

Inpainting from Multiple Views

Sung Ha Kang
Department of Mathematics
UCLA
skang@math.ucla.edu

Tony F. Chan
Department of Mathematics
UCLA
chan@math.ucla.edu

Stefano Soatto
Computer Science Department
UCLA
soatto@ucla.edu

Abstract

Inpainting refers to the task of filling in missing or damaged regions of an image. In this paper, we are interested in the inpainting problem where the missing regions are so large that local inpainting methods fail. As an alternative to the local principle, we make use of other images with related global information to enable a reasonable inpainting. Our method has roughly three phases: landmark matching, interpolation, and copying. The experimental results are promising.

1 Introduction

Inpainting refers to the specific image restoration task of reconstructing an image with missing or damaged regions. Digital inpainting was first proposed by Bertalmio et al. [4], and many approaches followed. There are some approaches based on solving PDEs [1, 4, 6, 8, 9, 17] and some restoration and the inpainting studies done for movies [10, 12, 15].

We are proposing a problem with large missing domains such that it is not enough to recover the region using only local information. However, the recovery process does assume there is additional information available. This additional reference image can be obtained from a movie sequence, an image of the same object taken at a different time or from a different view point.

As one possible method, we propose using landmark matching, interpolation, and inpainting. Landmark matching is often used in image registration, image morphing, and shape matching. In this way, we can enforce the relative position information explicitly, allowing the objects to move far and to be distorted. After identifying high variance points, the matching of these landmarks is achieved by using modified shape context information [3]. After the (matching) correspondence is assigned for each extracted point, we interpolate the transform information to the whole image by thin plate splines [3, 18, 21]. For the final step of

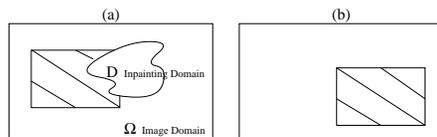


Figure 1. (a) is I_1 with D and (b) is I_2 .

inpainting, we copy the information from one image to another using the interpolated transformation.

One alternative method in the literature for point extraction to affine transformations is RANSAC [11]. However, with RANSAC many different affine transforms need to be tested to get the best match. We will carefully investigate each of the three steps; landmark matching, interpolation and inpainting the missing information.

The outline of the paper is as follows. In section 2, we present the model formulation and explain why optical flow analysis is not used. In section 3, we describe the method in detail. Examples of reconstructed results are shown in section 4, followed by concluding remarks in section 5.

2 Problem Formulation

In this paper, we assume there are only two images: one image, I_1 , with missing information and another, I_2 , with the information missing from I_1 . (Fig. 1) Let $I_1, I_2 : \Omega \rightarrow \mathbb{R}_+$ be those two images, where $\Omega \subset \mathbb{R}^2$ is the image domain. I_1 has a domain D representing the missing information. This D is called the *inpainting domain*, and we assume it has already been identified. I_2 is the second image (reference image), supposedly undamaged.

We can find the transformation g as the minimizer of a functional for measuring how well $|I_1(\vec{x}) - I_2(g(\vec{x}))|$ is satisfied. Adding the regularizing term in the form of minimizing the Total Variation of g , the functional becomes

$$\vec{g}(\cdot) = \arg \min \int_{\Omega \setminus D} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\mu(\vec{x}) + \int_{\Omega} |\nabla \vec{g}| d\mu(\vec{x}). \quad (1)$$

One approach to solving (1), when the images come from a movie sequence, is optical flow [2, 13]. However, optical flow is not very useful for large domain inpainting problems when the two images are very different.

In our approach, instead of using the optical flow model, we directly use the translational model and minimize $|I_1(\vec{x}) - I_2(g(\vec{x}))|$ and get g independently. This is equivalent to minimizing the following functional:

$$\begin{aligned} \vec{g}(\cdot) = & \arg \min \int_{Landmarks} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\vec{x} \\ & + \int_{\Omega} [g_{xx}^2(\vec{x}) + g_{xy}^2(\vec{x}) + g_{yy}^2(\vec{x})] d\vec{x}. \end{aligned} \quad (2)$$

The first term is the fitting term for landmark matching, and the second regularizing term corresponds to a thin plate spline interpolation. In our actual implementation instead of directly finding \vec{g} by solving the minimization problem (2), we decouple the two terms and first identify the landmark correspondence separately and then use this information for the spline interpolation.

3 Description of the Method

We describe each of the three main phases of our method: landmark matching, interpolation and inpainting.

3.1 Landmark matching with shape context

Landmark Extraction : There is significant literature on landmark extraction, and any effective method can be used for this purpose [5, 16]. We use salient features like corners, intersections and high curvature points by finding high local variance points. Also, we set the values next to the inpainting domain D to be zero which prevents those points from being extracted as feature points.

Modified Shape Context : After extracting the points, we need to incorporate the global positional information to match landmarks. We assign correspondence using modified *Shape Context* profiles. Shape context was introduced by Belongie et al. [3], and the simplified version is illustrated in Figure 2. After the shapes are represented by boundary points, for each point, the shape context is calculated by storing relative positions of other points to itself.

We modify the shape context to better serve our purpose. First, since we are using only salient features, for I_1 and I_2 the number of points extracted are much smaller than in [3]. Therefore, instead of 60 cells in a sector, we only use 6. In addition, we add intensity for better matching the correspondence. As an illustration, let H_p be the shape information for a point $p = (i, j)$; H_p is a vector of size 8. The first component stores the intensity at point p , ($H_p(1) = I(p)$), the second component, the intensity average of 8 neighborhood from point p , $H_p(2) = Av(p)$. The

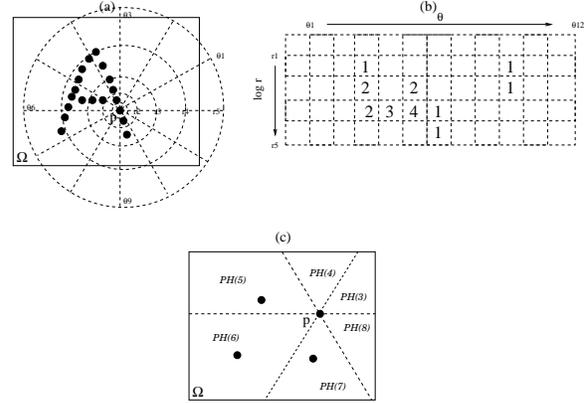


Figure 2. (a) sector used in [3], each sector corresponds to one square cell in (b). Each cell in (b) stores number of points in corresponding sector in (a). (c) Sector used for our purpose. For point p , the profile vector is $H_p = (I(p), Av(p), 0, 0, 1, 1, 1, 0)$.

remaining six components are shape-context like information storing the number of points from each corresponding sections (Fig 2 (c)).

After each profile H_p for all extracted points p from I_1 , and each profile H_q for all extracted points q from I_2 are calculated, we find the best match by comparing the vector profile and comparing the distance. For each point p_i from I_1 , we find the point q_j from I_2 which minimizes the following energy:

$$\begin{aligned} c_1 |H_p(p_i) - H_q(q_j)| + c_2 \|p_i - q_j\| + c_3 \|f(p_i) - f(q_j)\|, \\ \text{where } f(p_i) = \frac{p_i - \frac{1}{N_p} \sum_{N_p} p_k}{\text{Max}_{j \in N_p} \{p_j - \frac{1}{N_p} \sum_{N_p} p_k\}}. \end{aligned} \quad (3)$$

The first term is an L^1 -norm for comparing vector profiles, the second term is a regular Euclidean distance L^2 to keep the points from moving very far between I_1 to I_2 . The third term is for comparing the relative distance of points in I_1 and I_2 . The constants c_1 , c_2 and c_3 are weights chosen to prevent the problem of having more than one minimum for each point.

After the best match is found for each point p_i in I_1 , it is stored as $Match = \{(p_i, q_j)\}$, i.e. point p_i is matched with point q_j in I_2 . Defining $U(p_i) = \vec{q}_j - \vec{p}_i$ to be the transformation from point p_i in I_1 to point q_j in I_2 , the transformation $U(p_i)$ is assigned.

3.2 Affine Mapping and Interpolation

Compared to the image domain Ω , the number of extracted points is relatively small, yet only the matched points have directional information $U(p_i)$. Therefore, we

need to extend the matched point directions to the whole image domain. Thin plate splines [3, 18, 21] are commonly used for smooth interpolations. They minimize the bending energy of the embedded function: $E = \int_{\Omega} [f_{xx}^2(\vec{x}) + f_{xy}^2(\vec{x}) + f_{yy}^2(\vec{x})] d\vec{x}$. Solving this thin-plate spline is equivalent to interpolating with bi-harmonic radial basis functions [19]. We used $K(r) = r^2 \log|r * r|$ for the radial basis function. Since $U(p_i)$ is a 2D directional vector, thin plate spline interpolation is applied twice to each x and y coordinates separately.

The known information is $U(\vec{p}_i)$ which has directional information from point p_i in I_1 to I_2 . The interpolation function is given as :

$$U(\vec{x}) = \mathbf{A}\vec{x} + \mathbf{t} + \sum_{i=1}^n w_i \mathbf{K}(|\vec{x} - \mathbf{p}_i|),$$

where n is the number of correspondence, \vec{x} is the point (x, y) in I_1 and \mathbf{A} , \mathbf{t} , and w_i are unknowns. We first calculate \mathbf{A} and \mathbf{t} , independent of w_i , then w_i are calculated using \mathbf{A} , \mathbf{t} , K and $U(p_i)$ for x and y coordinates separately.

3.3 Inpainting - Filling in the information

Once the transformation is found, what remains is copying the information from I_2 to I_1 . This copying can be achieved by using U which is the transformation from I_1 to I_2 , defined by the interpolation of $U(p_i)$. The new recovered image is

$$I_{NEW}(\Omega) = (1 - \chi_D) I_1 + \chi_D I_2(-U)$$

$$\text{where } \chi_D = \begin{cases} 0 & (i, j) \in \Omega \setminus D \\ 1 & (i, j) \in D \end{cases}$$

χ_D is the characteristic function of D . The first term is to keep the information of $\Omega \setminus D$ from image I_1 and second term is to copy the missing information using the inverse transformation $-U$ from image I_2 .

In addition, when the background is non-uniform and moving independently of the object or there are multiple objects moving in different directions, one transformation is not enough to capture the different movements of the object and the background. For these cases, we first separate the extracted points into background points and object points, then get two or more different transformations. And using the Segmentation techniques [7] or masks of the object in I_2 , D is divided into object domain and background domain. Then, we copy the information from the object and the background separately, using two separate transformations U_{obj} (object transformation) and U_{bk} (background transformation),

$$I_{NEW}(D) = \chi_{object \cap D} I_2(U_{obj}) + \chi_{background \cap D} I_2(U_{bk}).$$

4 Experimental Results

We present some experimental results in this section. Fig. 3 shows an example of how a given image is restored using the reference image I_2 . From Fig. 3 (a), if we use local inpainting methods, the result will look like an arrow heading up to the right corner. Instead, by using additional information from I_2 , the star shape can be recovered, even though the star in I_2 is different from the star in I_1 .

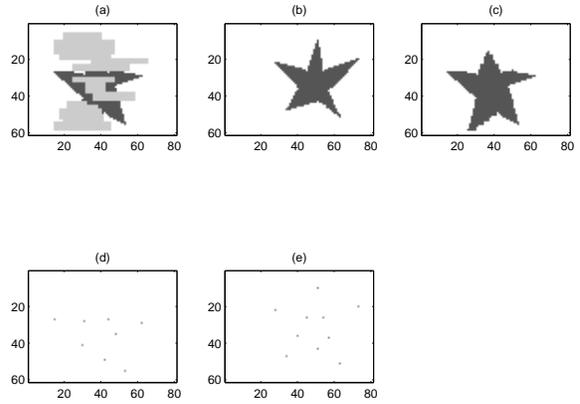


Figure 3. (a) I_1 , (b) I_2 , (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. The star in I_1 is different from the star in I_2 , however, the method recovered the result (c) as a star shape.

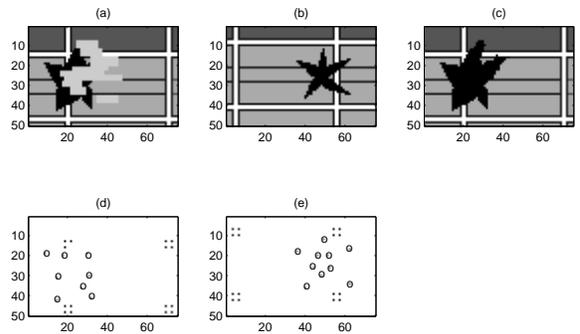


Figure 4. (a) I_1 , (b) I_2 , (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. The object is moved to different position, deformed and the background is also moved. However, both the object and the background are successfully recovered.

In Fig. 4, the object and the background are moving with different transformations from I_1 to I_2 . Therefore, points

are separated into object points and background points. (The black dots are from the background and the 'O' points are from the object.) Using the mask of the object in I_2 , inside D is separated into object domain and background, using the inverse transformation.

Also, some experiments are done using real images. In Fig. 5, I_1 and I_2 are taken at different places and different times. The missing eye is successfully recovered.

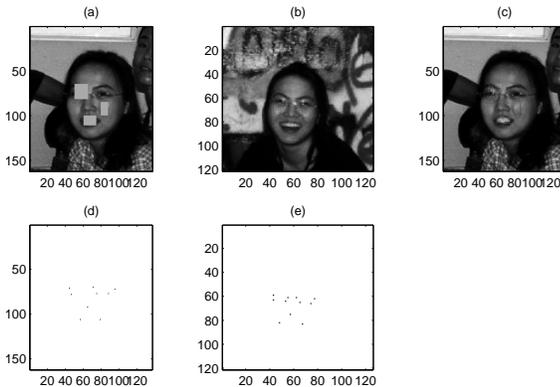


Figure 5. (a) I_1 , (b) I_2 , (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. I_1 and I_2 are taken in different places and different time. However, the missing eye and the teeth are successfully recovered.

5 Conclusion

Given a good set of landmark points, the method outlined in this paper will work well even when the two images I_1 and I_2 are quite different and with significantly large inpainting domains. The method is scale-invariant and works well even when the object has moved very far. Rotation invariance can be also achieved by rotating the shape context sectors.

One shortcoming of the method is in getting appropriate landmarks. Identifying feature points is not easy, and sometimes manual identification may be needed. Our method is as accurate as the landmark matching is accurate. However, when the landmarks are detected, using our modified shape context method gives good correspondence, making use of the global information about the object and image.

For further studies, we can combine this method with segmentation to better capture the texture of the copied quality. When copying the information, the two images' texture or lighting may be different, therefore by segmenting the objects in both images and modifying the intensity information inside the region, better restoration may be possible. More detail can be found in [14].

References

- [1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Vergera. Filling-In by Joint Interpolation of vector Fields and Gray levels. *IEEE Trans. Image Processing*, 10(8):1200–1211, 2001.
- [2] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.
- [3] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *Eight IEEE Int. Conf. on Computer Vision*, July 2001.
- [4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Balleste. Image Inpainting. *Technical report, ECE-University of Minnesota*, 1999.
- [5] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.
- [6] V. Caselles, J. M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Trans. Image Processing*, 7(3):376–386, 1998.
- [7] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001.
- [8] T. F. Chan, S. H. Kang, and J. Shen. Euler's Elastica and Curvature Based Inpaintings. *UCLA Math CAM 01-12*, 2001.
- [9] T. F. Chan and J. Shen. Mathematical models for local deterministic inpaintings. *SIAM J. Appl. Math.*, 62(3):1019–1043, 2001.
- [10] L. Chanas, J. Cocquerez, and J. Blanc-Talon. Highly degraded sequences restoration and inpainting. *preprint*, 2001.
- [11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [12] F. Guichard. A morphological, affine and galilean invariant scale-space for movies. *IEEE Trans. Image Process.*, 7(3):444–456, 1998.
- [13] B. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [14] S. H. Kang, T. F. Chan, and S. Soatto. Landmark based Inpainting from Multiple Views. *UCLA Math CAM 02-11*, 2002.
- [15] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Interpolation of Missing Data in Image Sequences. *IEEE Trans. Image Process.*, 11(4):1509–1519, 1995.
- [16] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joint Conf. on Art. Intell.*, 1981.
- [17] S. Masnou and J. M. Morel. Level Lines based Disocclusion. *Proceedings of 5th IEEE Int'l Conf. on Image Process., Chicago*, 3:259–263, 1998.
- [18] M. Powell. A thin plate spline method for mapping curves into curves in two dimensions. *Computational Techniques and Applications (CTAC95)*, 1995.
- [19] M. J. D. Powell. The Theory of Radial Basis Function Approximation. *Advances in Numerical Analysis*, II:105–210, 1990.
- [20] C. Tomasi and J. Shi. Image deformations are better than optical flow. *Mathematical and Computer Modeling*, 24(5/6):165–175, 1996.
- [21] G. Wahba. Spline models for observational data. *SIAM*, 1990.