

Transforming Image Completion

Alex Mansfield¹

<http://www.vision.ee.ethz.ch/~mansfiea/>

Mukta Prasad¹

<http://www.vision.ee.ethz.ch/~mprasad/>

Carsten Rother²

<http://research.microsoft.com/people/carrot/>

Toby Sharp²

<http://research.microsoft.com/people/tsharp/>

Pushmeet Kohli²

<http://research.microsoft.com/people/pkohli/>

Luc Van Gool¹

<http://www.vision.ee.ethz.ch/~vangool/>

¹ Computer Vision Laboratory

ETH Zürich

Switzerland

² Microsoft Research Ltd.

Cambridge

U.K.

Abstract

Image completion is an important photo-editing task which involves synthetically filling a hole in the image such that the image still appears natural. State-of-the-art image completion methods work by searching for patches in the image that fit well in the hole region. Our key insight is that image patches remain natural under a variety of transformations (such as scale, rotation and brightness change), and it is important to exploit this. We propose and investigate the use of different optimisation methods to search for the best patches and their respective transformations for producing consistent, improved completions. Experiments on a number of challenging problem instances demonstrate that our methods outperform state-of-the-art techniques.

1 Introduction

Image completion is an important photo-editing task which involves synthetically filling a user-marked region (also called a hole) in an image, such that the image still appears natural to the user. The definition of the problem hints at its difficulty: successful completion involves reasoning about which textures and appearance look more natural and how a user defines a plausible completion. State-of-the-art image completion methods search for patches in the image that overlap consistently with existing nearby regions, locally match other image patches and produce an overall appearance that does not deviate much from regions elsewhere in the same image [5, 6, 9, 13] or different images [7].

However, in many cases, the unknown region may never have been observed. For example, the part of the clock in the hole of Fig. 1 is not repeated in the image. However, it can be filled by rotating other parts of the clock. Similarly, noise, shading, perspective distortion and texture irregularities may force us to expand our training set by considering source



Figure 1: Image completion with transformations can lead to much improved results over previous state of the art. As with all results we show, the red area shows the hole.

patches in the larger space of natural transformations (scale, rotation, brightness change *etc.*) of the training set. Our contribution is to extend image completion using the state-of-the-art method of Wexler *et al.* [13] to include transformations, and to investigate the best way to search for the best image patches and the optimal transformations to produce natural completions. We evaluate our work quantitatively as well as qualitatively in terms of the quality of results obtained.

2 Related Work

In [8], texture synthesis (a special case of image completion) is addressed using a simple greedy algorithm. Hole pixels are iteratively filled by patches most similar to their support region. Though greedy, this algorithm produces good results with few parameters. Subsequent methods such as [9, 10] predict and merge patch chunks over the hole, while ensuring smoothness at the seams; this allows larger changes and flexibility. In greedy approaches, once a few pixels are set, the fate of most others is sealed, making the pixel traversal order critical. A number of user-driven [11] and intelligent [9] heuristics have been proposed for achieving structural image coherence. An attempt to formulate and solve the global problem was first made in [8], where a modified message passing algorithm efficiently and approximately optimises the model. Differently from the MRF approach, Wexler *et al.* [13] formulated the energy as a sum of terms which capture agreement of patches over the filled hole, iteratively searching and filling to yield a consistent image. Though also approximate, this method is easy to implement and shows very promising results. The computational expense was reduced to real time speeds with the patch sampling methods of Patch Match [12].

Most previous work tries to improve the optimisation. The underlying model remains the same, exploiting image redundancy by completing hole regions such that they are similar to the training regions. Most methods consider training regions only in their original form. We also consider transforming these regions, to exploit additional redundancy. Some previous methods also use transformations, by augmenting their datasets with discrete transformations [9, 12] or using invariant features [13]. The former fundamentally limits the resolution of the search, and the latter can be unstable for small patches and holes. Generalized Patch Match [12] extends [12] to search for transformation parameters by random sampling and propagation, but few results are shown. However, as the number of parameters increases, the training set can be large and difficult to search. In addition to exploring optimisation for transformations, we perform a thorough investigation of how using transformations can affect results.

3 Problem Statement

Given an image S with a hole, we want to produce a completed target image T . A binary visibility map V takes a value 1 when a pixel is filled and 0 otherwise, as illustrated in Fig. 2.

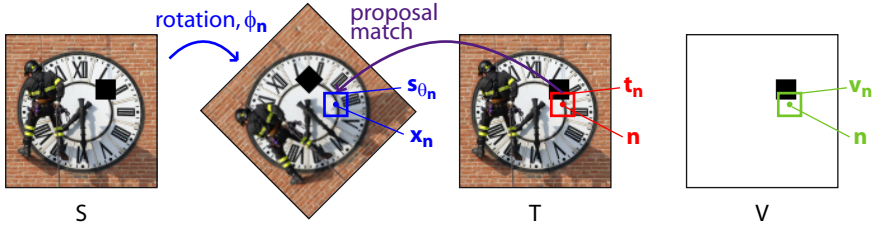


Figure 2: Our notation. For each patch in the target image T , we find matching patches in the source image S under transformations, *e.g.* rotation. The pixels still to be completed are specified in the binary visibility map V .

Notation: Our model is based on image patches, each containing P pixels. The relative position of each pixel in the patch is given by \mathbf{i}_p for all $p \in \{1 \dots P\}$ such that a patch centred at \mathbf{n} contains pixels $\mathbf{n} + \mathbf{i}_p$. We express the patch in S around \mathbf{n} compactly in vector form as $\mathbf{s}_{\mathbf{n}} = \{S(\mathbf{n} + \mathbf{i}_p) \mid p \in \{1 \dots P\}\}$, and similarly for patches $\mathbf{t}_{\mathbf{n}}$ and $\mathbf{v}_{\mathbf{n}}$ in T and V respectively. For non-integer pixel positions, the pixel value is found by bilinear interpolation.

Model: We use the model of [13], in which the completed image is defined as having global visual coherence with the source image if every patch in the completed image is contained within the visible region of the source image. Their model can be simply written as:

$$\mathcal{E}_0 = \sum_{\mathbf{n}} \min_{\mathbf{x}_{\mathbf{n}}} \|\mathbf{t}_{\mathbf{n}} - \mathbf{s}_{\mathbf{x}_{\mathbf{n}}}\| \cdot \mathbf{v}_{\mathbf{n}} \|^2, \quad (1)$$

where $\mathbf{x}_{\mathbf{n}}$ is the location of the best matching source patch for the target patch around pixel \mathbf{n} . We consider all feasible source patches *i.e.* those that do not go over the edge of the image.

Model with transformations: For a transform parametrised by a set θ , the source patch becomes $\mathbf{s}_{\theta_{\mathbf{n}}} = \{f_{\theta_{\mathbf{n}}}(S(g_{\theta_{\mathbf{n}}}(\mathbf{i}_p))) \mid p \in \{1 \dots P\}\}$, where $f_{\theta_{\mathbf{n}}}$ is the photometric transformation function acting on image pixel colours and $g_{\theta_{\mathbf{n}}}$ is the geometric transformation function acting on patch indices. Therefore, the energy now becomes:

$$\mathcal{E}_1 = \sum_{\mathbf{n}} \min_{\theta_{\mathbf{n}}} \|\mathbf{t}_{\mathbf{n}} - \mathbf{s}_{\theta_{\mathbf{n}}}\| \cdot \mathbf{v}_{\mathbf{n}} \|^2. \quad (2)$$

The overall problem is to find the optimal completed image T^* by the energy minimisation

$$T^* = \underset{T}{\operatorname{argmin}} \left(\sum_{\mathbf{n}} \min_{\theta_{\mathbf{n}}} \|\mathbf{t}_{\mathbf{n}} - \mathbf{s}_{\theta_{\mathbf{n}}}\| \cdot \mathbf{v}_{\mathbf{n}} \|^2 \right) \text{ subject to } V(\mathbf{i}) = 1 \quad \forall \mathbf{i}. \quad (3)$$

Note the two levels of optimisation. Firstly, in order to calculate the energy of a completed image T , a minimisation over the parameters $\theta_{\mathbf{n}}$ of the best matching source patch for each target patch is performed. Secondly, in order to find the optimal completed image, minimisation is performed over the completed image T .

Transformation functions: We consider the following geometric transformation:

$$g_{\theta_{\mathbf{n}}}(\mathbf{i}_p) = \alpha_{\mathbf{n}} \mathbf{R}_{-\phi_{\mathbf{n}}} \mathbf{i}_p + \mathbf{x}_{\mathbf{n}}, \quad (4)$$

where \mathbf{i}_p is a pixel in a patch around \mathbf{n} , $\phi_{\mathbf{n}}$ is the *rotation*, $\alpha_{\mathbf{n}}$ the *scale* and $\mathbf{x}_{\mathbf{n}}$ is the *translation* as defined above. A rotation of the source image by $\phi_{\mathbf{n}}$ is equivalent to rotating the patch indices by $-\phi_{\mathbf{n}}$, hence the indices are rotated by rotation matrix $\mathbf{R}_{-\phi_{\mathbf{n}}}$. Images are captured

at a fixed resolution, and hence the full scale space of patches is not available. We consider only $\alpha_{\mathbf{n}} \geq 1$, such that the patch is never increased in resolution. To aid the optimisation, we further limit the possible scale range to $2 \geq \alpha_{\mathbf{n}} \geq 1$. These parameters form part of the parameter set θ that is optimised over.

The general photometric transformation we consider is

$$f_{\theta_{\mathbf{n}}}(s) = s + \beta_{\mathbf{n}} \quad , \quad (5)$$

where s is the value of one colour channel of a pixel and $\beta_{\mathbf{n}}$ is a *brightness shift*, which completes the parameter set θ .

Note that to ‘turn off’ a transformation, we can simply fix the appropriate parameter to its default value: $\phi_{\mathbf{n}} = 0$, $\alpha_{\mathbf{n}} = 1$ and $\beta_{\mathbf{n}} = 0$.

Patch shape: Our definition is independent of the patch shape. As a default, we use square patches, as in most previous work. However, when we consider the rotation transformation, we use (discretised) circular patches, for ease of interpolation at different angles.

4 Optimisation

Finding a globally optimal completion is intractable in general, given the large solution space and the cost of evaluating the energy of one solution. The Wexler *et al.* algorithm [13] is one approximate optimisation scheme. The optimisation iterates between two steps, which we refer to as **search** and **filling**.

1. **Search:** For each patch in the current target image, the nearest neighbour patch in the source image is found, thus performing the minimisation of Equation 2. The result is a set of parameters $\theta_{\mathbf{n}}$ for target patches around each pixel \mathbf{n} , which we refer to as in [14] as the *nearest neighbour field*.
2. **Filling:** A new target image is generated from the nearest neighbour field.

These two steps are iterated until the target image converges. Note that in the search step, we always first propose a matching patch location $\mathbf{x}_{\mathbf{n}}$ and then search for the other transformation parameters to minimise the energy of this match.

The original algorithm suggests a hierarchical approximation to speed up the algorithm and provide increased robustness to the chosen patch size, but for simplicity we do not consider this extension in this work.

4.1 Search

4.1.1 Discrete Search

For discrete variables, such as location, we can perform exhaustive search over all possible values. Approximate discrete search such as Patch Match [15, 16] can maintain high accuracy at a fraction of the cost.

We also consider discrete search for approximate search over continuous variables. This method was considered in previous work for optimising over transformations [15, 16]. The number of discrete values must be chosen to trade-off speed and accuracy.

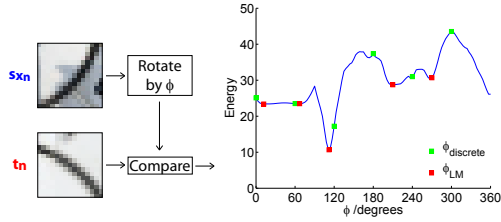


Figure 3: Continuous search with Levenberg-Marquadt (ϕ_{LM}) guarantees finding a local minimum, unlike discrete search ($\phi_{discrete}$), and in some cases finds the global minimum.

4.1.2 Continuous Search

For the best accuracy, continuous variables must be optimised by continuous optimisation methods. However, even for relatively simple transformations such as rotation, the problem is non-linear and non-convex. For continuous variables, we use the Levenberg-Marquadt algorithm. To achieve robustness, we use multiple starting points over the range of transformation parameters. A typical example is shown in Fig. 3. We use our own implementation of Levenberg-Marquadt based on [14]. In order to speed up the search, we constrain the parameters to lie within an appropriate region around the current starting point.

Jacobian calculation: The Jacobian with respect to the transformation parameters is required by Levenberg-Marquadt. The function that is squared and summed over in our model (Eqn. 2) is given by

$$e_p = T(\mathbf{i}_p) - S(g_{\theta_n}(\mathbf{i}_p)) . \quad (6)$$

Hence the Jacobian for pixel p with rotation ϕ_n varying is given by the chain rule as

$$\frac{\partial e_p}{\partial \phi_n} = \left(\frac{\partial e_p}{\partial g_{\theta_n}(\mathbf{i}_p)} \right)^T \frac{\partial g_{\theta_n}(\mathbf{i}_p)}{\partial \phi_n} = -\alpha_n \nabla S(g_{\theta_n}(\mathbf{i}_p))^T \frac{\partial R_{-\phi_n} \mathbf{i}_p}{\partial \phi_n} . \quad (7)$$

For scale, the element of the Jacobian for pixel p is derived similarly as

$$\frac{\partial e_p}{\partial \alpha_n} = -\nabla S(g_{\theta_n}(\mathbf{i}_p))^T R_{-\phi_n} \mathbf{i}_p . \quad (8)$$

4.1.3 Closed-Form Solution

For some transformations it is possible to determine the closed form optimal transformation parameter. This is possible for the brightness change transformation we consider (Eqn. 5). The contribution to the energy of matching one target patch with a source patch is given by

$$\mathcal{E}_n = \min_{\theta_n} \|[t_n - s_{\theta_n}] \cdot \mathbf{v}_n\|^2 = \min_{\theta_n} \sum_k \sum_p \left([t_{pk} - s_{pk} - \beta_n]^2 v_{pk} \right) , \quad (9)$$

where indices pk give the value at pixel p and colour channel k . Differentiating, we find

$$\beta_n^* = \frac{\sum_k \sum_p ([t_{pk} - s_{pk}] v_{pk})}{\sum_k \sum_p v_{pk}} . \quad (10)$$

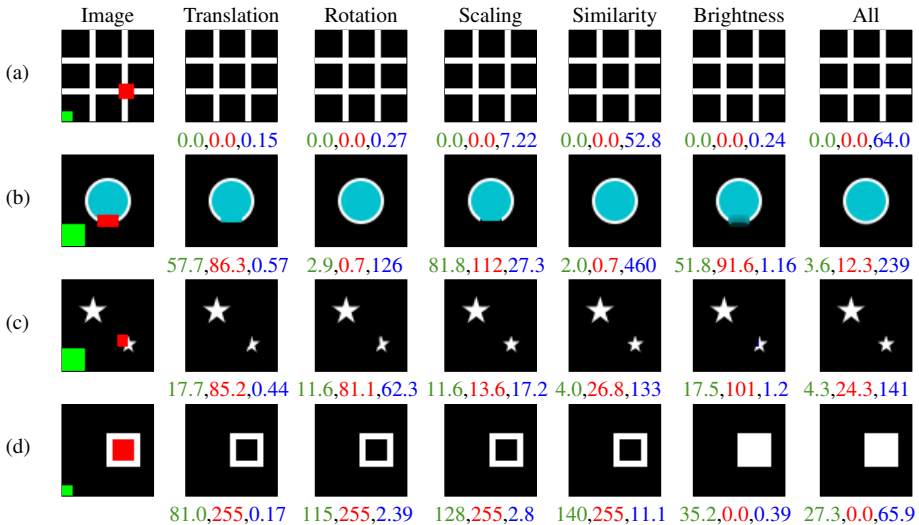


Figure 4: $Ex+c$, toy images. These results demonstrate clearly the effectiveness of our method when redundancy exists only under a transformation. In most cases, the error is minimised when only the required transformation is used; the error when using all transformations increases. For the input image (left column), the red area shows the hole and the green box illustrates the patch width. For each result we give the **energy**, **error** with respect to the ground truth and the **processing time in minutes**.

4.1.4 Constraints

In some cases, we wish to constrain a transformation parameter to lie between lower and upper thresholds. It can be shown that if the problem is convex, the new optimum parameter lies at either the unconstrained optimum or at one of the thresholds. Our objective functions are non-convex, but are modelled locally as convex by the Levenberg-Marquadt algorithm. If we reach a threshold during the optimisation, we stop and take the threshold value. We find in practice this makes our optimisation faster with no discernable loss of accuracy.

4.2 Filling

The hole is filled by combining the nearest neighbour source patches specified by the nearest neighbour field. Wexler *et al.* [13] formulate this as a weighted voting scheme. We use a similar scheme that we found to work well, in which the pixel colour taken is the median of all votes, with the median found independently for each colour channel.

Table 1: Search method nomenclature. Bold text shows novel search methods and transformations.

Name	Translation	Geometric	Brightness
Ex	Exhaustive	(none)	Closed form
PM*	Patch Match with * iterations	(none)	Closed form
Ex+c	Exhaustive	Continuous	Closed form
Ex+d6	Exhaustive	Discrete, 6 values	Closed form
PM**c	Patch Match with * iterations	Continuous	Closed form
PM**+d6	Patch Match with * iterations	Discrete, 6 values	Closed form
GPM*	Generalized Patch Match with * iterations		

	(a)	(b)	(c)	(d)
Method	Brightness Ex	Brightness Ex	Brightness Ex	Scale Ex+c
Image				
Translation only				
Transformation (as specified at the top)				
	6.01	49.9, 1.46	5.64	37.2, 9.85
	8.62	16.4, 4.48	5.93	27.7, 4.07

Figure 5: Using transformations can improve results on real images qualitatively and quantitatively. For the input image (top row), the red area shows the hole and the green box illustrates the patch width. The numbers under the images give the **error** and the **time in minutes**.

5 Experiments

We perform many experiments to evaluate the results when using our various search methods and transformations. Our nomenclature is given in Table 1. Note that we **always search over translation** *i.e.* different patches from the source image, and for each proposed translation then search over other transformation parameters where specified.¹

The true objective of image completion is to find a completion satisfactory to the user. Visual inspection is the natural test (see Figs. 4–9). We also present quantitative results: the final energy of the result (Eqn. (2)) and the RMS pixel colour error with respect to the ground truth, where possible.² Though not the objective of image completion, this error is a suitable measure of plausibility when the completion problem is sufficiently well constrained. Note that the energy function depends on the source database, transformations considered and patch size, and hence energies calculated when changing these factors are not directly comparable. However, we can always make comparisons with the ground truth error, where the ground truth is available.

5.1 Evaluating the Model

We investigate the use of transformations on toy images (Fig. 4) and realistic images (Figs. 5, 9). We tested many combinations of translation, rotation, scale and brightness. In general, rotation and brightness shift showed the most influence on output quality, but all transformations show the ability to improve the results when used appropriately. We observe a positive correlation between the energy and error for the methods over many images and patch widths, as shown in Fig. 7. These observations validate our belief that the inclusion of transformations has improved our model of the problem. Note though that we observe that the use of additional transformations must be sensibly constrained to prevent degeneracies (see Fig. 6).

5.2 Evaluating the Optimisation

Overall optimisation: Our overall optimisation algorithm guarantees only a local minimum be reached. This can be seen from our results most clearly in Fig. 7, where in general,

¹All results were created with our implementations, which are available under GNU General Public License at <http://www.vision.ee.ethz.ch/~mansfiea/transformic/>

²Note that in order to aid intuition, for all given energy and error values in the paper, we average over each colour channel and pixel, and square root. Hence all lie within the range [0, 255].

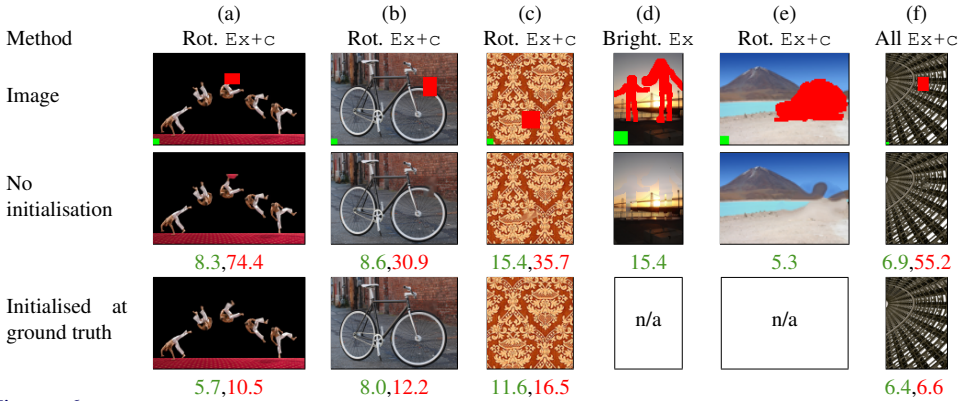


Figure 6: Failure cases. Using a patch size too small to capture the texture (a–c) or too large to exploit redundancy (d) can lead to poor completion. Allowing for additional transformations can lead to degenerate solutions (e). Our optimisation can fail for complex textures (f). For the input image (top row), the red area shows the hole and the green box illustrates the patch width. Results marked (GT) were initialised at the ground truth. We give the final energy and error with respect to the ground truth.

initialising at the ground truth leads to a lower energy result. This is shown even more clearly by Fig. 6, where initialising at the ground truth results in a better completion in each case. Only local minimisation also implies that changing the model, for example the transformations considered, may lead to a result at a local minimum with higher energy. For example in Fig. 4, using all transformations does not always lead to the lowest energy.

Despite no guarantees that the energy is reduced at each iteration, this is shown to be the case in practice as in Fig. 8. After all pixels are filled, or if we initialise with hole pixels filled from the ground truth, the energy falls until convergence after a few iterations.

Search: We observe that for more approximate search methods, the energy and error of the final result is higher. This suggests that the search limits the optimisation. This can most clearly be seen by comparison with the results initialised at the ground truth in Fig. 7, where energy at convergence is limited then only by the power of the search method.

In many cases, running the most accurate search Ex+c may not be practical due to its computational expense. However, very approximate schemes are not able to search well in the transformation space. A good balance is shown by the methods PM+c and GPM1000, which reach low energies and errors in a time much less than Ex+c.

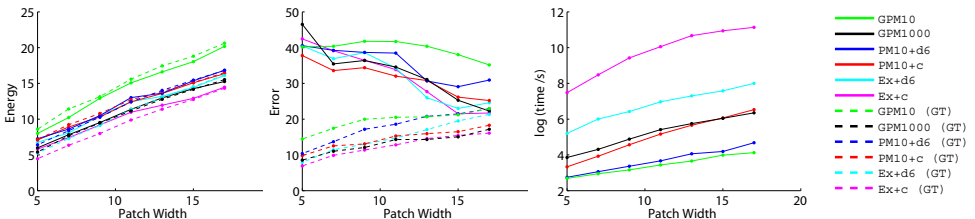


Figure 7: The average energy, error and running time against patch width for a selected set of 8 images with rotational redundancy, considering the rotation transformation. A positive correlation between energy and error is shown, with more approximate search leading to higher energies and errors, though lower running time. Using a larger patch size can make the energy harder to minimise, but can lead to even lower error in the final result. Results marked (GT) were initialised at the ground truth.

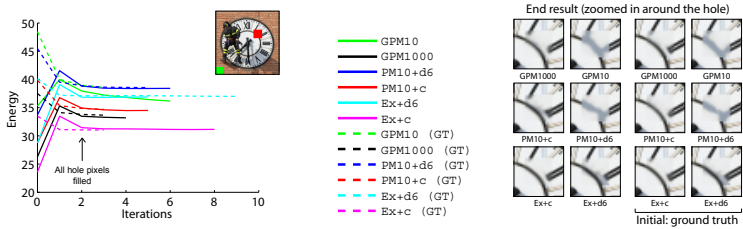


Figure 8: The energy throughout iterations of optimisation with the rotation transformation. Note that when initialising from the hole, 2 iterations are required before all hole pixels are filled. In these first iterations, as feasibility is reached, the energy may increase. After that, we observe that the energy decreases. Results marked (GT) were initialised at the ground truth.

Choosing patch size: In all our results, we manually chose an appropriate patch size. We found using patches too small or too large can lead to failures as in Fig. 6.

6 Conclusions and Future Work

We have shown that including transformations improves a model for completion of images which exhibit redundancy over such transformations. Optimising this model is difficult, but the use of continuous search over the transformation parameters brings significant improvement in accuracy compared to previous search methods, with a good approximation achieved in much less time by the Patch Match based search algorithms. We show that it is advantageous to limit the transformations considered to those that are necessary, to limit the addition of local minima. Finally, we show that the patch size is an important parameter that must be chosen in accordance with the problem image and the transformations considered.

In future work we hope to investigate more approximate search methods, such as the use of transformation invariant patch descriptors. Although we hand-picked the transformations and patch sizes for different images, we hope to explore automatic methods for choosing these, for example based on detection of repeating elements and symmetries. Finally, we would like to test our model in other related application domains, including superresolution, denoising, retargeting and shape completion.

References

- [1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. In *Proc. ACM SIGGRAPH*, Aug 2009.
- [2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patch-match correspondence algorithm. In *Proc. ECCV*, 2010.
- [3] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, September 2004.
- [4] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *Proc. ACM SIGGRAPH*, 22(3):303–312, 2003.




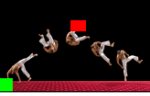


































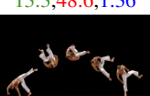

	(a)	(b)	(c)	(d)	(e)
Image					
Translation Ex					
	54.1,73.2,5.04	14.4,38.0,4.46	25.4,48.1,1.15	17.7,75.6,5.71	20.7,40.2,4.62
Rotation Ex+c					
	31.1,28.3,1920	6.7,16.8,1190	17.9,31.4,196	12.3,49.0,1450	13.6,21.6,1690
Rotation Ex+d6					
	36.9,36.5,46.7	8.1,19.5,25.6	18.3,34.0,5.77	13.7,41.9,59.5	15.6,15.6,43.8
Rotation PM10+c					
	34.5,32.3,8.51	7.1,16.7,19.3	18.0,37.4,1.24	13.4,48.9,4.06	14.9,22.0,8.12
Rotation PM10+d6					
	38.4,54.5,1.90	8.5,20.1,1.54	19.9,34.6,0.34	14.5,56.9,1.28	17.6,35.8,1.53
Rotation GPM100					
	34.0,44.3,1.95	7.1,17.8,2.38	18.2,34.8,0.42	13.3,48.6,1.36	16.1,24.4,1.64
Rotation GPM10					
	39.2,51.1,1.22	9.5,22.1,0.64	19.4,38.0,0.31	14.2,72.2,1.36	17.5,39.1,1.13

Figure 9: Results on real images with the rotation transformation. Using rotation improves the result qualitatively and quantitatively. For the input image (top row), the red area shows the hole and the green box illustrates the patch width. The numbers under the images give the **energy**, the **error** and the **processing time in minutes**.

- [5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, pages 1033–1038, Corfu, Greece, September 1999.
- [6] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. *Proc. ACM SIGGRAPH*, pages 341–346, August 2001.
- [7] James Hays and Alexei A Efros. Scene completion using millions of photographs. *Proc. ACM SIGGRAPH*, 26(3), 2007.
- [8] Nikos Komodakis. Image completion using global optimization. In *Proc. CVPR*, pages 442–452, Washington, DC, USA, 2006. IEEE Computer Society.

-
- [9] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. *Proc. ACM SIGGRAPH*, 22(3): 277–286, 2003.
- [10] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *Proc. ACM SIGGRAPH*, 24(3):795–802, 2005.
- [11] M. Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar, 2005. URL <http://www.ics.forth.gr/~lourakis/levmar/levmar.pdf>.
- [12] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24:861–868, July 2005.
- [13] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007.
- [14] Marta Wilczkowiak, Gabriel J. Brostow, Ben Tordoff, and Roberto Cipolla. Hole filling through photomontage. In *Proc. BMVC.*, pages 492–501, July 2005.
- [15] Qing Wu and Yizhou Yu. Feature matching and deformation for texture synthesis. In *Proc. ACM SIGGRAPH*, pages 364–367, New York, NY, USA, 2004. ACM.