining the correct orientation based on the text on each piece.

User Interface

Users are provided with a HTML5 canvas interface on which to draw their hallucinations (see Figure 3). HTML5 canvas allows for simple pixel level manipulation, and makes it easy to export drawings as image files that can be directly passed on to our template matching algorithm or stored in a database for later processing. In the current implementation a user only hallucinates on the most distinguishing feature, namely the black ink text.

Template Matching

We perform all of our template matching using binary images. For user drawings, the background is white and the hallucinations are treated as black. For potential neighbors, the ink in the image is black and all other pixels are treated as white.

To measure the similarity between a user-specified template and a corresponding patch of a potential neighbor, we compute as the score their *normalized cross-correlation* (Lewis 1995), which is a standard similarity metric. For any given position of a potential neighbor, we let $\mathbf{T}$ and $\mathbf{M}$ be 2-dimensional 0-1 arrays that respectively hold the template and potential neighbor's values in the region of overlap. By subtracting the mean of values in $\mathbf{T}$ from each element in $\mathbf{T}$ and similarly subtracting the mean of values in $\mathbf{M}$ from each element in $\mathbf{M}$, we derive the mean-normalized matrices $\bar{\mathbf{T}}$ and $\bar{\mathbf{M}}$. The score for the position is the cosine similarity between $\bar{\mathbf{T}}$ and $\bar{\mathbf{M}}$:

$$\frac{\bar{\mathbf{T}} : \bar{\mathbf{M}}}{|\bar{\mathbf{T}}||\bar{\mathbf{M}}|}$$

where : denotes a double dot product[6] and $|\mathbf{A}|$ is defined as $\sqrt{\mathbf{A} : \mathbf{A}}$. A score of 1 indicates that the values for the template and the potential neighbor in the region of overlap are identical (a perfect match), while a value of -1 indicates that they are complete opposites.

[6] A double dot product of two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same size is defined as the sum of component-wise multiplied elements, $\sum_{ij} A_{ij} B_{ij}$.

In general settings, the normalized cross-correlation can be efficiently computed by performing calculations in Fourier space (Lewis 1995). In our setting, we can directly compute scores in the spatial domain because we are able to drastically reduce the number of positions we search through in the following ways. First, because a user's template is drawn relative to the original piece, we know where lines should be (relative to the template), and need only compute scores for positions where the lines on the original piece and its potential neighbor align.[7] Second, we only compute scores for the first few or last few columns based on where the user is drawing. For example, if the user's drawing is on the right hand side of the original piece, we need only consider the first few columns of any potential neighbor. Combining these two restrictions results in a small number of positions to score for each potential neighbor, which allows us to compute the normalized cross-correlation directly.

Experiments

To evaluate the effectiveness of the hallucination approach, we conducted experiments using pieces from the puzzle shown in Figure 2. There are a total of 227 pieces, which we preprocessed in the manner described in the previous section by first extracting them from the original image and then aligning their orientation. We discarded pieces with very little ink as these pieces cannot serve as pieces to match or as potential neighbors.[8] This left us with 109 pieces. To streamline our experiments, we focused on finding matches on the right side of a piece, and only considered matching pieces with right side neighbors in this set of 109 pieces. This left us with 98 pieces for matching.

Metric

The goal of any pairwise matching technique is to significantly reduce the amount of human time and effort required to find matches. We assume that the output from a pairwise matching method is an ordered list of pieces, and consider as a metric the *rank* (or position) at which the first matching neighbor occurs (lower is better). For pieces that have multiple matching neighbors, we consider the lowest rank across the possible neighbors. We take this rank as a proxy for the amount of manual effort required for verification, with the view that the fewer pieces that need to be verified before finding a match, the better.[9]

Conditions

We considered three conditions. The first condition, *hallucination*, considered the hallucination approach where we

[7] We search 2 pixels in each direction to account for any noise.

[8] We discarded pieces for which the text bounding box is smaller than 20 pixels by 20 pixels.

[9] The matching methods we considered do provide position information, but we did not use this information for comparison. Our assumption is that the highest scoring positions based on hallucinations will tend to be the correct ones, and this was indeed the case when we visually examined the best scoring positions of matching neighbors in the hallucination condition.

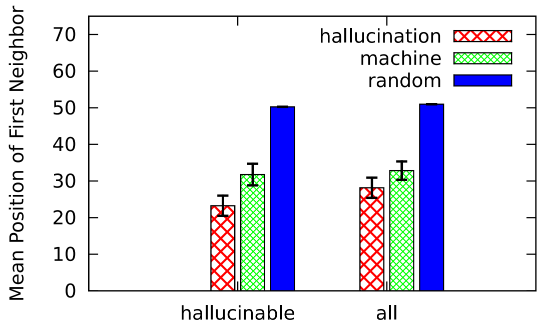


Figure 4: Average rank of the first matching neighbor under the three conditions (lower is better). *all* indicates the results when all 98 pieces are considered, and *hallucinable* indicates the results when only the 76 hallucinable pieces are considered. Bars (in black) show the standard error.

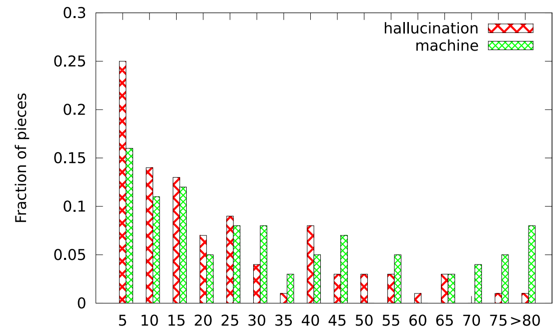


Figure 5: Breakdown of the ranks of the first matching neighbor for the hallucination and machine conditions when only considering hallucinable pieces. The labels on the horizontal axis gives the right end point of the rank bucket. For instance, the bars corresponding to 5 provides the fraction of pieces where the first matching neighbor was in rank 1–5.

used user drawings as templates to score potential neighbors. To obtain hallucination drawings, we posted tasks on Amazon Mechanical Turk and obtained five hallucinations for every piece considered (see interface in Figure 3). Workers were instructed to draw on a HTML5 canvas what ink is likely to be immediately to the right of the displayed piece. To help workers learn how to perform the task, we created a short tutorial containing illustrative examples and true/false questions about the task's instructions that workers must successfully complete prior to drawing. In addition to making sure that workers understood the instructions, the tutorial aimed to teach workers that it is not always possible to hallucinate entire letters (or at all) based on the displayed piece, and that they should hallucinate as much as possible but should not guess. In cases where no hallucination is possible (i.e., because there is too little ink or no ink crossing the border), workers were asked to click the 'Nothing I can do here' button. We recruited U.S. workers with a 98% or higher approval rating, and paid them 25 cents for completing the tutorial and five cents for each drawing.

For a given piece and a user drawn template, we assigned a score to each potential neighbor by computing the maximum normalized cross-correlation across all positions at which the potential neighbor and template overlap. Sorting the potential neighbors by this score resulted in an ordered list of pieces for each user drawn template. To remove noise and raise the ranking of pieces that were well-ranked in multiple hallucinations, we formed a final ordered list by aggregating the individual template-based ordered lists using a simple Borda count (Borda 1781), such that pieces with a smaller sum of ranks appear at lower ranks in the final list.[10]

The second condition, *machine*, considered a completely automated algorithm. We implemented a matching algorithm that forms a template by extending the ink that is along the border of the piece being matched.[11] The template was then used in the same manner as user created templates in the hallucination condition. This method aims to take advantage of the fact that ink text from cutoff characters tends to continue along borders.

As a baseline, the third condition, *random*, considered a randomly ordered list. This captures the case in which a human randomly scans through the pieces to look for a match. Since a matching neighbor is equally likely to be in any position, the expected minimum rank of a matching neighbor is $\frac{k+1}{\#\text{ of neighbors} +1}$, where k is the number of pieces considered.

Results

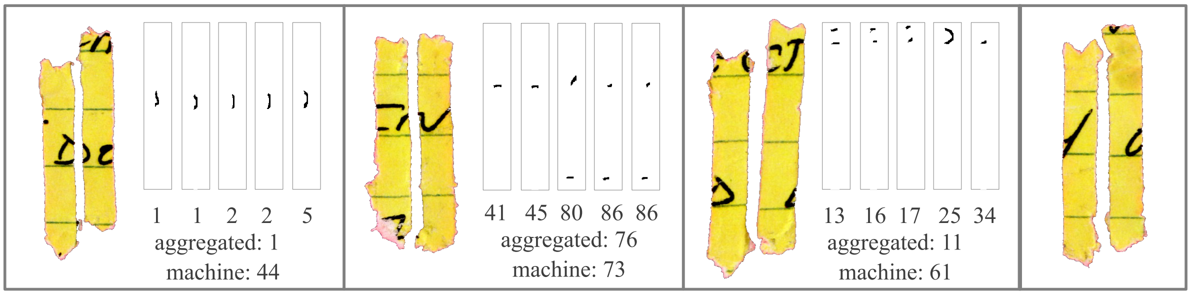
Figure 4 shows the average rank of the first matching neighbor for the three conditions. When considering all pieces, the Wilcoxon test shows a difference (z = -1.42, p = 0.078) between the rank of the first matching neighbor in the hallucination condition (μ = 28.2) and the machine condition (μ = 32.8), where hallucinations allow for a 14% saving in the number of pieces that need to be reviewed before a matching neighbor is found. When compared to the random baseline (μ = 51.0), hallucinations allow for a 44% saving.

In some cases, there is little to no context at the boundaries based on which to hallucinate, i.e., there is no cutoff character at the right border (see Figure 6(d)). When we considered only these *hallucinable* pieces where ink crosses the right border (leaving 76 of the 98 pieces), we found an even larger difference in the rank of the first matching neighbor (see Figure 4), where the Wilcoxon test shows that the hallucination condition (μ = 23.2) significantly outperforms (z = -2.2, p = 0.014) the machine condition (μ = 31.8) by 27%, and the random condition (μ = 50.3) by 54%.

Figure 5 shows the distribution of the first matching neighbor's rank for the hallucination and machine conditions for hallucinable pieces. We observe that hallucinations

[10] For any piece where three or more workers selected 'Nothing I can do here,' we applied the machine algorithm when there is ink near the edge and assumed a randomly ordered list otherwise.

[11] For each row we consider the 5 right-most pixels, and extend the right-most ink by 5 pixels (if there is any).



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

Figure 6: Example pieces and their hallucinations. In (a)–(c), the left-most piece is the piece that is being matched, and its matching neighbor is placed to its right at the correct matching position. Next to these pieces are user-submitted drawings for the piece being matched. Under each drawing is the rank of the first matching neighbor when using the drawing as a template. 'aggregated' indicates the rank of the first matching neighbor after rank aggregation is applied to the human drawings. 'machine' indicates the rank of the first matching neighbor when using the machine-generated template (not shown).

lead to matching neighbors with lower ranks much more frequently, with the most significant difference occurring in position buckets 5 and 10, representing positions 1 through 10.

To better understand the scenarios under which hallucinations are effective and ineffective, we examine how a piece's content and the workers' drawings influence a matching piece's rank, and present in Figure 6 a number of illustrative examples. In all of these examples, the left-most piece is the piece being matched. In (a), we see that hallucination works well as the human drawings correctly complete the letter 'D' in the piece being matched. In (b), hallucination performs poorly because it is hard to determine the matching letter. Following the task's instructions to not guess, the subjects were conservative and only provided a small amount of ink in their drawings. In (c), based on the piece being matched, the letter looks like it might be an 'O' or a 'C.' Most of the workers' hallucinations were conservative and would allow matches to either letter, but the markings were nevertheless rich enough to significantly decrease the search space, making hallucination effective. In (d), the piece is not hallucinable as there is no text crossing the right border of the piece. All subjects correctly marked this piece as non-hallucinable, and did not provide drawings.

Discussion and Future Work

The hallucination method draws on the complementary strengths of humans and machines to identify neighboring pieces to reconstruct a document or an artifact. The method can be generally applied to reconstruction tasks whenever there exist visual cues from piece content that humans can leverage effectively to visually describe potential matches by drawing. Experiments on a document from the DARPA Shredder Challenge show that the method is effective on a large number of instances, and can provide significant savings in time and effort over humans or machines working in isolation.

An area for future work is to identify matches more robustly when using hallucinations. One direction is to improve the matching algorithm to better handle imprecision in human drawings, and to account for similarities and differences in hallucinations when aggregating results. Another direction is to obtain better inputs, by using observations of human visual biases and other failure cases to improve instructions, train workers, and more finely identify 'hallucinable' pieces. From our experience, training workers to not over-hallucinate seems to improve results, but we also observe cases in which workers could have provided better hallucinations had they drawn more. Future work should continue to help workers learn how best to contribute; one interesting direction is to provide workers with direct feedback for hallucinations by displaying potential matches in real time.

While the specific template matching approach we implemented for puzzles in the DARPA Shredder Challenge is rather simple, template matching remains tractable even in more complex scenarios where the template may need to be rotated or scaled, and when matches occur over a dense set of points (Barnes et al. 2009; 2010; Korman and Avidan 2011). This suggests that the hallucination method remains useful and tractable in settings with more complex pieces (e.g., in 3-dimensions or with far more detail per piece) and for more expressive hallucinations (e.g., where hallucinations may describe possible combinations of patterns and colors).

The hallucination method is meant to be integrated as part of a larger human-machine system for document reconstruction, where numerous interesting questions and challenges arise for effectively and efficiently combining human and machine intelligence. In the local context of pairwise matching, one can imagine maintaining probabilistic models over the likelihood of particular matches, and using such information to determine whether more hallucination or human verification is required when optimizing the use of human effort (Dai, Mausam, and Weld 2010; Kamar, Hacker, and Horvitz 2012). In the global context, there are also opportunities to apply decision-theoretic reasoning to determine where human efforts are most needed, and to direct efforts toward making progress where it is most crucial, e.g., to obtain useful context that helps for reconstructing the rest of the document, or to prioritize the reconstruction of particular parts of the document.

Beyond pairwise matches, we believe our hallucination approach can be effectively applied at different levels of granularity through the course of a document reconstruction effort. In this direction, we are interested in exploring interactions for global hallucinations (e.g., 'this part looks like a building of sorts') and more generally discovering effective interfaces for supporting the expression of people's intuitions and knowledge from context.

Acknowledgments

We thank Zak Stone, David Parkes, Eric Horvitz, Rob Miller, Hanspeter Pfister and members of the user interface design group at MIT for helpful discussions. Haoqi Zhang and John Lai are generously funded by a NSF Graduate Research Fellowship and NDSEG Fellowship respectively.

References

Ballard, D. H. 1987. Generalizing the hough transform to detect arbitrary shapes. In Fischler, M. A., and Firschein, O., eds., *Readings in computer vision: issues, problems, principles, and paradigms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 714–725.

Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28(3).

Barnes, C.; Shechtman, E.; Goldman, D. B.; and Finkelstein, A. 2010. The generalized PatchMatch correspondence algorithm. In *European Conference on Computer Vision*.

Borda, J. C. 1781. *Memoire sur les elections au scrutin*. Histoire de l'Academie Royale des Science.

Branson, S.; Wah, C.; Schroff, F.; Babenko, B.; Welinder, P.; Perona, P.; and Belongie, S. 2010. Visual recognition with humans in the loop. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, 438–451. Berlin, Heidelberg: Springer-Verlag.

Brunelli, R. 2009. *Template Matching Techniques in Computer Vision: Theory and Practice*. John Wiley & Sons.

Castañeda, A. G.; Brown, B.; Rusinkiewicz, S.; Funkhouser, T.; and Weyrich, T. 2011. Global consistency in the automatic assembly of fragmented artefacts. *Intl. Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*.

Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popovi, Z.; and Players, F. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466(7307):756–760.

Criminisi, A.; Perez, P.; and Toyama, K. 2003. Object removal by exemplar-based inpainting. In *IEEE Computer Vision and Pattern Recognition*, volume 2, 721–728.

Dai, P.; Mausam; and Weld, D. S. 2010. Decision-theoretic control for crowdsourced workflows. In *AAAI Conference on Artificial Intelligence*.

Freeman, H., and Garder, L. 1964. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Trans. on Electronic Computers* 13:118–127.

Kamar, E.; Hacker, S.; and Horvitz, E. 2012. Combining human and machine intelligence in large-scale crowdsourcing. In *AAAMAS*.

Korman, S., and Avidan, S. 2011. Coherency sensitive hashing. In *IEEE International Conference on Computer Vision (ICCV)*, 1607–1614.

Lewis, J. P. 1995. Fast template matching. *Vision Interface*.

Mao, D.; Cao, Y.; Xu, J.; and Li, K. 2011. Object tracking integrating template matching and mean shift algorithm. In *2011 International Conference on Multimedia Technology (ICMT)*, 3583 –3586.

Parikh, D., and Grauman, K. 2011. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR'11*, 1681–1688.

Simakov, D.; Caspi, Y.; Shechtman, E.; and Irani, M. 2008. Summarizing visual data using bidirectional similarity. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 0:1–8.

Tamuz, O.; Liu, C.; Belongie, S.; Shamir, O.; and Kalai, A. T. 2011. Adaptively learning the crowd kernel. In *ICML'11*.

Toler-Franklin, C.; Brown, B.; Weyrich, T.; Funkhouser, T.; and Rusinkiewicz, S. 2010. Multi-feature matching of fresco fragments. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 29(6).

Ukovich, A.; Ramponi, G.; Doulaverakis, H.; Kompatsiaris, Y.; and Strintzis, M. 2004. Shredded document reconstruction using mpeg-7 standard descriptors. In *4th IEEE International Symposium on Signal Processing and Information Technology*, 334–337.

Vijayanarasimhan, S., and Grauman, K. 2011a. Cost-sensitive active visual category learning. *Int. J. Comput. Vision* 91(1):24–44.

Vijayanarasimhan, S., and Grauman, K. 2011b. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 1449–1456.

von Ahn, L., and Dabbish, L. 2008. Designing games with a purpose. *Commun. ACM* 51(8):58–67.

von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D.; and Blum, M. 2008. reCAPTCHA: Human-based character recognition via web security measures. *Science* 321(5895):1465–1468.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*, 2424–2432.

Wolfson, H.; Schonberg, E.; Kalvin, A.; and Lamdan, Y. 1988. Solving jigsaw puzzles by computer. *Annals of Operations Research* 12:51–64.