Landmark based Inpainting from Multiple Views

Sung Ha Kang *and Tony F. Chan †and Stefano Soatto ‡

April 4, 2002

Abstract

Inpainting refers to the task of filling in missing or damaged regions of an image. Most existing inpainting algorithms are local in nature and extrapolate information from neighboring pixels into the inpainting regions. In this paper, we are interested in the inpainting problem where the missing region are so large that these local inpainting methods fail. As an alternative to the local principle, we assume that there are other images with related global information to enable a reasonable inpainting. These additional images could be from a movie sequence, an image of the same object from a different time and a different viewpoint, or an image of a similar object.

Our method has roughly three phases: landmark matching, interpolation, and copying. For the landmark matching, modified shape context information is used to exploit the global information. Then matched information is interpolated (and regularized) using thin plate splines. Finally, we copy the information from one image to another. Using landmark matching and interpolation, allows the missing regions to be significantly larger compared to the local inpainting methods, and can be used when the object is distorted from one image to another. The experimental results are promising.

1 Introduction

Inpainting refers to the specific image restoration task of reconstructing an image with missing or damaged regions. For example, when part of a painting is damaged with a crack or a part of a photograph is covered by some letters, inpainting algorithms use local information to fill in the crack, gap and undesirable letters.

Digital inpainting was first proposed by Bertalmio et. al. [4], and many approaches followed. There are some approaches based on solving PDEs including Masnou and Morel's level line based disocclusion [26] and Chan and Shen's TV minimizing inpainting [10]. More literature can be found in [1, 9, 11, 12].

These inpainting methods only make use of local information. Therefore, the missing region needs to be small enough to have good results. Fig. 1 is one of the results from curvature based inpainting [9], which is an improved version of *Total Variation* minimizing inpainting [10]. With Fig. 1 (a), it is reasonable to think the object is a perfect square, however, even with curvature inpainting in (c), it is not possible to recover the square corner. In order to recover the corner, global information or additional shape information is needed. These are the problems we are interested in this paper.

We are proposing a problem with large missing domains such that it is not enough to recover the region only using local information. However, the recovery process does assume there is additional information available. This is reasonable since when humans look at an image with a missing domain, we automatically recover image without any difficulty. As in Fig. 2, it is reasonable to think that (a) is an image of a checkerboard box, therefore, (b) can be used as the reference. In addition, when there is a damaged image of a known object or a building, similar photo can be used as a reference. An image of reconstructed building (d) can be used to retouch the same building in image (c).

There are some restoration and inpainting studies done for movies (image sequence). Guichard [16] introduced AMG model to deal with movie images to successfully denoise and inpaint movies. Kokaram et

*Department of Mathematics, UCLA, 405 Hilgard Ave.,Los Angeles, CA 90095-1555, skang@math.ucla.edu

†Department of Mathematics, UCLA, 405 Hilgard Ave.,Los Angeles, CA 90095-1555, chan@math.ucla.edu. This research was supported in part by NSF grant, number DMS-9973341 and from ONR under N00014-96-1-0277

‡Computer Science Department, UCLA, Boelter Hall 4531, Los Angeles, CA 90095-1596, soatto@ucla.edu

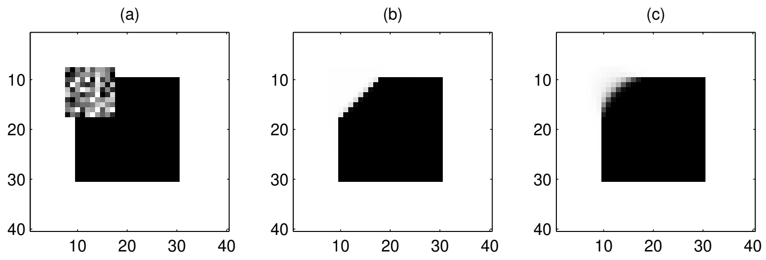

Figure 1: (a) is the original image with the square noisy area as the region to be filled, (b) is the result using Total Variation inpainting [10], and (c) is the result using curvature based inpainting [9].

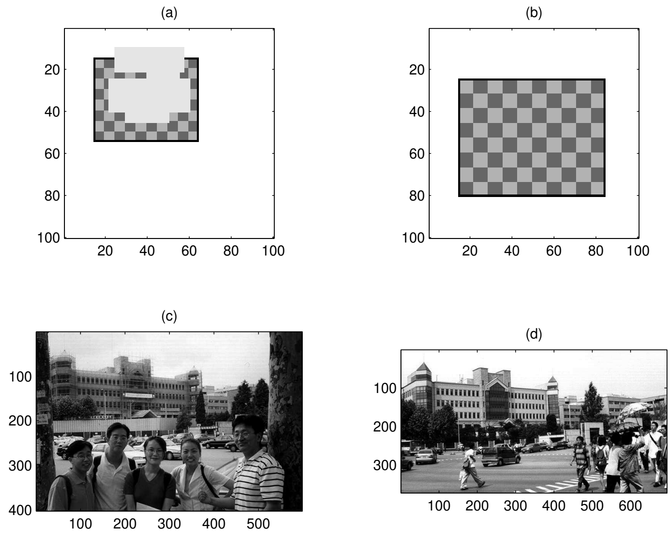

Figure 2: (a) has a gray missing region, and (b) is an image of the same object in different scale. (b) can be used for inpainting image (a). (c) and (d) are images of the same building taken at different times. In (c) the building is under construction and (d) is after the construction is completed. (d) can be used as reference image to replace the building in (c).

al. [18, 19] have studied detection and interpolation of missing data in image sequences. Chanas et al. [13] proposed using 3D diffusion for recovering image sequences, considering time as the third axis. However, in our proposed problem, the reference image and the damaged images are not assumed to be in any sequence. We are allowing the additional image to be distorted, image from any different viewpoint, or image of similar object, not the same.

As one possible method, we propose using Landmark matching, interpolation, and inpainting. Landmark matching is often used in image registrations, 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 spline [3, 29, 32]. For the final step of 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 [15]. RANSAC will automatically extract points, pick a small number of points from that set of extracted points, get the affine transformation, and then choose the best affine transformation which minimizes the difference. This method is useful for the case with a small number of outcast points and for affine transformation.

However, with RANSAC, many different affine transforms need to be tested to get the best match. We carefully looked into each steps, points extraction, matching and interpolation to better understand the image information and carefully take care of the inpainting region. In addition, when the object is deformed it is better to add spline interpolation together with affine transformation.

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. Let $I_1, I_2 : \Omega \longrightarrow \mathbb{R}_+$ 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 (Fig. 3). I_2 is the second image (reference image), supposedly undamaged.

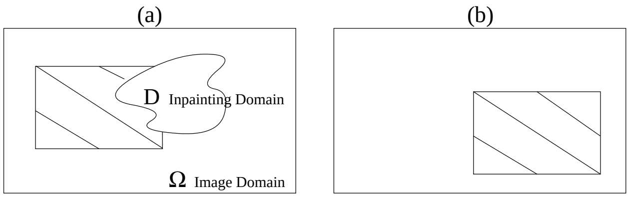

Figure 3: Setting of the problem : (a) is I_1, the original image with D as the inpainting domain. (b) is I_2 with a complete object information, which can be used to inpaint I_1.

We are looking for a map $g : \Omega \longrightarrow \mathbb{R}^2$ which satisfies,

$$I_1(\vec{x}) = I_2(g(\vec{x})) \quad \forall\, x \in \Omega \cap g^{-1}(\Omega). \tag{1}$$

We can find this g as the minimizer of a functional for measuring how well (1) is satisfied. Adding the regularizing term to the minimizing Total Variation of g, the functional becomes

$$\vec{g}(\cdot) = \arg\min \int_{\Omega \backslash D} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\mu(\vec{x}) + \int_{\Omega} |\nabla \vec{g}| d\mu(\vec{x}) \tag{2}$$

One approach to solving (2), when the images come from a movie sequence, is optical flow. Equation (1), and the first term in equation (2),can be seen as the brightness constancy constraint in computer vision. When the two images are from a movie sequence and are consecutive, we can define I_1, I_2 to be $I_1 = I(\vec{x}, t)$ and $I_2 = I(g(\vec{x}), t+1)$, where t is time. In the simplest case when g is of the form $g(\vec{x}) = \vec{x} + \vec{u}$, the vector field $\vec{u}$ is called the *optical flow field* [2, 5, 17, 22], and it satisfies the following optical flow equation : $\nabla I \cdot \vec{u} + I_t = 0$. One approach to this problem can be computing u outside the inpainting domain D, and enforce the smoothness of u throughout the whole domain Ω:

$$\vec{u}(\cdot) = \arg\min \int_{\Omega \backslash D} |\nabla I \cdot \vec{u} + I_t| d\mu(\vec{x}) + \int_{\Omega} |\nabla \vec{u}| d\mu(\vec{x}).$$

Then, once the vector field $\vec{u}(x)$ in $\Omega \backslash D$ has been filled in, the restored image can be obtained as $I(t, \vec{x}) = I(t+1, g^{-1}(x))$, $\forall\, x \in D$ where $g(\vec{x}) = \vec{x} + \vec{u}(\vec{x})$.

Using optical flow might work for movie sequence, when $I_1 = I(t)$ and $I_2 = I(t+1)$ are not very different. However, optical flow is not very useful for large domain inpainting problems when the two images are very different. One major disadvantage is the local aperture problem: "Flow" related methods depend only on local information, not on global position information (see Fig. 4). If the object moves too far, even with

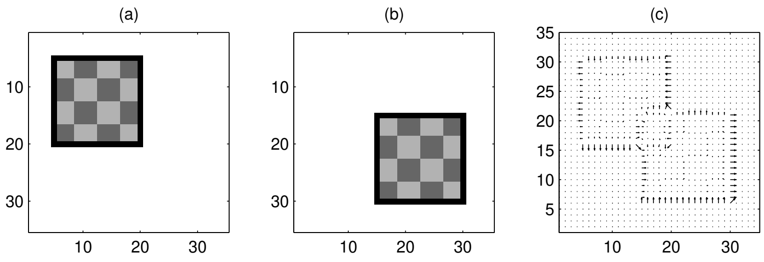

Figure 4: (a) is I_1, (b) is I_2 and (c) is optical flow. From I_1 to I_2, the ideal directional field should be uniform arrows of length $10\sqrt{2}$, in down right direction. However, calculated optical flow result (c) is far from what is expected.

multiscale analysis [21, 24], it may be difficult to capture the right positions. In addition, in order to get the directional information with discontinuities, sophisticated numerical schemes like min-mod or Eno-like scheme are needed to better capture the accurate flow estimation.

In addition, the g from equation (2) could be a finite-dimensional group, therefore, there may not be a need to regularize with respect to g. The second term from the cost functional, equation (2), can introduce a bias towards small displacements.

In our approach, instead of using the optical flow model, we directly use the translational model and minimize

$$|I_1(\vec{x}) - I_2(g(\vec{x}))| \tag{3}$$

and get g independently. To find the appropriate g for equation (3), we use landmark matching, and spline interpolation for regularization (thin plate spline approximation). Landmarks are the points extracted from the images which have the directional information and thin plate spline is needed to extend those directional information to the whole domain. This is equivalent to minimizing the following functional:

$$\vec{g}(\cdot) = \arg\min \int_{Landmarks} |I_1(\vec{x}) - I_2(g(\vec{x}))| d\vec{x} + \int_{\Omega} [g_{xx}^2(\vec{x}) + g_{xy}^2(\vec{x}) + g_{yy}^2(\vec{x})] d\vec{x}. \tag{4}$$

The first term is the fitting term for landmark matching, and the second regularizing term corresponds to thin plate spline interpolation. In our actual implementation, instead of directly finding $\vec{g}$ by solving the minimization problem (4), 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

In this section, we describe each of the three main phases of our method: landmark matching, interpolation and inpainting.

First, any method for extracting points can be used for landmark extraction. After the landmark points are extracted, we carefully match the landmarks between I_1 and I_2. We use shape context information as well as intensity information to match the points. The shape context information contains global position information for each point relative to all the other extracted points.

The number of extracted points are small compared to entire discretized image domain; yet, these are the only points with correspondence information. Therefore, we need to interpolate the directional information to the whole domain. This is similar to sparse data interpolation, for which radial basis functions are commonly used. We use thin plate spline for this interpolation.

After determining the interpolation, which gives the transformation for every points in the image domain, we finally copy the information from one image to another. When there are more than one object or the object and the background are moving in separate direction, mask of the object is needs to be identified to get better results. Also, local inpainting methods on I_1 can be combined with the copying information from I_2 using interpolated transformation.

3.1 Landmark matching with modified shape context

Landmark Extraction : There is a significant literature on landmark extraction, and any effective method can be used for this purpose. We tried to use salient features like corners, intersections and high curvature points by finding high local variance points and thresholding it with some constant. However, points near the inpainting domain D of the damaged image I_1 can be mistaken as salient features. Therefore, we set the values on D to be zero which prevents the points next to the inpainting domain from being extracted as feature points. We extract the high variance points by thresholding the following V with some constant,

$$V(i,j) = \sqrt{\sum_{(k,l)\in N} \chi(k,l)\cdot\chi(i,j)\cdot(I_1(k,l)-I_1(i,j))^2}$$
$$where\quad \chi = \begin{cases} 1 & (i,j)\in\Omega\backslash D \\ 0 & (i,j)\in D \end{cases},\quad and\quad N=\{(i+1,j),(i-1,j),(i,j+1),(i,j-1)\}.$$

Shape Context : After the landmark extraction, we need to incorporate the global positional information to match landmarks. Therefore, we assign correspondence using modified *Shape Context* profiles. Shape context was introduced by Belongie et al. [3], to incorporate global information of shape to better match and compare shapes. First the shapes are represented using points along the boundary. Then for each point p, the shape context is calculated storing relative positions of other points to itself. The simplified version is illustrated in Fig. 5. From the sectors in (a), the number of points contained in each sectors are stored at each corresponding cells in (b). By using histogram comparison of these profiles for all the points, the authors were able to find correspondence between the points from two shapes.

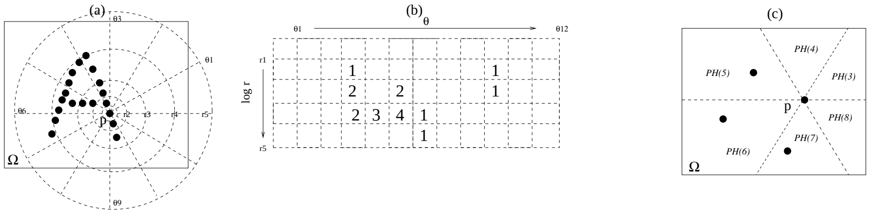

Figure 5: (a) sector used in [3]. Each sector in (a) corresponds to one square cell in (b). The number in each cell in (b) is the number of points in each corresponding sector. (In [3], histogram is used instead of the number of points.) (c) is the sector used for our shape context, for point p, the profile vector is $H_p = (I(p), Av(p), 0, 0, 1, 1, 1, 0)$. (The third through eighth components store the number of points in the each sectors.)

Modified Shape Context : Despite the heavy amount of calculations and much information to be stored, using relative position information is useful in our problem. Therefore, 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 numbers 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. (Belongie et al. [3] did not use the intensity information. The points are representing the boundary of the shape and intensity information is not necessary for shape matching.)

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, $H_p(1) = I(p)$, and the second component $H_p(2) = Av(p)$ stores the intensity average of 8 neighborhood $\frac{1}{8}\sum_{(k,l)\in A} I_{(k,l)}$ where $A=\{(i-1,j-1),(i-1,j),(i-1,j+1),(i,j-1),(i,j+1),(i+1,j-1),(i+1,j),(i+1,j+1)\}$. The remaining six components are shape context like information storing the number of points from each corresponding sections. For example, from Fig. 5(c), the profile for point p is $H_p = (I(p), Av(p), 0, 0, 1, 1, 1, 0)$.

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:

$$\begin{aligned} &c_1|H_p(p_i) - H_q(q_j)| + c_2\|p_i - q_j\| + c_3\|f(p_i) - f(q_j)\| \\ &where \quad f(p_i) = \frac{p_i - \frac{1}{N_p}\sum_{N_p} p_k}{Max_{j\in N_p}\{p_j - \frac{1}{N_p}\sum_{N_p} p_k\}} \end{aligned} \tag{5}$$

The first term is an L^1-norm for comparing vector profile, the second term is a regular Euclidean distance L^2 for keeping the points from moving very far between I_1 to I_2. The third term[1] 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. The number of correspondence is small compared to the whole image domain, since we only extract a small number of salient features, and not all the points extracted are matched between I_1 and I_2. Therefore, we need to interpolate the information to the whole domain (next section). 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.

Remarks

1. Depending on the size of the image,'block' instead of landmark point may be used. Size of the block needs to be determined, which will guarantee all the points inside the block move to same direction. In addition, when the two images are taken from movie sequence, information of extracted points from I_1 can be used to better extract the points from I_2.

2. Since I_1 have a large inpainting domain D, it is efficient to work with a smaller number of extracted points. For example, in Fig. 6, let the point p in I_1 be the same point with point q in I_2. If we extract many number of points, I_1 is missing many points around the inpainting domain, thus, the point p and q will have very different shape context profiles and result in the point p not corresponding to the right match q. Therefore, it is best to work with a smaller number of points.

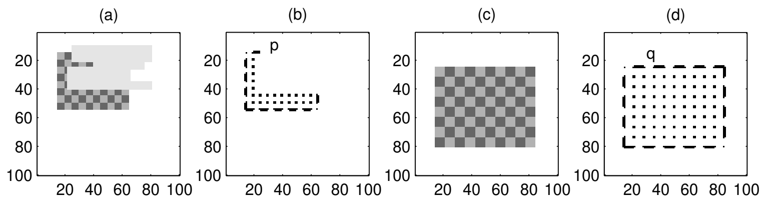

Figure 6: (a) is I_1 with a gray inpainting domain and (b) shows landmarks extracted from I_1. (c) is I_2 and (d) landmarks from I_2. The point p in (b) and q in (d) are the same point from the object, therefore, need to be matched. However, with too many landmarks extracted, the shape context information will be highly different from point p to point q. It is better to extract a smaller number of landmarks.

3. For the matching landmark correspondence, in some literatures, the Hungarian method is used. From two sets of same number of points, by trying possible permutations, the method looks for the best match which minimizes a certain energy functional. However, in our problem the number of points extracted from the two images I_1 and I_2 are most likely different due to the existence of D, also adding dummy points will result in unstable solutions. Therefore, instead of using permutation, we pick the

[1] $f(p_1)$ is the relative distance from point p_1 to the center of all p_is, i.e. $\frac{1}{N_p}\sum_{N_p} p_i$. This is scaled with maximum distance from center to furthest extracted points. Therefore, $0 \le f \le 1$. This term is effective when the object moves very far, i.e. all the points move far, yet, keeping the similar distance relative to each other. The scaling term $Max_{j\in N_p}\{p_j - \frac{1}{N_p}\sum_{N_p} p_i\}$ in the denominator is for scale invariance.

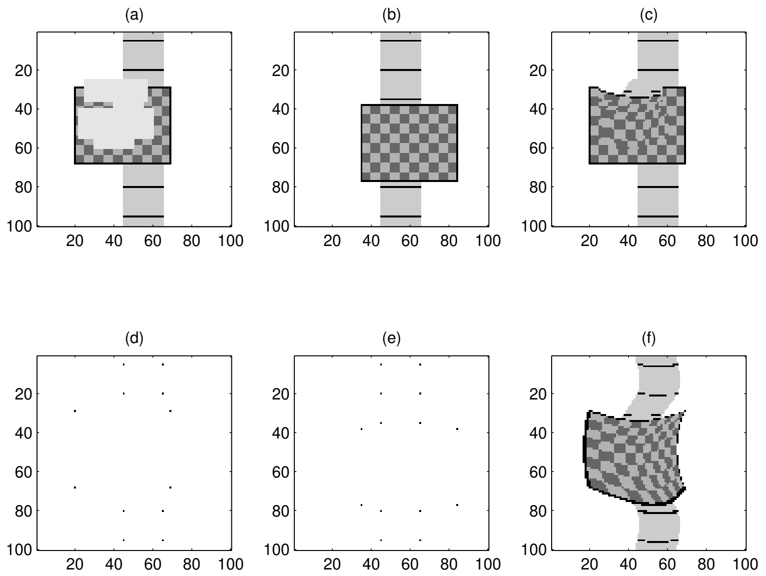

Figure 7: (a) is I_1 with the irregular gray area as the inpainting domain and (b) is I_2. The checkerboard is the object and the gray bar with horizontal strips is a part of the background. (c) is the recovered I_1 using (d) and (e) as the landmarks for I_1 and I_2 respectively. (f) shows the transformation U. Since the background is not uniform, if we use one transformation for all the extracted landmarks, the spline interpolation (f) gets highly distorted and the result in unsatisfactory recovery of I_1 in (c).

best match for each landmark p for I_1, that is, for each p, all landmarks q from I_2 is compared and get only the best match. Not all points will necessarily be assigned but only the best match.

4. When picking the best match for each landmarks, there are cases when one point p_1 from I_1 corresponds to multiple q's from I_2 (or vise versa) or due to occlusion, the appropriate match q_{right} from I_2 is hidden and another point q_{wrong} is matched for the best correspondence to p. Therefore, we also match the points from I_2 to I_1 as well, i.e. get both $MatchP = \{(p_i, q_j)\}$ and $MatchQ = \{(q_k, p_l)\}$ and use the correspondence which in both sets of matches (I_1 to I_2, $MatchP$ and from I_2 to I_1, $MatchQ$). For example, when the set $MatchP$ includes multiple correspondence and a wrong correspondence $\{(p_1, q_1), (p_2, q_1), (p_3, q_{wrong})\}$, the set $MatchQ$ including $\{(q_1, p_1), (q_{wrong}, p_4)\}$ will help eliminate the incorrect correspondence (p_2, q_1) and (p_3, q_{wrong}).

5. From I_1 to I_2, when the object and background are moving independently, the object and background points need to be separated to get better transformation. When the object moves far away in a non-uniform background, one transformation is not enough, Fig. 7 illustrates this point. Therefore after the landmark extraction, we need to separate the points into two classes : object points and background points. One way is to use intensity similarity and group the them to object points and background points, then find the correspondence only from each group (of same or similar intensity). For some cases, when the objects and the background have similar intensities, the landmarks may need to be manually identified and separated.

3.2 Affine Mapping and Interpolation

Compared to the image domain Ω, the number of extracted landmarks is relatively small, yet only the matched landmarks have directional information $U(p_i)$. Therefore, we need to interpolate the matched point directions to the whole image domain. This is similar to sparse data interpolation, for which radial basis functions are commonly used. In particular, we used thin plate spline for this interpolation.

Thin plate splines [3, 29, 32] are commonly used for smooth interpolations. They minimizes 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 [28]. 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 $U(\vec{p_i}) : R^2 \to R^2$ which have directional information from point p_i in I_1 to I_2. The interpolation function is given as :

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

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, i.e. solve $U(\vec{p_i}) = \mathbf{A}\vec{p_i} + \mathbf{t}$ using only the known values of $U(\vec{p_i})$ and $\vec{p_i}$. (We used Gaussian elimination and, when the matrix is not square, the solution in the least squares sense.) Then the weights w_i are calculated using $\mathbf{A}$, $\mathbf{t}$, K and $U(p_i)$ for x and y coordinates separately [3, 29, 32].

3.3 Inpainting - Filling in the information

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

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

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

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

In some cases, the recovered image might have some intensity discontinuity due to simply copying the information. For such cases, combining local inpainting techniques will help improve the results. We can incorporate *Total Variation*(TV) inpainting in addition to the transformation interpolation by minimizing following:

$$a_1 \int_\Omega |\nabla I| + a_2 \int_{\Omega/D} |I - I_1| + a_3 \int_D |I - I_2(-U(\vec{x}))|.$$

The first term is the TV minimization enforced in the whole image domain Ω. The second term is the fitting term so that the image outside the inpainting domain Ω/D will be close to the original I_1. The third term is the transformation inpainting, which copies information from I_2 to I_1.

As in Remark 5 of Section 3.1, when the background is non-uniform and moving independently to 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 need to consider the object and the background (or multiple objects) separately. We first separate the extracted points into background points and object points, then get two or more different transformations. In this way, we can insure better transformations corresponding to appropriate movements of the objects and the background separately.

With two (or more) different transformations, when we fill in the information, extra caution is needed. The inpainting domain should be separated into two regions, corresponding to the object and the background, see Fig. 8. Some techniques like optical flow segmentation or motion estimation [20, 27, 30] can be used for this purpose. However, these techniques are computationally very expensive and we prefer to use a heuristic method that gives good experimental results. We use, segmentation [8] or masks of the object in I_2. With this mask of the object, we use the object transformation U_{obj} and inverse transform this mask back to I_1. This transferred mask shows the approximated object region in I_1, also inside the inpainting domain D. Therefore, the transformation inpainting for the inpainting domain D becomes,

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

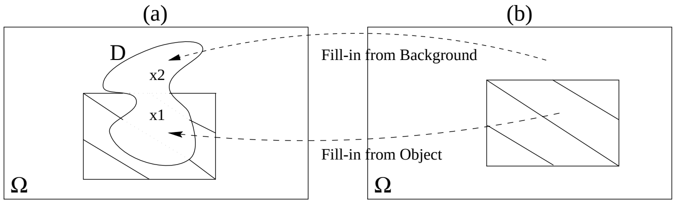

Figure 8: (a) is I_1 and (b) is I_2. When the inpainting domain in (a) covers both the object and the background, extra caution is need to fill-in the region with appropriate information either with the object transformation or the background transformation.

copying the information from the object and the background separately – using two separate transformations U_{obj} (object transformation) and U_{bk} (background transformation).

4 Experimental Results

We presents some experimental results in this section. Some results are simple synthetic objects over an uniform background, some are objects in non-uniform background, and some examples using real images are presented.

Fig. 9 shows a couple of results when I_1, (a) and (f), are a textured box with a large inpainting domain. The large inpainting domain covers most of the texture as well as one corner of the box. The first row shows results using (c) as reference image and the second row using (h). This object in (c) is in different position and the object in (f) is in different scale and position. The recovered results (e) and (j) show that the method can recover the checkerboard texture as well as the missing corner. By using modified shape context information without distance sectors, the scale change is automatically taken care of.

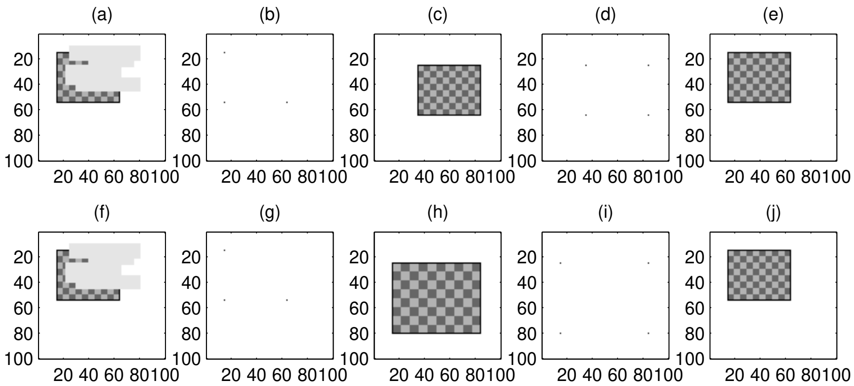

Figure 9: (a) is I_1 with the gray inpainting domain covering most of the texture and one corner of the box. (b) is the landmarks from I_1. Using (c) as reference image I_2 and (d) as landmarks of I_2, the recovered result is shown in (e). Images (f) to (j) shows another results using (h) as I_2. The method successfully recovers the texture and the corner of the box.

Our method can also be used for missing shape matching. Fig. 10 shows an example of how a given image is restored using the reference image I_2. From Fig. 10 (a), if we use local inpainting method, the result will look like an arrow heading up to the right corner. Instead, by using additional information from I_2, star shape can be recovered, even though the star in I_2 is different from the star in I_1. However, as in Fig. 11, when the reference image, I_2, is very different from I_1, landmark matching fails to find a good correspondence. Fig. 11 (c) shows a different result compared to Fig. 10 (c). This is reasonable, since I_1 and I_2 are very different, and hard to be seen as the same object.

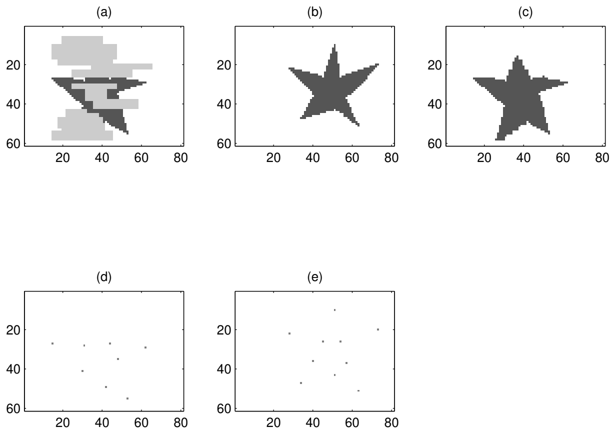

Figure 10: (a) I_1 can be recovered relatively similar to I_2 in (c). (e) is the recovered result. (b) and (d) are landmarks from I_1, I_2 respectively. The star in I_1 is different from the star in I_2, however, the method recovered the result (e) as a star shape.

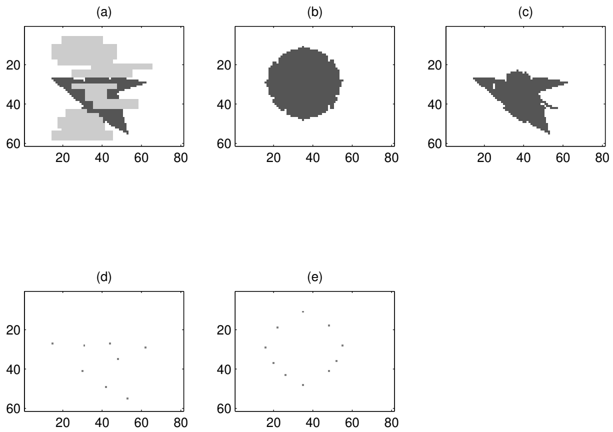

Figure 11: (a) is the same I_1 as in Fig. 10. However, due to the highly different I_2 in (c), the result (e) is unsatisfactory. (b) and (d) are landmarks from I_1, I_2 respectively.

In Fig. 12, 13,and 14, it shows results for images with object and background. After the landmarks are extracted, the points are separated into object points and background points. (The black dots are from the background and the 'O' points are from the object.) In Fig. 12, the checkerboard is in different position (c) and in different scale (h) in steady background. Using two separate transformations for the object and the background, the results successfully recovered the checkerboard texture and the corner. The stars in Fig. 13 (a) and (c) are different, and the background in more complicated. However, the method recovers the object similar to star in I_2, and the text background is successfully recovered using information from I_2. In Fig. 14, the stars in (a) and (b) are very different and the background is also shifted independent to the object. Both the object and the background are successfully recovered, using object and background transformations.

Next, we show some experiments using real images. Fig. 15 shows images of UCLA Royce Hall, (a) with couple of white disks as inpainting domains. Using (b), an image of Royce Hall taken from different time and different viewpoint, as reference image, (c) shows the recovered result. Missing windows at top left corner of the building is recovered and the three missing pole in front of the building is also added using information from I_2. In Fig. 16, I_1 and I_2 are two photo of same person taken at different places and at different times. With local inpainting methods, the missing eye can not be recovered. However, using information from I_2, the missing eye and the teeth are successfully recovered. In addition, in Fig. 17, two photos are of two

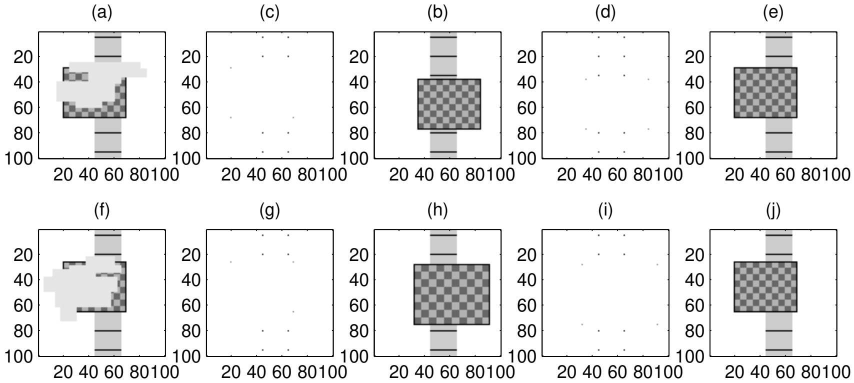

Figure 12: Two results are shown for inpainting an object moving in a non-uniform steady background. In the first row, (c) shows I_2 with the checkerboard box moved, and the second row, (h) shows the I_2 with the object moved and scaled. In I_1, (a) and (f), the large inpainting domain covers most of the checkerboard texture, one corner of the box as well as some part of the background. Using separate transformations for the object and the background, the inpainted results successfully recovered the texture and the corner.

different people, taken at different time and place. However, using the information from the other photo, I_1 can recover the shape of the eyes. However, due to the intensity difference between I_1 and I_2, the recovered results show some discontinuities, post processing with local inpainting method will help improving these results.

We can apply this method to modify the undesirable photos. In Fig. 18, I_1 and I_2 are taken in the year 1998 and 2001 respectively (at Yonsei University, Seoul, Korea). I_1 has friends but the building is under construction. Using the whole building as inpainting domain D, I_1 is recovered using the more recent picture, I_2, of the completed building. The two images are taken from two different time and positions. In Fig. 19, 20, I_1 and I_2 are taken consecutively. In I_1, the person in the middle has an odd facial expression and in I_2, the person in the middle is slightly occluded by a person from left. Using I_2 as reference and using the whole face as the mask to be inpainted, I_1 can be modified.

5 Conclusion

This method allows the missing region to be significantly larger compared to local inpainting methods, since there is additional information available. Using the modified shape context method [3] helps to incorporate the global shape information. Thin plate spline includes affine transformation with spline interpolations, which help to find transformations for distorted objects from one image to another.

Given a good set of landmarks, this method 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 rotational invariance can be also achieved by rotating the shape context sectors.

One shortcoming of the method is in getting appropriate landmarks. Identifying feature points may not be easy, and sometimes manual identification may be needed. Our method is as accurate as the landmark matching is accurate. However, when the points 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.

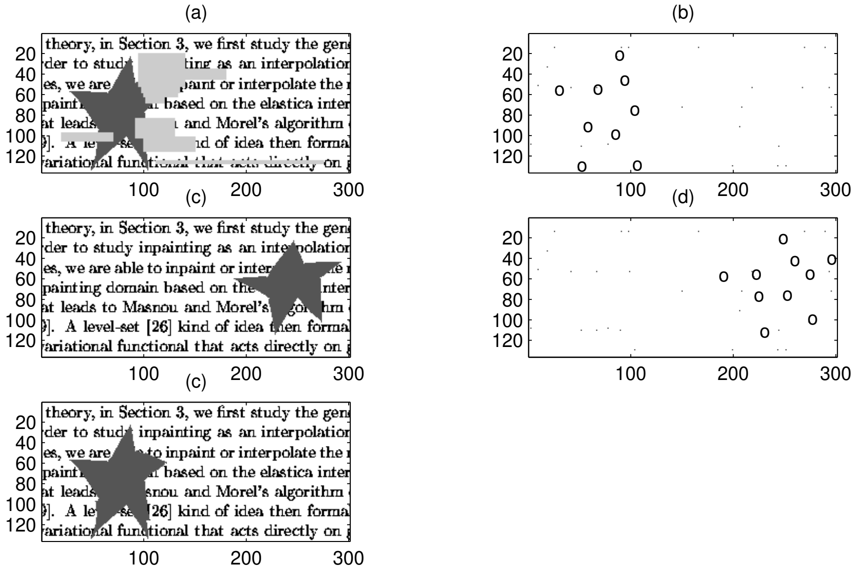

Figure 13: The result with deformed objects are shown. (a) I_1, (c) I_2, (b) and (d) the landmarks from I_1 and I_2 respectively, and (c) is the recovered result. The object in I_1 and I_2 are in different positions and deformed. The method recovers the object according to the given object in I_2, and text background is successfully recovered as well, using information from I_2.

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] A. Borzi, K. Ito, and K. Kunisch. Optimal control formulation for determining optical flow. *preprint*, 2001.

[6] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[7] V. Caselles, J. M. Morel, and C. Sbert. An axiomatic approach to image interpolation. *IEEE Trans. Image Processing*, 7(3):376–386, 1998.

[8] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. Image Process.* , 10(2):266–277, 2001.

[9] T. F. Chan, S. H. Kang, and J. Shen. Euler's Elastica and Curvature Based Inpaintings. *UCLA CAM 01-12*, 2001.

[10] T. F. Chan and J. Shen. Mathematical models for local deterministic inpaintings. SIAM J. Appl. Math, 62(3):1019–1043, 2001.

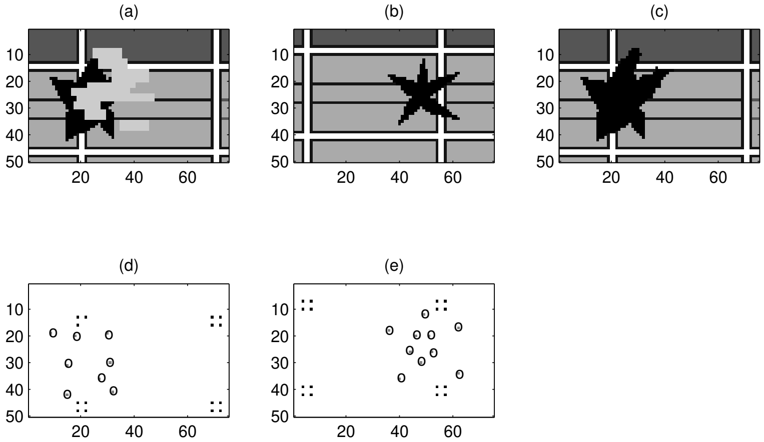

Figure 14: (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. The object is in different positions highly deformed and the background is also moved in a separate direction. Both the object and the background are successfully recovered.

[11] T. F. Chan and J. Shen. Morphology Invariant PDE Inpaintings. Technical report, UCLA Dept. of Math., CAM 01-15, 2001.

[12] T. F. Chan and J. Shen. Non-Texture Inpainting by Curvature-Driven Diffusion (CDD). *J. Visual Comm. Image Rep.*, 12(4):436–449, *to appear* 2001.

[13] L. Chanas, J.P. Cocquerez, and J. Blanc-Talon. Highly degraded sequences restoration and inpainting. *preprint*, 2001.

[14] J. H. Duncan and T.-C. Chou. On the Detection of Motion and the Computation of Optical Flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(3), 1992.

[15] 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.

[16] F. Guichard. A morphological, affine and galilean invariant scale-space for movies. *IEEE Trans. Image Process.*, 7(3):444–456, 1998.

[17] B.K.P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[18] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Detection of Missing Data in Image Sequences. *IEEE Trans. Image Process.*, 11(4):1496–1508, 1995.

[19] 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.

[20] J. Konrad and E. Dubois. Bayesian Estimation of Motion Vector Fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[21] R. Krishnamurthy, J. W. Woods, and P. Moulin. Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(5):713 –726, 1999.

[22] A. Kumar, A. R. Tannenbaum, and G.J. Balas. Optical flow: A curve evolution approach. *IEEE Trans. on Image Proc.*, 5(4):598–610, 1996.

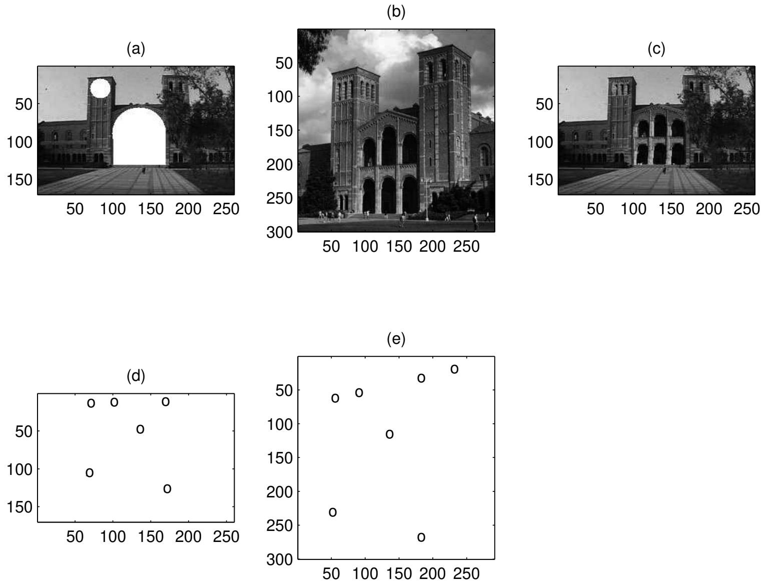

Figure 15: Application to a real image. (a) I_1 with white inpainting domain, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note that, I_1 and I_2 are taken in different time and from different viewpoints. However, the missing windows at the top left of the building are recovered and three poles are recovered.

[23] B.D. Lucas and T.Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joint Conf. on Art. Intell.*, 1981.

[24] M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Trans. on Image Processing*, 3(1):41–64, 1994.

[25] S. Masnou. Variational Formulation of the Disocclusion problem. *The abdus salam, int'l centre for theoretical physics,school on mathematical problems in image processing*, 2000.

[26] 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.

[27] N. Paragios and R. Deriche. Geodesic Active Regions for Motion Estimation and Tracking. *Proceedings of IEEE Int'l Conf. on Computer Vision*, 1:688–694, 1999.

[28] M. J. D. Powell. The Theory of Radial Basis Function Approximation. *Advances in Numerical Analysis*, II:105–210, 1990.

[29] M.J.D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. *Computational Techniques and Applications (CTAC95)*, 1995.

[30] S. M. Smith and J. M. Brady. ASSET-2: real-time motion segmentation and shape tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):814 –820, 1995.

[31] C. Tomasi and J. Shi. Image deformations are better than optical flow. *Mathematical and Computer Modeling*, 24(5/6):165–175, 1996.

[32] G. Wahba. Spline models for observational data. *SIAM*, 1990.

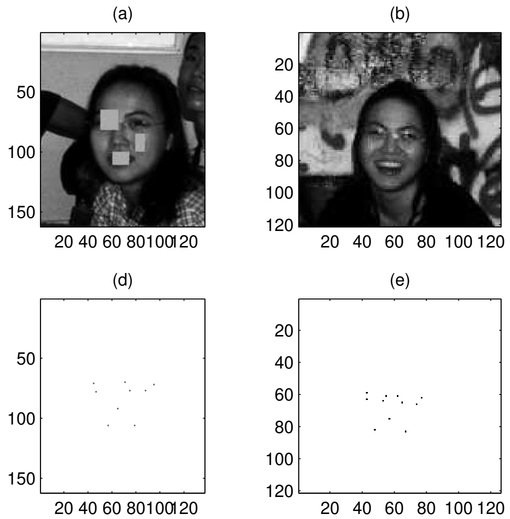

Figure 16: Application to a real image. (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note that, I_1 and I_2 are taken in different place and different time. However, the missing eye and the teeth are successfully recovered.

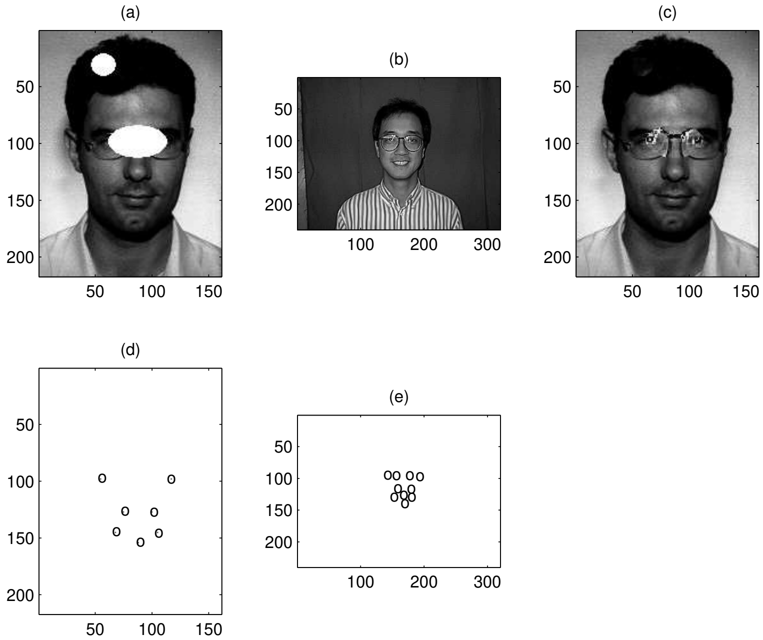

Figure 17: Application to a real image. (a) I_1, (b) I_2, (c) recovered result, and (d),(e) the landmarks from I_1 and I_2 respectively. Note, I_1 and I_2 are photos of two different people. However, information from I_2 can be used to recover I_1.

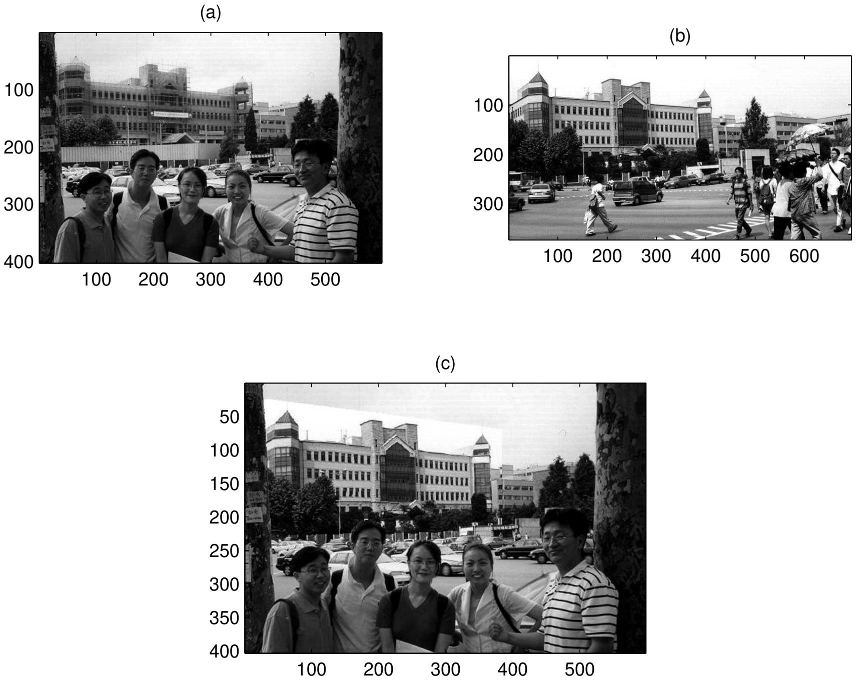

Figure 18: (a) I_1, (b) I_2, (c) recovered result. I_1 is from the summer of 1998 when the building was under construction, and I_2 was taken in the summer of 2001 when the construction is completed. By keeping the people from the old photo, only the building is successfully replaced by the new building. We used the whole building as inpainting domain in I_1.

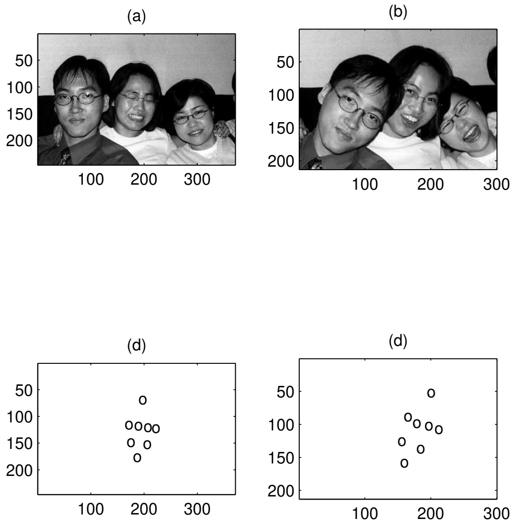

Figure 19: Some photos with odd facial expression can be modified. The middle person in I_1 (a), is modified using the other image with smiling face I_2 in (b). (c) shows the result which is modified from I_1 middle person is now smiling. We used the whole face as inpainting domain in I_1.

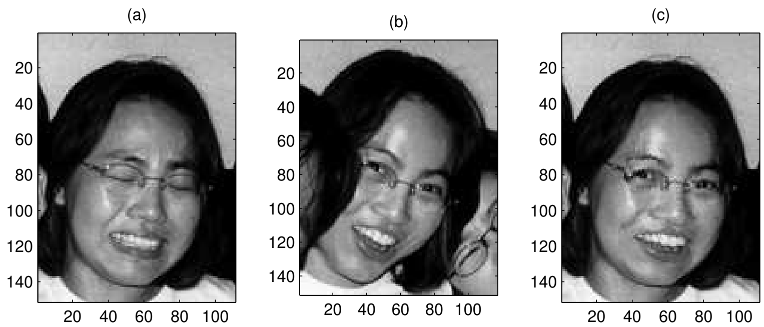

Figure 20: This is a close-up of Fig. 19, showing detailed recovered result. The mask refilled was the whole face. Comparing I_1(a) and (c) result, the eyes are opened and the mouth is modified according to I_2 in (b).