Noname manuscript No.
(will be inserted by the editor)

Globally Optimal Estimation of Nonrigid Image Distortion

Yuandong Tian and Srinivasa G. Narasimhan
The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, USA
{yuandong, srinivas}@cs.cmu.edu
http://www.cs.cmu.edu/~ILIM/projects/IM/globalopt/research_globalopt.html
Lab Website: **http://www.cs.cmu.edu/~ILIM/**

the date of receipt and acceptance should be inserted later

Abstract Image alignment in the presence of non-rigid distortions is a challenging task. Typically, this involves estimating the parameters of a dense deformation field that warps a distorted image back to its undistorted template. Generative approaches based on parameter optimization such as Lucas-Kanade can get trapped within local minima. On the other hand, discriminative approaches like nearest-neighbor require a large number of training samples that grows exponentially with respect to the dimension of the parameter space, and polynomially with the desired accuracy $1/\epsilon$. In this work, we develop a novel data-driven iterative algorithm that combines the best of both generative and discriminative approaches. For this, we introduce the notion of a "pull-back" operation that enables us to predict the parameters of the test image using training samples that are not in its neighborhood (not ϵ-close) in the parameter space. We prove that our algorithm converges to the global optimum using a significantly lower number of training samples that grows only *logarithmically* with the desired accuracy. We analyze the behavior of our algorithm extensively using synthetic data and demonstrate successful results on experiments with complex deformations due to water and clothing.

1 Introduction

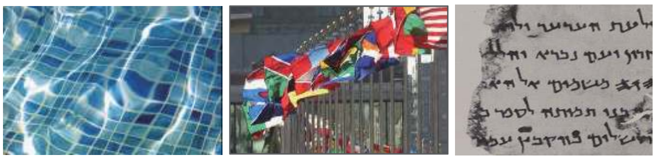


Fig. 1 Typical distortions caused by water fluctuation, non-rigid cloth deformation and optical scanning of old manuscripts. Recovering from different types of distortion is important for water surface shape estimation, 3D reconstruction of deforming cloth and digitization of ancient documents.

Images that capture non-rigid deformations of objects such as water, clothing and human bodies, exhibit complex distortions (Fig. 1). Aligning or registering such images despite the distortions is an important goal in computer vision that has implications for tracking and motion understanding, object recognition, OCR and medical image analysis. Typically, given a distorted image $I_{\mathbf{p}}$ (e.g., of a scene observed through an undulating water surface) and its template T (the scene observed when the water is still), the task is to estimate the parameters $\mathbf{p}$ of a distortion model that warps the image back to the template[1].

Most techniques for non-rigid image alignment can be classified into three broad categories, i.e., feature-based, generative and discriminative approaches. *Firstly*, feature matching techniques aim to match a set of sparse local features in the distorted image with those in the template [16,15,20]. Then, the parameters of a distortion model are estimated from the matchings. These methods work well when the dimension d of the parameter space is low (e.g., 6 for affine), but often fail in the presence of repetitive textures or high dimensional non-rigid distortions. *Secondly*, template matching techniques obtain dense correspondence between a dis-

Acknowledgements: This work was supported in parts by ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, an Okawa Research grant and an NSF CAREER Award IIS-0643628.

[1] Other works [34,14,6] use a set of distorted images or videos as the input and compute distortions and/or the template.

torted image and its template by minimizing a non-convex objective function $J(\tilde{\mathbf{p}}) = \|I_{\mathbf{p}} - I_{\tilde{\mathbf{p}}}\|^2$ using numerical methods [3] that often converge to local minima. *Thirdly*, discriminative approaches [7,1] learn a mapping M that directly predicts the distortion parameters $\mathbf{p}$ of a distorted image $I_{\mathbf{p}}$. As a classical example, the nearest-neighbor (NN) approach finds the neighbor closest to $I_{\mathbf{p}}$ and the neighbor's parameters are used as the prediction. Thus, with sufficient training samples it is possible to obtain the globally optimal solution. However, an enormous number of training samples $O((1/\epsilon)^d)$ are needed to achieve an accuracy of $1/\epsilon$ (i.e., $\|\tilde{\mathbf{p}} - \mathbf{p}\| \leq \epsilon$ for prediction $\tilde{\mathbf{p}}$ and true $\mathbf{p}$), resulting in inaccurate prediction for high-dimensional distortions. This phenomenon, known as the curse of dimensionality, remains even in more advanced machine learning techniques.

In this work, we develop a novel data-driven algorithm that combines the best of the generative and discriminative approaches for distortion estimation. Our algorithm adopts an iterative strategy that successively warps back the distorted test image towards the template until convergence. Unlike many previous works, we prove under mild conditions the algorithm always reduces the amount of distortion of a test image by a constant factor in each iteration, and thus converges to the global optimum. By the term *global optimum*, we mean it returns the global optimum solution $\tilde{\mathbf{p}}^* = \mathbf{p}$ of the minimization problem $J(\tilde{\mathbf{p}}) = \|I_{\mathbf{p}} - I_{\tilde{\mathbf{p}}}\|^2$, where $\|\cdot\|$ could be any norm in the image space. Furthermore, the number of training samples N, if distributed properly, is $O(C^d \log 1/\epsilon)$, which grows *logarithmically* with respect to the accuracy $1/\epsilon$ (Note C is independent of ϵ.). More importantly, the dimension d is decoupled from the accuracy $1/\epsilon$, breaking the curse of dimensionality.

The intuition behind this result is that two distorted images with very different distortion parameters still can share a large portion of the image content (albeit with different permutations of pixels) and can help each other in prediction. Using such training samples that are far away from the test sample enables our algorithm to achieve the same accuracy with much fewer samples compared to the nearest-neighbor case.

Our framework can be applied to a broad class of 2D image distortions including affine warps, and more complex spatially nonlinear distortion (e.g. water and cloth deformation). In particular, our framework does not require the warping family to form a group, hence has fewer restrictions than previous works [10,2,18,35] that use a similar "warp-back" strategy.

We have extensively analyzed the performance of our algorithm using synthetic experiments. Our theoretical analysis makes certain assumptions: **(a)** the form of the distortion model is known a priori, the mapping M from the distorted images to the parameters is one-to-one, and the training samples can be accurately generated from the template; **(b)** the occlusions caused by distortions (e.g. cloth folding) are negligible, **(c)** the artifacts of the imaging process such as aliasing, motion blur and defocus arising due to scene deformations are negligible. In practice, these restrictions are not severe — our algorithm is still able to demonstrate strong results on real experiments with complex deformations due to water fluctuation and cloth deformation, outperforming several existing methods [34,23]. In the future, we will explore broader applications such as face alignment, 3D registration of CT, markerless motion capture and pose estimation.

2 Related Work

There has been a long and rich history of studying geometric transformations between two images. To list them all is beyond the scope of this paper. In the following, we only discuss the works that are most relevant to our approach.

Generative Approaches. Starting from the classical optical flow algorithm by Lucas and Kanade [17], these approaches minimize the function $J(\tilde{\mathbf{p}}) = \|I_{\mathbf{p}} - I_{\tilde{\mathbf{p}}}\|^2$ with respect to the parameter $\tilde{\mathbf{p}}$. The intensity difference between the distorted template $I_{\tilde{\mathbf{p}}}$ under the current parameter estimate $\tilde{\mathbf{p}}$ and the test image $I_{\mathbf{p}}$, is iteratively minimized until it reaches a local minimum.

Under the same minimization framework, many successive works achieve faster convergence by using a constant Hessian matrix. As a trade-off, restrictions on the type of warping have to be placed. For example, the forward compositional approach [30] requires the warping to be compositional. The inverse additive method [10] requires the warping to be separable or spatially linear. Inverse compositional approaches [2,18,35] require the warping to be both compositional and invertible. These conditions restrict the possible applications of these methods. Other methods, including Active Appearance Models [5,18], Direct Appearance Models [12] and Difference Decomposition [9,27] are applicable to a wider class of distortions and are fast. However, it is not clear which function is minimized during iterations and there is no guarantee for convergence.

Free-form medical image registration [23] adopts a multilevel approach in which distortion parametrized by a B-spline is optimized to align two images at each level. The resulting estimated distortion is nonparametric and hence no predefined types of warping are required. But the algorithm may still be trapped within local optima. A Markov Random Field can also be used to model image deformation [28], but the underlying combinatorial problem is NP-hard and approximate inference techniques, such as linear programming relaxation or Tree-reweighted Message Passing, have to be used to obtain an locally optimal solution. Recently, to address the problem of local optima, a convex approximation to the objective function has been learned [19,36], but whether it remains faithful under large distortions is unclear.

Discriminative Approaches. This research direction starts from the idea of learning a direct mapping from the distorted image to the template, based on a training set with known distortion parameters. The simplest example is the nearest-neighbor approach, while more advanced approaches include Relevant Vector Regression [1], Gaussian Processes [38], Boosting [4], Mixture of Experts [32], or using multiple regressors chosen by the response of a gate classifier on the distorted images [21]. However, all of them require many samples to address the curse of dimensionality. Another way to address this problem is to find a low-dimensional representation (called "latent variables") of the parameter space, e.g. using PCA or GPLVM[29]. Then the prediction is made in the low-dimensional space.

Feature-based Approaches. The third research direction uses highly distinctive local features for sparse matching, e.g. SIFT [16]. Being rotation and scale invariant, such local features can be used to match images with large viewpoint changes, under analytic transformations such as affine or perspective, and with occlusions. Salzmann and Fua [24] also use such local features to find the point correspondences in the case of non-rigid deformation, but trustworthy local matches are sparse and spatial models have to be included to obtain denser correspondences [37,11].

Combining discriminative and generative approaches. Since both generative and discriminative approaches have their advantages and disadvantages, there have been many attempts to combine both. One popular strategy [31,26] is to first find a coarse estimation using the discriminative approach. Then, using this estimation as the initialization, a generative method is applied in the second stage for refinement. This requires that the first prediction be sufficiently close to the global optimum. Randomly generated training samples are also used in the iterative procedure, e.g. Hyperplane Approximation [13], which is similar in spirit to our approach. However, they use a spatially linear distortion model along with a linear estimator (hyperplane) that does not guarantee global optimality. Also they do not relate the distribution of random training samples to the convergence of the algorithm. In Rosales and Sclaroff [22], from candidate predictions made by multiple predictors, a generative model is used to choose the best one as the final output. However, none of the above approaches have the theoretical guarantees as in our work.

3 Distortion Model

We first describe the distortion model used in this work. Given a template image T and a d-dimensional vector of parameters $\mathbf{p}$, a distorted image $I_\mathbf{p}$ is computed using a *generating function* G:

$$I_\mathbf{p} = G(T, \mathbf{p}) \tag{1}$$

In particular, the template is at the origin of the parameter space, i.e., $T = I_0 = G(T, 0)$. We denote $\mathcal{I}$ as the manifold that consists of all possible distorted images that can be generated from Eqn. (1):

$$\mathcal{I} = \left\{ I_\mathbf{p} = G(T, \mathbf{p}) \mid \forall \mathbf{p} \in \mathbb{R}^d \right\} \tag{2}$$

The function G can be implemented using an image warp $W(\mathbf{x}, \mathbf{p})$ that maps a pixel $\mathbf{x}$ to the position $W(\mathbf{x}, \mathbf{p})$. Typically $W(\mathbf{x}, 0) = \mathbf{x}$. The warp $W(\mathbf{x}, \mathbf{p})$ can be applied to the template in either forward or backward direction:

$$G_\mathrm{F}(T, \mathbf{p}) : I_\mathbf{p}(W(\mathbf{x}, \mathbf{p})) = T(\mathbf{x}) \tag{3}$$

$$G_\mathrm{B}(T, \mathbf{p}) : I_\mathbf{p}(\mathbf{x}) = T(W(\mathbf{x}, \mathbf{p})) \tag{4}$$

Intuitively, the forward generating function pushes every pixel $\mathbf{x}$ in the template to the location $W(\mathbf{x}, \mathbf{p})$ in the distorted image, while the backward generating function pulls the pixel located at $W(\mathbf{x}, \mathbf{p})$ of the template back to the location $\mathbf{x}$ of the distorted image. A particular family of distortions may satisfy either Eqn. (3) or Eqn. (4), but not necessarily both. For invertible warpings, both representations are equally valid.

The main task of distortion estimation is to estimate the distortion parameters $\mathbf{p}$, if $I_\mathbf{p}$, T and G (or warping function W) are known. In this paper, we will focus on occlusion-free warps in the 2D image space and use a particular family of distortions as follows:

$$W(\mathbf{x}, \mathbf{p}) = \mathbf{x} + B(\mathbf{x})\mathbf{p} \tag{5}$$

where $B(\mathbf{x}) = [\mathbf{b}_1(\mathbf{x}), \ldots, \mathbf{b}_d(\mathbf{x})]$ is a set of warp bases that can be obtained a priori using either analytic models or measured data or complex physical simulations. Such bases $B(\mathbf{x})$ can capture spatially nonlinear distortions. As a result, this warping family covers a broad range of distortions, including affine transform, lens distortion, water distortion and changes of facial expressions [18]. Note that Eqn. (5) does not usually form a group. Our framework achieves global convergence for this broad family of distortions.

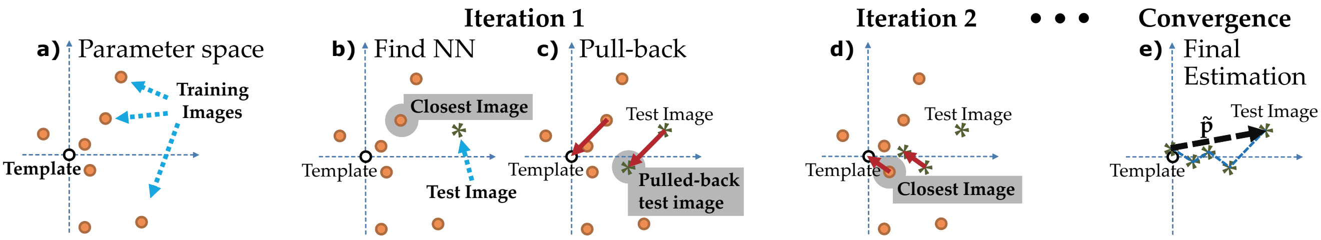


Fig. 2 Algorithm for distortion estimation. **(a)** The template (origin) T and distorted training images $\{I_{\rm tr}\}$ with known parameters $\{\mathbf{p}_{\rm tr}\}$ are shown in the parameter space. **(b)** Given a distorted test image, its nearest training image $(I_{\rm tr}, \mathbf{p}_{\rm tr})$ is found. **(c)** The test image is "pulled-back" using $\mathbf{p}_{\rm tr}$ to yield a new test image, which is closer to the template than the original one. **(d)** Step (b) and (c) are iterated, taking the test image closer and closer to the template. **(e)** The final estimate $\tilde{\mathbf{p}}$ is the summation of estimations in each iteration.

4 Iterative Algorithm for Distortion Estimation

In this section, we introduce the proposed algorithm for estimating parameters of the image distortion model.

4.1 The Intuition

Imagine a spaceship that wishes to return to the Earth. However, for some reason the navigation system is faulty and does not know the coordinates of the Earth relative to the current position. Fortunately, there are satellites around the Earth. Each satellite broadcasts a signal containing its coordinates, which can be received by the spaceship.

A straightforward way to localize the spaceship is to find the strongest signal from the closest satellite, and treat the received coordinates as its own. This is the well-known nearest-neighbor approach. The accuracy of such approaches heavily depends on how close the nearest satellite is to the spaceship, or, the local density of satellites.

However, a *fundamentally* different and more efficient way would be to drive the spaceship to another part of the space by the amount of displacement that *sends its nearest satellite back to the Earth*. If satellites are reasonably dense, then the spaceship should go closer to the earth. The spaceship can now receive new information at the new location, find the nearest satellite again and continue to move accordingly. With a proper distribution of satellites, the spaceship can land on the Earth. The original location of the spaceship can be estimated as the summation of all the consecutive readings of the coordinates.

Let us briefly analyze this approach. Obviously, this approach is beyond nearest-neighbor since it uses satellites that are far from each other, instead of just a nearby cluster. Hence, it requires only a sparse distribution of satellites around the original location of the spaceship, but a dense distribution near the Earth. That is, a coarse estimation suffices to bring the spaceship to the portion of the space with more satellites, where the estimation can be further refined. As a result, using fewer satellites can achieve the same accuracy as compared to the nearest-neighbor approach.

4.2 The Algorithm

We can do the same for images, by regarding the Earth as the template, the satellites as the training images (samples) and the spaceship as the distorted test image. As illustrated in Fig. 2, we start with the distorted test image I^0 and distorted training images $\{I_{\rm tr}\}$ with known parameters $\{\mathbf{p}_{\rm tr}\}$. In each iteration k the algorithm finds the closest training image $(I^k_{\rm tr}, \mathbf{p}^k_{\rm tr})$ to the distorted image I^k in terms of *image metric* and performs a "pull-back" operation H using $\mathbf{p}^k_{\rm tr}$ to obtain a new image I^{k+1}, that is *less distorted* compared to I^k and is *closer* to the template image T in the parameter space. Then, the training sample nearest to I^{k+1} is found, the parameter estimation is updated and the procedure is iterated until the desired accuracy $1/\epsilon$ is obtained, i.e., the estimation is ϵ-close to the template. Finally, the estimate of the distortion parameter $\mathbf{p}$ is given by the cumulative estimation $\tilde{\mathbf{p}}^K_{\rm tr}$. This algorithm is listed below.

Algorithm 1 The algorithm for distortion estimation

```
INPUT The training images $\{I^k_{\rm tr}\}$ with known parameters $\{\mathbf{p}^k_{\rm tr}\}$.
The test image $I^0$.
for $k = 0 : K$ do
    1. Find $I^k$'s nearest training image $I^k_{\rm tr}$ with known parameter $\mathbf{p}^k_{\rm tr}$
    i.e., $I^k_{\rm tr} = \arg\min_i \|I^k - I^i_{\rm tr}\|$.
    2. Set cumulative estimation $\tilde{\mathbf{p}}^k_{\rm tr} = \sum_{j=0}^k \mathbf{p}^j_{\rm tr}$.
    3. Set pulled-back test image $I^{k+1} = H(I^0, \tilde{\mathbf{p}}^k_{\rm tr})$.
        (a). For invertible warpings, $H$ is the inverse of the generating function $G$.
        (b). For non-invertible warpings, $H$ is the one opposite to $G$.
        E.g., $H = G_F$ if the generating function is $G_B$, and vice versa.
end for
OUTPUT $\tilde{\mathbf{p}}^K_{\rm tr}$ is the final estimation.
```

To alleviate the possible error accumulation with successive resampling (interpolation), we obtain I^k by pulling-back the original test image I^0 using the cumulative estimation $\tilde{\mathbf{p}}_{\mathrm{tr}}^{k-1} \equiv \sum_{j=0}^{k-1} \mathbf{p}_{\mathrm{tr}}^{k}$ in each iteration.

In the following, we will analyze the three key components of Alg. 1:

(1) How nearest-neighbor in the image space is related to the nearest-neighbor in the parameter space.
(2) The pull-back operation H.
(3) The distribution and the number of training samples required for the algorithm to converge globally.

We finally give a proof of convergence if the three components satisfy mild conditions. The idea of the proof is to show after each iteration the norm of residue always shrinks by a constant factor, and thus converges to zero. In other words, it is a coarse-to-fine strategy in the parameter space.

To keep the intuition clear, we start with the family of invertible warps. In this case, H is just the inverse operator of the generating function G. This operator partially cancels out the distortion in $I_{\mathbf{p}}$ by an amount of $\mathbf{q}$, yielding a new distorted image $I_{\mathbf{p}-\mathbf{q}}$ that remains on the distortion manifold $\mathcal{I}$ (Eqn. (2)). This substantially simplifies our analysis. Then we generalize the conclusion to non-invertible warps that take the form of Eqn. (5).

5 Global Optimality for Invertible Warping Case

In this section, we prove under the family of invertible warps, including specific kinds of warps that form a group, such as affine and projective transforms [9,35,2], that Alg. 1 converges to the global optimum if the mapping between the parameter space and the distortion manifold $\mathcal{I}$ is one-to-one, and the training samples are properly distributed. We also give an upper bound on the number of training samples as a sufficient condition to instantiate this distribution.

5.1 Nearest-Neighbor in the Image Space

Let us consider the set of distorted images whose distortion parameters $\mathbf{p}$ are within the sphere $S_{r_0} = \{I_{\mathbf{p}}, \|\mathbf{p}\| \le r_0\}$. The origin of this space corresponds to the undistorted template image T. Let M be the unknown *one-to-one* mapping function predicting the parameters $\mathbf{p}$ given the image $I_{\mathbf{p}}$:

$$M(I_{\mathbf{p}}) \equiv \mathbf{p} \tag{6}$$

Note M is only defined on $\mathcal{I}$ (Eqn. (2)) and is undefined on images which cannot be generated from the distortion model (Eqn. (5)). This is acceptable in the case of invertible warping, since the partially undistorted images always lie on $\mathcal{I}$.

Unlike the spaceship metaphor, we can no longer apply nearest-neighbor in the parameter (coordinate) space since the parameter of the test image is unknown. Instead, we find the nearest-neighbor according to an *image metric*, hoping that it will also give a close image in the parameter space. For this, we require the two metrics to be closely correlated, i.e., two images that are far or near in the parameter space have to be also far or near in the image space. Mathematically, this can be represented by the following Lipschitz continuity condition: there exist two universal constants $0 < L_1 \le L_2 < +\infty$ so that for two images I and I' within $\mathcal{I} \cap S_{r_0}$:

$$L_1\|I - I'\| \le \|M(I) - M(I')\| \le L_2\|I - I'\| \tag{7}$$

Without loss of generality, L_1 and L_2 are assumed to be the tightest bounds.

Note that $L_1 = 0$ is the case where two distinct images I and I' share the same parameters, and $L_2 = +\infty$ is the multi-valued mapping case in which a single image is associated with multiple parameters. In both cases, the one-to-one assumption is invalid and an infinite number of samples would be required to obtain an accurate estimation.

5.2 The distribution of training samples

Consider the distribution of the training images so that they are dense near the origin (template) and sparse at the periphery of the parameter space. Mathematically, given a distorted image $I \in \mathcal{I}$ generated from the distortion model with $\|M(I)\| \le r$, we assume that there always exists a training image $I_{\mathrm{tr}} \in \mathcal{I}$ so that:

$$\|I - I_{\mathrm{tr}}\| \le \beta r / L_2 \tag{8}$$

where $\beta < 1$. Eqn. (8) shows the density decays when moving away from the template to the peripheral of the parameter space (increasing r). With this condition, the following theorem shows Alg. 1 always yields a global optimum estimation for *any* test distorted images within S_{r_0}.

Theorem 1 (The global convergence of Alg. 1 in the invertible warping case.) *If Eqn. (7) and Eqn. (8) hold with $\beta < 1$, then Alg. 1 computes an estimated mapping function $M'_K(I) \equiv \tilde{\mathbf{p}}_{\mathrm{tr}}^{K} = \sum_{k=0}^{K} \mathbf{p}_{\mathrm{tr}}^{k}$ so that for $\|M(I)\| \le r_0$:*

$$\|M'_K(I) - M(I)\| \le \beta^{K+1} r_0 \tag{9}$$

where $1 - \beta$ is the rate of convergence.
In particular, $M'_K(I) \to M(I)$ if $K \to +\infty$.

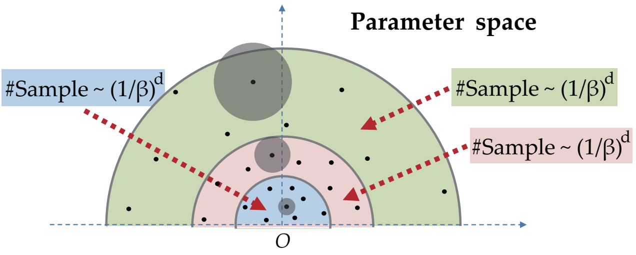


Fig. 3 The number of samples needed to fill a given sphere $\|\mathbf{p}\| \le r$ is independent of r since the allowed prediction uncertainty (shown in gray solid circle) is proportional to r. As a result, only a small neighborhood of the origin O (the template) requires dense sampling. This is the key to decouple the accuracy from the dimension of the parameter space, which is not attainable for the nearest-neighbor and regression-based approaches.

In contrast, in the nearest-neighbor case, the training images have to be distributed uniformly in the parameter space to achieve optimal performance for any test sample distributions.

Please see Appendix A for the detailed proof. The intuition of the proof is similar to the spaceship metaphor. With the density condition (Eqn. (8)) and the right-hand side of the Lipchitz condition (Eqn. (7)), it is guaranteed that in iteration k, the parameter difference between I^k and its nearest-neighbor is bounded by $\beta\|M(I^k)\|$. As a result, the norm of such difference goes down exponentially with k and the algorithm converges to the true distortion parameters.

5.3 The number of training samples

An interesting question is how many samples are needed to satisfy the density condition (Eqn. 8). We now show the number N of required training images grows only logarithmically with respect to the prediction accuracy $1/\epsilon$.

For this, we define the concept of ball-covering.

Definition 1 (Ball-Covering) A d-dimensional sphere $D_{r_1} = \{\|\mathbf{p}\| \le r_1\}$ of radius r_1 is said to be *covered* with a set of small spheres $\{D^i_{r_2}\}$ of radius $r_2 < r_1$, if for any $\mathbf{p} \in D_{r_1}$, there exists at least one small sphere $D^{i_0}_{r_2}$ so that $\mathbf{p} \in D^{i_0}_{r_2}$.

Given this definition, we have the following lemma:

Lemma 1 *To fill a d-dimensional sphere of radius r_1, $O((r_1/r_2)^d)$ small spheres of radius r_2 suffice.*

The proof is trivial. We now present a sufficient condition for Eqn. (8) to hold:

Lemma 2 *For a given radius r, if the sphere $\|I - T\| \le r_1 \equiv r/L_1$ in the image space can be covered by smaller spheres of radius $r_2 \equiv \beta r/L_2$, then Eqn. (8) holds.*

Proof If we could achieve this covering, then given any distorted image $I \in \mathcal{I}$ such that $\|M(I)\| \le r$, we would have

$$r \ge \|M(I)\| = \|M(I) - M(T)\| \ge L_1\|I - T\| \tag{10}$$

using the left-hand side of Eqn. (7) and $M(T) = 0$. Thus I satisfies $\|I - T\| \le r/L_1$ and by the definition of ball-covering, there *exists at least one* small sphere in the image space centered at I_{tr} so that $\|I - I_{\text{tr}}\| \le r_2 = \beta r/L_2$, which matches the condition of Eqn. (8). □

Now let us consider how many small spheres (or essentially, the training samples) are required for ball-covering in Lemma 2. Use Lemma 1, it turns out that the following number of samples suffices to satisfy the condition of Lemma 2:

$$O\left(\left(\frac{r_1}{r_2}\right)^d\right) = O\left(\left(\frac{L_2}{\beta L_1}\right)^d\right) \tag{11}$$

Crucially, this is independent of r (See Fig. 3). Thus, if Alg. 1 terminates in K iterations, $O(K(L_2/\beta L_1)^d)$ samples would suffice.

On the other hand, using Eqn. (9), we can compute $K = \lceil \log(r_0/\epsilon)/\log(1/\beta) \rceil - 1$ for a given accuracy $1/\epsilon$. As a result, the total number $N(\epsilon, \beta)$ of training images that is sufficient to make Alg. 1 converge to the true parameters (global optimum) is the following:

$$N(\epsilon, \beta) = O\left[\left(\frac{L_2}{\beta L_1}\right)^d \frac{\log r_0/\epsilon}{\log 1/\beta}\right] \tag{12}$$

A large β implies fewer training samples in each iteration but requires more iterations to achieve the same accuracy, and vice versa. The optimal β^*, which is independent of ϵ, can be obtained by minimizing Eqn. (12).

Note for any L_1 and L_2 that satisfy Eqn. (7), following the same reasoning, we conclude the number of training samples is bounded above by Eqn. (12). The tightest bound is given by largest L_1 and smallest L_2 satisfying Eqn. (7).

As a result, Eqn. (12) grows logarithmically with respect to the accuracy $1/\epsilon$. In contrast, with a similar analysis, nearest-neighbor requires $O((L_2/\epsilon L_1)^d)$ samples for the same accuracy. In Fig. 5(b), we show the significant differences in performance between the two methods on synthetic data. Intuitively, the existence of a generating function G substantially restricts the degree of freedom of its inverse mapping M. Thanks to this, we can establish M with good accuracy using significantly fewer samples.

6 Global Optimality for Non-Invertible Warping

So far, we have discussed the case where the warping $W(\mathbf{x}, \mathbf{p})$ is invertible. Under this assumption, each intermediate undistorted image I^k lies on the manifold $\mathcal{I}$. This greatly simplified our discussion. In the case of non-inverible warping, we can still arrive at the same global convergence conclusion with the same order of training samples. To achieve this, we need to address a central technical problem:

How to characterize the intermediate undistorted images I^k in Alg. 1 that no longer lie on the manifold $\mathcal{I}$.

Specifically, by "characterization", we mean the two following sub-problems:

- How to define the parameters of I^k?
- How far could I^k be from the manifold $\mathcal{I}$?

The first problem determines whether in each iteration, the estimated parameters remain reasonable, and the second determines whether the nearest-neighbor operations remain valid. In the following, we will show that the first sub-problem can be addressed by properly extending the domain of the inverse mapping M, and the second can be addressed by defining a generalized inverse operator as the pull-back operation H in Alg. 1.

6.1 An Extension of the Inverse Mapping M

We first show that an extension of the inverse mapping M to the entire image space that satisfies the bi-Lipchitz conditions (Eqn. (7)) is impossible.

Here is a proof by contradiction. Let us assume M is now defined everywhere satisfying Eqn. 7. Recall that $\mathcal{I}$ contains all the distorted images generated from G. We thus pick an image $I \notin \mathcal{I}$ but very close to the template image T (i.e. $\|I - T\| \le \eta$, for some small $\eta > 0$). Finding such image is easy since the dimension d of the manifold is typically much lower than the dimension of the entire image space. For example, one could swap two pixels in T to produce I.

Since M is defined in the entire image space, let $\mathbf{q} \equiv M(I)$. If $\|\mathbf{q}\| \le r_0$, then we have $I_\mathbf{q} \in \mathcal{I} \cap S_{r_0}$ and by Eqn. (7) we have:

$$L_1\|I - I_\mathbf{q}\| \le \|M(I) - M(I_\mathbf{q})\| = \|\mathbf{q} - \mathbf{q}\| = 0 \tag{13}$$

which implies $L_1 = 0$ since $I_\mathbf{q}$ is on the manifold but I is not. On the other hand, if $\|\mathbf{q}\| > r_0$, then by Eqn. (7) we have:

$$r_0 < \|\mathbf{q}\| = \|M(I) - M(T)\| \le L_2\|I - T\| \le L_2\eta \tag{14}$$

Since η could be arbitrarily small, $L_2 = +\infty$. Thus, for an image outside the manifold, both conditions in Eqn. (7) may not be satisfied simultaneously.

Fortunately, the following (weaker) extension of M is sufficient for proving a generalized version of the convergence theorem.

(a) The following bi-Lipchitz condition holds on the manifold $\mathcal{I} \cap S_{r_0}$. That is, for $I, I' \in \mathcal{I}$, we have:

$$L_1\|I - I'\| \le \|M(I) - M(I')\| \le L_2\|I - I'\| \tag{15}$$

(b) In the entire image space, the following (single-sided) Lipchitz condition holds. That is, for $I \notin \mathcal{I}$ or $I' \notin \mathcal{I}$, we have:

$$\|M(I) - M(I')\| \le L_2\|I - I'\| \tag{16}$$

So, outside $\mathcal{I} \cap S_{r_0}$, only the right-hand side of Eqn. (7) holds. Constructing this extension is easy. Note the only case that makes $L_2 = +\infty$ is that in the entire image space there exists $M(I) \ne M(I')$ for $I = I'$, or M is a multi-valued mapping. So any (single-valued) function M that is defined on the image space and satisfies $M(I_\mathbf{p}) = \mathbf{p}$ on the manifold $\mathcal{I}$ is a legitimate extension with a finite L_2.

The intuition behind is that in order to keep the number of training samples finite in the parameter space, it is required that the bi-Lipchitz conditions (Eqn. 15) hold on the manifold. However, outside the manifold what we need is just a continuity condition that bounds the distance between parameters using the distance between images.

6.2 The pull-back operator H for non-invertible warping

The warping family in the form of Eqn. (5) generally does not form a group and is not invertible. Thus it is impossible to find H that takes $\mathbf{q}$ as input and maps $I_\mathbf{p}$ to a less distorted image $H(I_\mathbf{p}, \mathbf{q})$ that is on $\mathcal{I}$.

However, if we allow $H(I_\mathbf{p}, \mathbf{q})$ to be outside $\mathcal{I}$, then there exists a simple construction of H so that the difference between $H(I_\mathbf{p}, \mathbf{q})$ and $I_{\mathbf{p}-\mathbf{q}}$ is bounded (see Appendix B for the construction of pull-back functions):

$$\|H(I_\mathbf{p}, \mathbf{q}) - I_{\mathbf{p}-\mathbf{q}}\| \le R\|\mathbf{p} - \mathbf{q}\| \tag{17}$$

where R is dependent on the maximum gradient of both the template and the bases, and is independent of $\mathbf{p}$ and $\mathbf{q}$. As the estimate $\mathbf{q}$ gets closer and closer to the true $\mathbf{p}$, $H(I_\mathbf{p}, \mathbf{q})$ indeed approaches $\mathcal{I}$ and concides with the template when $\mathbf{q} = \mathbf{p}$, as indicated in the right-hand side of Eqn. (17). In particular, $H(I_\mathbf{p}, \mathbf{p}) = T$. Thus we call H *generalized inverse*.

The mild requirement of generalized inverse enables Alg. 1 to deal with broader warping families than many previous works [35, 2, 10]. Please see more detailed discussion in Section 11.

The pull-back operation H that satisfied Eqn. (17) can be constructed as follows. For forward distortion (Eqn. (3)), we use backward warping with the same bases; for backward distortion (Eqn. (4)), we use forward warping with the same bases. For details, please see the Appendix B.

6.3 The Generalized Theorem for Convergence

With a proper extension of M (Eqn. (15) and Eqn. (16)) and the pull-back function H returning a less distorted image that is close enough to the manifold $\mathcal{I}$, we can prove the following generalized theorem for non-invertible warping:

Theorem 2 (The global convergence of Alg. 1 in the general case.) *If Eqn. (15), Eqn. (16), Eqn. (17) and Eqn. (8) hold and $\gamma \equiv 2RL_2 + \beta < 1$, then Alg. 1 computes an estimated mapping function $M'_K(I) \equiv \tilde{\mathbf{p}}^K_{\mathrm{tr}} = \sum_{k=0}^K \mathbf{p}^k_{\mathrm{tr}}$ so that for $\|M(I)\| \le r_0$:*

$$\|M'_K(I) - M(I)\| \le \gamma^{K+1} r_0 \tag{18}$$

where $1 - \gamma$ is the rate of convergence.
In particular, $M'_K(I) \to M(I)$ if $K \to +\infty$.

We verify that $\gamma < 1$ on synthetic data in Section 8.2.

The required number of training samples can be computed in a similar fashion as in the previous section, using the same ball-counting arguments. The only difference is that in the general case, since the rate of convergence is slower due to the additional factor $2RL_2$, more training samples are required:

$$N(\epsilon, R, L_1, L_2, \beta) = O\left[\left(\frac{L_2}{\beta L_1}\right)^d \frac{\log r_0/\epsilon}{\log 1/\gamma}\right] \tag{19}$$

However, compared to Eqn. (12), they are of the same order.

7 Possible Extensions to Algorithm 1

Using features. Instead of the raw image I, one can also use features $\phi(I)$ for nearest-neighbor search. In this situation, L_1 and L_2 are defined between the feature space and the parameter space:

$$L_1\|\phi(I) - \phi(I')\| \le \|M(I) - M(I')\| \le L_2\|\phi(I) - \phi(I')\| \tag{20}$$

With this definition, Theorem 1 and Theorem 2 still hold. A good image feature corresponds to a smaller ratio of L_2/L_1. This means that the feature metric is more correlated to the parameter metric. If they are perfectly correlated ($L_1 = L_2$), then fewest training samples are required.

Using generative approaches as the second stage. When the parameter estimation is very close to the true value, one could use a generative approach to save samples without being trapped into local optima. In such a case, Algorithm 1 can be regarded as a discriminative approach that gives a good initialization.

K_{NN} nearest-neighbors. In practice, due to the constant factor $(L_2/\beta L_1)^d$, the N given by Eqn. (19) can still be a large number. In this situation, using K_{NN} nearest-neighbors with weighted voting (i.e., kernel regression) can further reduce the required samples, as shown in Fig. 5(e).

Fast nearest-neighbors. For N training samples and K iterations, the time complexity of a naïve implementation of Algorithm 1 is $O(NK)$. Currently it takes 5 seconds for a rectification of 300 by 300 image with $N = 1000$ training samples and $K = 20$ iterations using our unoptimized Matlab codes on a Pentium Core 2 machine with a single core. However, many methods used in retrieving approximate nearest-neighbors, such as locality sensitive hashing (LSH), can be applied to reduce the complexity substantially.

Incorporating temporal knowledge. Although Algorithm 1 does not assume temporal relationship between two distorted images, when dealing with distorted video sequence, temporal continuity can be easily incorporated as follows: after the parameter $\tilde{\mathbf{p}}_t$ of the current frame I_t is estimated, we add a new *synthetic* training pair $(\tilde{\mathbf{p}}_t, I_{\tilde{\mathbf{p}}_t})$ to the training set and proceed with the next frame I_{t+1}. If $\tilde{\mathbf{p}}_t$ is an accurate estimation, then I_{t+1} is similar to $I_{\tilde{\mathbf{p}}_t}$ by temporal continuity and will be pulled-back directly near the origin (template) in just one step. If $\tilde{\mathbf{p}}_t$ is not accurate, adding a perfectly labeled training pair will not hurt the performance of the algorithm and does not cause drifting that often occurs in frame-to-frame tracking approaches.

Active training samples. It is possible to include new training images using the generating function G *after* the test image is known. The temporal continuity described above is an example. More generally, the parameters $\tilde{\mathbf{p}}$ estimated by any regression-based method (e.g., Relevant Vector Regression [1] or Gaussian Processes [38]), associated with the synthetic image $I_{\tilde{\mathbf{p}}}$ can be used as a training pair. Multiple regressors may also be used. Then, our algorithm simply selects the one closest to the test in the image metric. Note this is similar in spirit to [22] in which multiple regressors are

Fig. 4 Some template images used in synthetic experiments. (See Section 8)

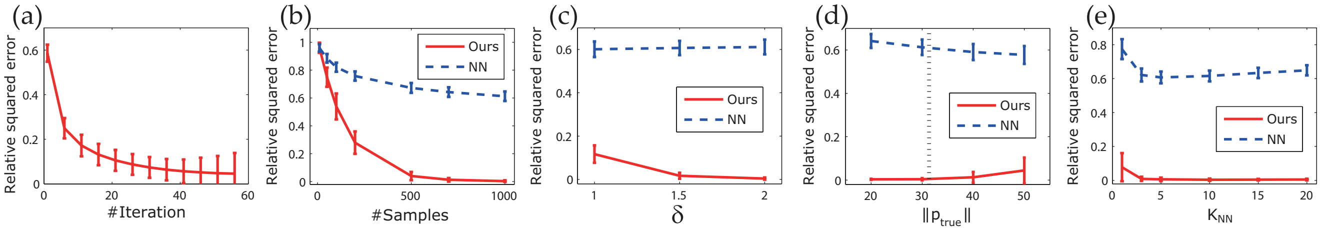


Fig. 5 The effects of four different factors on the performance of the algorithm in terms of relative squared error $\|\mathbf{p}_{\mathrm{true}} - \tilde{\mathbf{p}}\|_2^2 / \|\mathbf{p}_{\mathrm{true}}\|_2^2$. **(a)** Average convergence behavior computed over all test images. **(b)** The more training images, the better the performance. Note our method performs much better than nearest-neighbor given the same number of samples. **(c)** Estimation is more accurate if the training samples are more concentrated near the origin (template). **(d)** Performance drops when the test image is significantly more distorted than all the training images (The black dotted line shows the average magnitude of distortions $\|\mathbf{p}_{\mathrm{tr}}\|$ in the training images). **(e)** Using K_{NN}-nearest-neighbor with weighted voting reduces the number of training samples further.

used for candidate predictions which are then verified by a generative approach.

8 Analysis of the algorithm using simulations

8.1 Data synthesis

In order to verify the properties of our algorithm, we perform synthetic experiments where the true distortion parameters are known. We simulated distortions on 100 randomly selected images, some of which are shown in Fig. 4. The warps are of the form given by Eqn. (5), where $B(\mathbf{x})$ are composed of $d = 20$ orthonormal bases computed by applying PCA on randomly generated smooth deformation fields by Gaussian Processes. For each of the 100 template images, we synthesize $N = 1000$ distorted images for the training set and 10 for the test set. Note that a total of 1000 test samples are involved in the simulation and should be sufficient to justify our approach. Algorithm 1 is applied to each test image to obtain the relative (squared) error $e = \|\mathbf{p}_{\mathrm{true}} - \tilde{\mathbf{p}}\|_2^2 / \|\mathbf{p}_{\mathrm{true}}\|_2^2$.

In the following, we discuss how to generate the warping bases and training samples.

Generation of PCA bases. The Gaussian Processes used to generate deformation field has zero mean and covariance function $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2 / 2\sigma^2\right)$. $\mathbf{x}_1$ and $\mathbf{x}_2$ are locations of pixels and σ is a hyper-parameter that keeps the deformation smooth.

From the generated deformation field, we apply PCA and pick the first 20 eigenvectors as the deformation bases. The standard deviations of the 1-st and 20-th principle components are $s_1 = 11.63$ and $s_{20} = 7.95$ respectively. This shows that the energy is evenly distributed among 20 dimensions, and there is no degenerated dimension. We use the standard deviation in generation of training samples.

Generation of training samples. We follow Eqn. (8) in generating training images. Eqn. (8) says once the training images are distributed, the distance between a randomly picked image at radius r in the parameter space and its nearby training image should be proportional to r. Thus the density $m(r)$ of training samples, as a function of r, is proportional to $1/r^d$, where d is the dimension of the parameter space.

$m(r)$ only characterizes the distribution along the radial axis. The assumption (Eqn. (8)) is in a spherically symmetric form and thus we set the angular distribution of training samples to be spherically symmetric. Thus, the radial density $m_l(r)$ (the density function after marginalizing out all the angular components) is:

$$m_l(r) \propto m(r) \frac{\mathrm{dVol}_d(r)}{\mathrm{d}r} \propto \frac{1}{r} \tag{21}$$

where $\mathrm{Vol}_d(r)$ is the volume of d-dimensional sphere $\|\mathbf{p}\| \leq r$. As a sanity check, if Algorithm 1 returns the parameter with accuracy $1/\epsilon$, then along the radial axis, the training samples must be distributed along the interval $[\epsilon, r_0]$. By integrating $m_l(r)$ on this interval, we obtain:

$$\int_{\epsilon}^{r_0} m_l(r)\mathrm{d}r \propto \log r_0 - \log \epsilon = \log r_0/\epsilon \tag{22}$$

which is of the same order as Eqn. (12) (and Eqn. (19)). Finally, Fig. 6 shows the distribution $m_l(r)$.

From Eqn. (21) we thus obtain an algorithm for sampling training distributions. There are two practical issues. Firstly, in order to show how the shape of training distribution affects the performance, instead of directly sampling from the distribution $m_l(r)$ (Fig. 6), we first sample r from a uniform distribution and exponentiate r by the *shape parameter* δ. For $\delta > 1$, this will also yield a distribution peaked around the origin, and in particular when $\delta \to +\infty$ it will give exactly the $1/r$ fall-off. Secondly, instead of using a uniform r_0 for all PCA coefficients, using the standard deviation of each PCA basis will increase the sampling efficiency.

Algorithm 2 Sampling training images

```
INPUT The required accuracy $\epsilon$, the standard derivations $S = \mathrm{diag}(s_1, s_2, \ldots, s_d)$ of each PCA directions, the shape parameter $\delta$ and the number $N$ of training samples.
for $n = 1 : N$ do
    Draw sample $r$ from a uniform distribution on $[0, 1]$ and exponentiate $r$ by the shape parameter $\delta > 1$. A large $\delta$ yields peaked distribution around the origin.
    Uniformly sampling the angular coordinates by drawing $\mathbf{v}$ from multivariate normal dstribution $\mathbf{v} \sim \mathcal{N}(0, \mathbf{I})$ and normalize $\mathbf{v}$ so that $\|\mathbf{v}\| = 1$.
    The $n$-th training sample $\mathbf{p}_{\mathrm{tr}}^n = rS\mathbf{v}$.
end for
```

Note that sampling using Algorithm 2 will yield the distribution that matches the *outwards decaying shape* as indicated by Eqn. (8). However, a fairly large number of training samples have to be drawn to achieve the *density* requirement of Eqn. (8), i.e. $\beta < 1$. The actual number of training samples depends on the complexity of manifold $\mathcal{I}$, the ratio of L_2/L_1 and how effective the nearest-neighbor matching is. In this experiment, we use $N = 1000$ if not explicitly mentioned and the algorithm works well.

Fig. 5(a) shows the successful convergence of our algorithm averaged over all the test images. Fig. 7 shows example images warped with different magnitudes of distortion and the computed rectified images. Particularly, notice the significant improvement in the most distorted example. Fig. 8 illustrates an image distorted by a 60 degree rotation. Even if a coarse-to-fine strategy is used, gradient-descent methods like Lucas-Kanade can get stuck in a local minimum due to the seemingly large displacement in the rotation angle. However, our algorithm converges successfully to the correct parameters in just 3 to 4 iterations.

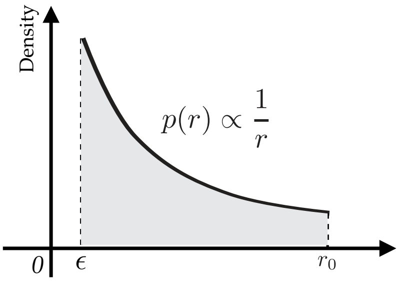
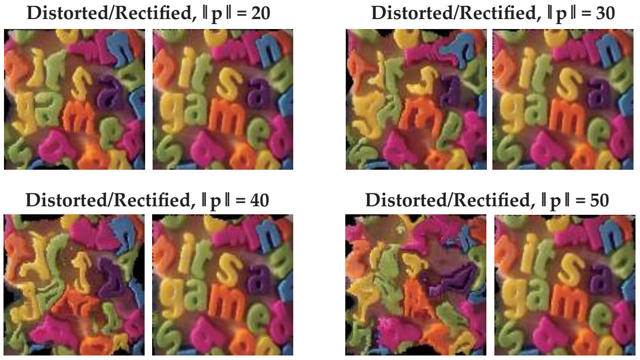


Fig. 6 The radial density distribution $m_l(r)$ of training samples. Sampling from $m_l(r)$ (Algorithm 2) will yield the distribution that has the same *shape* as Eqn. (8) (yet β could be larger than 1). On the other hand, β, or the *density* of the distribution, is determined by the number of samples drawn.


Distorted/Rectified, ‖p‖ = 20

Distorted/Rectified, ‖p‖ = 30

Distorted/Rectified, ‖p‖ = 40

Distorted/Rectified, ‖p‖ = 50

Fig. 7 Sample images distorted to various degrees and the recovered rectified images. The template is shown in Fig. 4

8.2 Factors that affect the algorithm

There are four major factors that affect the performance of the algorithm, including **(a)** the number N of training samples, **(b)** the number K_{NN} of nearest-neighbors involved in prediction, **(c)** the shape parameter δ of the distribution of training images, and **(d)** the magnitude of distortion $\|\mathbf{p}_{\mathrm{true}}\|$ of the test images.

We set the default values of the four factors to be $N = 1000$, $K_{\mathrm{NN}} = 10$, $\delta = 2$ and $\|\mathbf{p}_{\mathrm{true}}\| = 30$. Fig. 5(b)-(e) shows performance variations when perturbing one factor and keeping the others constant. Fig. 5(b) shows better performance is obtained with more training images. Although

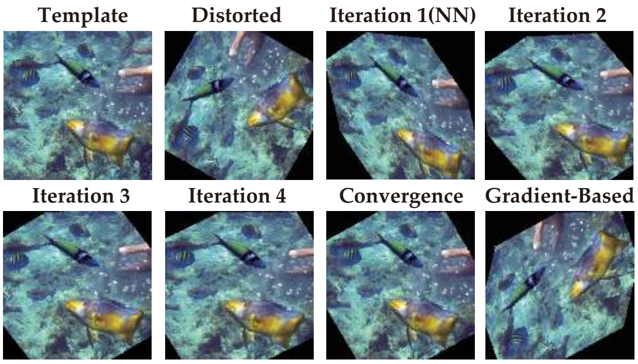


Fig. 8 Successful convergence of our algorithm for affine transformed image, given there is at least one training sample reaching that area. In contrast, gradient-descent methods (like Lucas-Kanade [3]) get stuck in local minima even with a coarse-to-fine strategy.

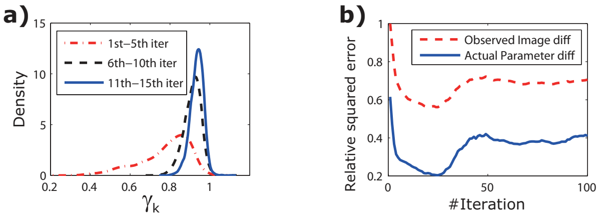


Fig. 9 **(a)** The empirical distribution of relative prediction error γ_k on test images in different iterations of the algorithm. 99.2% of the γ_k is small than 1, justifying $\gamma < 1$ in Theorem 2; others are due to insufficient samples. **(b)** The U-turn behavior in large distortion ($\|\mathbf{p}_{\text{true}}\| = 50$), when the resampling artifacts are severe.

Mild distortion ($\|\mathbf{p}\| = 20$)						
ρ_{P}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0646	0.0644	0.0668	0.0729	0.0863	0.1196
l_1-norm	0.0383	0.0419	0.0476	0.0599	0.0973	0.2440
Moderate distortion ($\|\mathbf{p}\| = 30$)						
ρ_{P}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0587	0.0607	0.0651	0.0751	0.0939	0.1427
l_1-norm	0.0363	0.0411	0.0481	0.0649	0.1195	0.2987
Large distortion ($\|\mathbf{p}\| = 40$)						
ρ_{P}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0595	0.0630	0.0703	0.0853	0.1164	0.1981
l_1-norm	0.0469	0.0508	0.0630	0.1009	0.2002	0.4207

Table 1 Relative squared errors of the estimated distortion of test images with salt & pepper noise. Note ρ_{P} is the percentage of contaminated pixels in the test image.

Mild distortion ($\|\mathbf{p}\| = 20$)						
ρ_{R}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0646	0.0686	0.0796	0.1202	0.2488	0.5146
l_1-norm	0.0383	0.0417	0.0486	0.0544	0.0858	0.6513
Moderate distortion ($\|\mathbf{p}\| = 30$)						
ρ_{R}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0587	0.0656	0.0825	0.1369	0.2292	0.4659
l_1-norm	0.0363	0.0431	0.0510	0.0772	0.1253	0.3055
Large distortion ($\|\mathbf{p}\| = 40$)						
ρ_{R}	No occ	10%	20%	30%	40%	50%
l_2-norm	0.0595	0.0729	0.1021	0.1624	0.3028	0.5437
l_1-norm	0.0469	0.0563	0.0821	0.1606	0.2937	1.2850

Table 2 Relative squared errors of the estimated distortion with rectangle occluded test images. Note ρ_{R} is the percentage of occluded pixels in the test image.

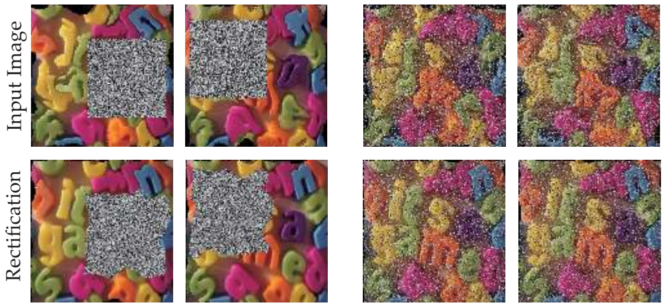
nearest-neighbor behaves similarly, its performance is much poorer for the same number of samples. Fig. 5(c) shows that a high accuracy is obtained if training samples are concentrated around the origin (larger δ) given the test image is within their range, as supported by the theoretical analysis. Conversely, the performance drops gradually if a test image is far away from the training set (Fig. 5(d)). Finally, Fig. 5(e) shows that given the same set of training samples, performance is better for K_{NN} nearest-neighbor with large K_{NN}. In other words, for the same performance, the parameter prediction using multiple neighbors requires fewer samples.

Verifying $\gamma < 1$ in Theorem 2. Fig. 9(a) shows how the distribution of relative prediction errors on the test images changes over iterations. The relative prediction error is defined as $\gamma_k \equiv \|\mathbf{p}^k_{\text{true}} - \tilde{\mathbf{p}}^k_{\text{tr}}\|/\|\mathbf{p}^k_{\text{true}}\|$, which corresponds to γ in our theoretical analysis in Theorem 2. For 99.2% of the simulated distortions, the number of samples (1000) we used are sufficient and $\gamma_k < 1$, indicating the algorithm's convergence. For the remaining 0.8%, the simulated distortions were too large and without sufficient training samples, hence $\gamma_k \geq 1$. The distributions of γ_k show that the rate of convergence slows down with increasing iterations. This is because more samples would be required around the origin to achieve a higher accuracy.

Performance under severe image resampling artifacts. Recall that resampling artifacts are not considered in our theoretical analysis. For large distortions where resampling artifacts can be overwhelming, our algorithm may not have the desired behavior. Interestingly, for many such cases, the observed difference between the rectified image and the template has the same shape as the actual distance between the true parameters and the estimated parameters (see Fig. 9(b)). Hence, we conjecture that the solution that produces minimum error in the image metric among many iterations will be a reasonable one, which is used as the stopping criterion in the real experiments.

8.3 Performance in the presence of noise and occlusion

We also check the usability of our method in the presence of noise and occlusion. In this experiment, we use the same 100 images as in Section 8.1. For each image, 1000 sam-



(a) Performance under occlusions (b) Performance under noise

Fig. 10 Distorted test images with noise/occlusion and their rectifications. **(a)** Distorted images with rectangle-shaped occlusion. **(b)** Distorted images with salt & pepper noise. Despite a large portion of the distorted image is contaminated, our algorithm still obtains a reasonable estimation of the distortion parameter and rectifies the image correctly. In all the results, we use the setting $\rho = 30\%$ and $\|\mathbf{p}\| = 30$. Note the algorithm is run on grayscale images and color is used here merely for illustration. (Best viewed in color)

ples are generated as training and 10 samples as testing. Each test image is contaminated with salt & pepper noise or rectangle-shaped occlusion before our algorithm is applied. To generate the salt & pepper noise, we randomly choose a portion ρ_{P} of pixels in the test image and set their values randomly (uniformly distributed in $[0, 1]$). In the case of rectangle-shaped occlusion, we choose a random position of a rectangle whose area is a portion ρ_{R} of the entire image, and fill in this rectangle with random noise that is uniformly distributed in $[0, 1]$. We use two pixel-wise image metrics, l_1 and l_2-norm on grayscale images, for nearest-neighbor.

Table 1 and Table 2 show our method is relatively robust to noise and occlusion in both cases. When the noise level is 10%-30%, our method still gives a reasonable estimation of distortion, with slightly increased squared prediction errors in the parameter space. Especially, l_1 metric performs better than l_2 metric in the rectangle-shaped occlusion case for occlusion rate up to 40%. Our method contrasts with many gradient-based approaches, in which a robust distance measure or a reweighting scheme has to be involved, and the initial parameters have to be carefully chosen.

9 Application I: Imaging through Water

The shapes of many deformable and time-varying interfaces between two media with different refraction indices, such as water surface, are very hard to measure directly. By perceiving the distortion of underwater scene, human vision can sense the fluctuation of the water surface qualitatively. In the following, we show that using Algorithm 1, we can estimate quantitatively the shape of the water surface, given both the appearance of the underwater scene when the water surface is still and a distorted image due to water fluctuation. This approach also works for a distorted video sequence by applying the same algorithm per frame. As a result, the shape of the water surface can be estimated over time.

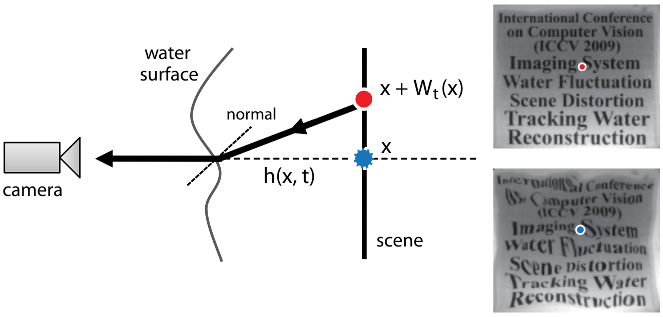


Fig. 11 Image formation in the presence of water distortion. The scene pixel at $\mathbf{x} + W_t(\mathbf{x})$ is perceived at location $\mathbf{x}$ in the distorted image.

9.1 Distortion Bases

Since the water distortion is caused by the bending normals of the water surface, its distortion bases can be obtained by physical simulation of water. According to Snell's law (Fig. 11), under first-order approximation, we can relate the distortion $W_t(\mathbf{x})$ to the height $h(\mathbf{x}, t)$ of the water surface at each time t:

$$W_t(\mathbf{x}) = \eta \nabla h(\mathbf{x}, t) \tag{23}$$

where η is a constant related to water height h_0 when the water surface is still, and relative refraction index between air and water. When the maximum surface fluctuation

$$\max_{\mathbf{x},t} |h(\mathbf{x}, t) - h_0| \tag{24}$$

is small compared to h_0, the water surface is governed by the following wave equation:

$$\frac{\partial^2 h(\mathbf{x}, t)}{\partial t^2} = c^2 \nabla^2 h(\mathbf{x}, t) \tag{25}$$

where $c = \sqrt{g h_0}$ is the velocity of wave (g is the gravity).

To simulate the wave equation, we use forward Euler method with a periodic boundary condition. This strategy is easy to implement and stable for small time step Δt:

$$h(\mathbf{x}, t+\Delta t) = 2h(\mathbf{x}, t) - h(\mathbf{x}, t-\Delta t) + c^2 \nabla^2 h(\mathbf{x}, t)(\Delta t)^2 \tag{26}$$

where $\nabla^2 h(\mathbf{x}, t)$ is the Laplacian operator on the water height image at time t. The initial conditions $h(\mathbf{x}, 0)$ and $h(\mathbf{x}, \Delta t)$

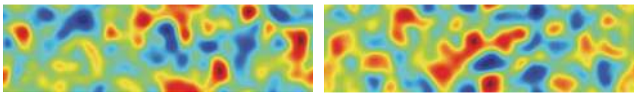


Fig. 12 Two samples of 2-D Gaussian processes used as the initial conditions of the wave simulator (Eqn. (25)).

are chosen to be a spatially correlated Gaussian Processes in a 2-D grid, as illustrated in Fig. 12. More specifically, $h(\mathbf{x}, 0)$ and $h(\mathbf{x}, \Delta t)$ are sampled from a multivariate Gaussian distribution $N(h_0\mathbf{1}, \Sigma)$ with each entry of the covariance $\Sigma_{\mathbf{x},\mathbf{x}'}$ inversely proportional to the spatial distance between $\mathbf{x}$ and $\mathbf{x}'$:

$$\Sigma_{\mathbf{x},\mathbf{x}'} = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2_{\text{synthesis}}}\right) \tag{27}$$

Note both the mean and variance of the Gaussian distribution are independent of the absolute coordinates of spatial locations. Thus the resulting initial condition is spatially stationary. $\sigma_{\text{synthesis}}$ is set by visually comparing the appearance of a known underwater planar scene at the bottom of the water tank with that from simulations. Importantly, $\sigma_{\text{synthesis}}$ is independent of the underlying scene. In the simulation, we set $c = 0.8$ pixel/frame and $\sigma_{\text{synthesis}} = 10$ pixels.

The simulator gives the time-evolving shape of the water surface. Since the initial condition is spatially stationary, and the wave equation is a time-invariant partial differential equation, we conclude that the evolving water surface is both temporally and spatially stationary. Thus, it suffices to capture the statistical properties on local patches. Based on this insight, we randomly sample space-time coordinates $(\mathbf{x}, t)$ and extract spatial patches (57×40) from W_t centered at $\mathbf{x}$. Then PCA is applied to these sampled patches to obtain the first 20 orthogonal principle modes $B(\mathbf{x}) = [\mathbf{b}_1(\mathbf{x}), \mathbf{b}_2(\mathbf{x}), \ldots, \mathbf{b}_{20}(\mathbf{x})]$ of water distortion, which we call *water bases* as shown in Fig. 13. The standard deviations of the 1-st and 20-th principle components are 610.08 and 42.82 respectively. By construction, the bases are translation invariant.

9.2 Experimental Setup

The water experiment consists of video camera observing vertically downward a 0.5m deep semi-transparent water tank with a planar scene at the bottom. The tank is illuminated from the side to avoid any surface reflections that are not modeled. The water surface is manually disturbed using a plastic ruler. The planar scene includes fonts of various sizes and natural textured underwater scene. The average dimension of distorted video sequences is around 350×250 with 500 frames. The variations of the dimension are due to a manual preprocessing step to trim the image boundaries corresponding to the water tank.

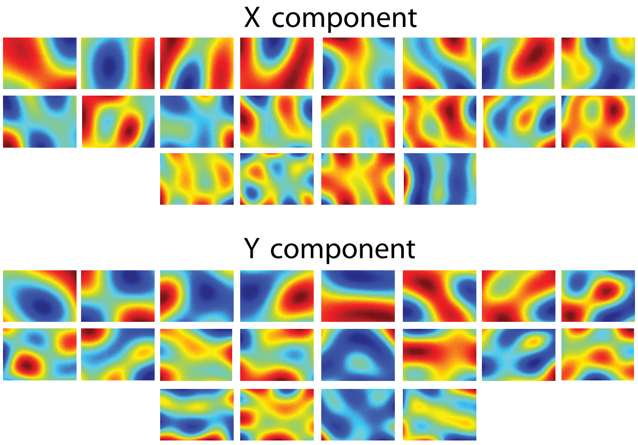


Fig. 13 The water bases $B(\mathbf{x}) = [\mathbf{b}_1(\mathbf{x}), \mathbf{b}_2(\mathbf{x}), \ldots, \mathbf{b}_{20}(\mathbf{x})]$. For both x and y components, the bases are sorted by their eigenvalues in a descending order, from left to right and from top to bottom.

We use the image taken under flat water surface as the template. Since the water distortion is local, we partition the image into overlapping patches and apply Algorithm 1 with the water bases (Fig. 13) on each patch to obtain a local deformation field. The image distance is computed using l_1 metric in grayscale after normalizing the pixel intensity into $[0, 1]$. 10000 training samples are synthesized from the template using the water bases, densely distributed around the original but sparsely elsewhere, as described in Section 8.1. For each distorted patch in the video sequence, 15 iterations are performed to obtain the parameter estimation on the water bases. Then these local deformation fields are stitched together, resulting in a global deformation field. At the overlapping regions between patches, we average the local deformation fields given by neighboring patches to obtain a smooth transition.

9.3 Results

Rectification of distorted images. We compared our algorithm to several previous representative techniques: free-form non-rigid image registration using B-splines [23], our previous work of water tracking [34] and a baseline approach in which we compute and match HOG (Histogram of Gradient) descriptors and interpolate the sparse correspondence using thin-plate interpolation to create a dense deformation

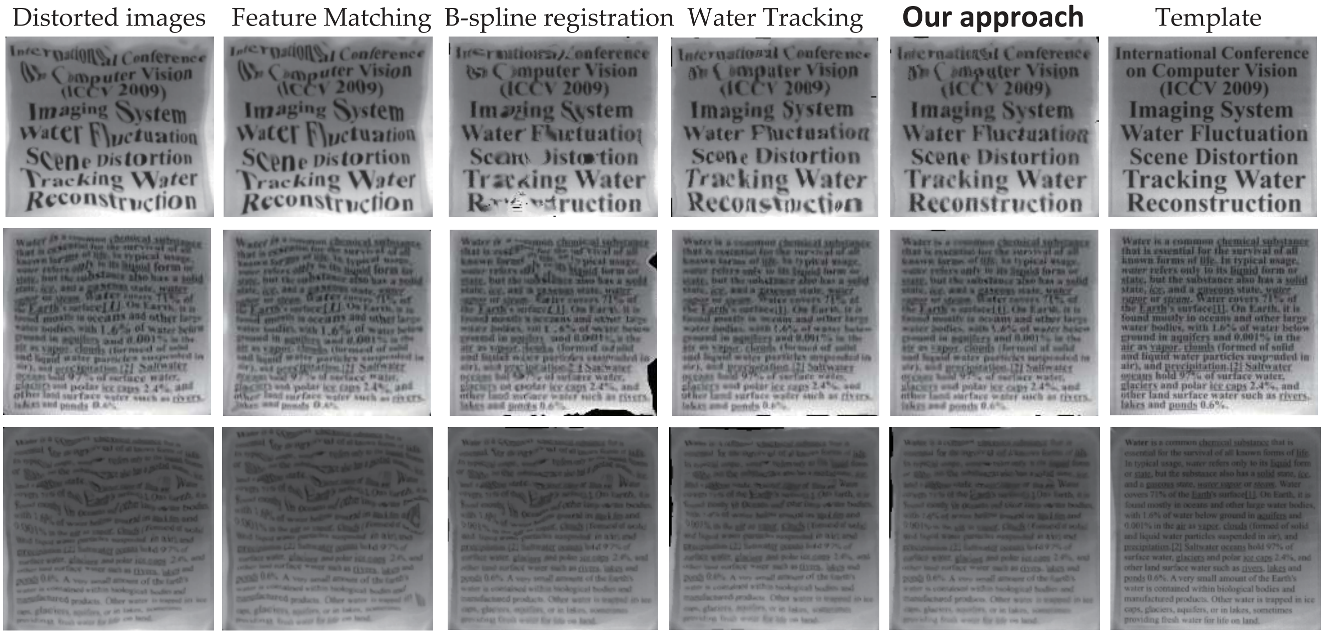


Fig. 14 Rectification of water distortion on text images of different font sizes (from the top row to the bottom row: MiddleFonts, SmallFonts and TinyFonts). Our approach outperforms HOG (Histogram of Gradient) feature matching, B-spline nonrigid registration [23] and yields slightly better results with water tracking [34]. However, water tracking relies on the entire video frames, while ours only needs two images.

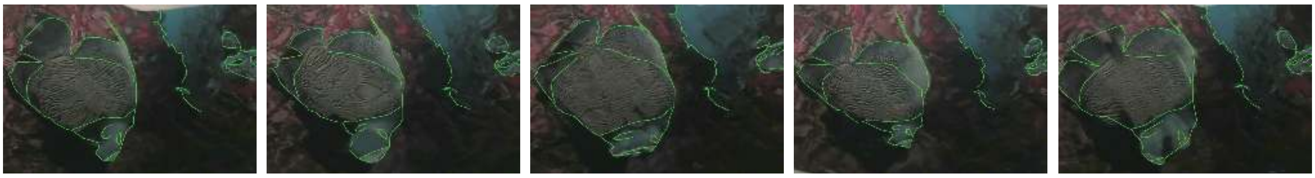


Fig. 15 Tracking a video sequence using estimated deformation fields. Although the underlying fish images are non-rigidly distorted, our method can still track it without drifting, using only grayscale images (We show color images for better illustration). Note the contour of the object in the first frame is manually labeled. See our website for the complete video sequence.

field. We also compare with the classic Lucas-Kanade method with the same set of water bases plus a coarse-to-fine strategy, as shown quantitatively in Section 9.4.

Fig. 14 shows the rectified images for a scene with text, and Fig. 25, 26 shows the results for a scene with colored textures. All the datasets, including three scenes with text (tinyFont, middleFont and smallFont) and scenes with textures, can be downloaded in our website. Since only sparse correspondences between two images are used, feature matching gives an inaccurate interpolated deformation field and fails to align details well. Nonrigid B-spline image registration [23] works better but fails occasionally on some image regions due to local minima. Our previous method, water tracking [34] produces better results than feature matching and B-spline registration. Yet it requires a short video sequence (61 frames) to rectify a single frame. In contrast, our method yields the best rectification results given only the template and one distorted image at a time.

Video tracking. Using the estimated distortion, one can find the corresponding points of an object's contour at each video frame, which gives the tracking result as shown in Fig. 15. We can see that although the shape of the fish undergoes large nonrigid distortions, our method still succeeded in tracking its contour reliably (note the first contour is manually labeled).

Water surface reconstruction. According to Eqn. (23), the deformation fields are proportional to the gradient of the water height at any time. Hence, one can recover the height of the water surface at each time using Frankot-Chellappa integration [8] on dense deformation fields of x and y directions. Some sample reconstructions are shown in Fig. 16.

Please check more video results on our website.

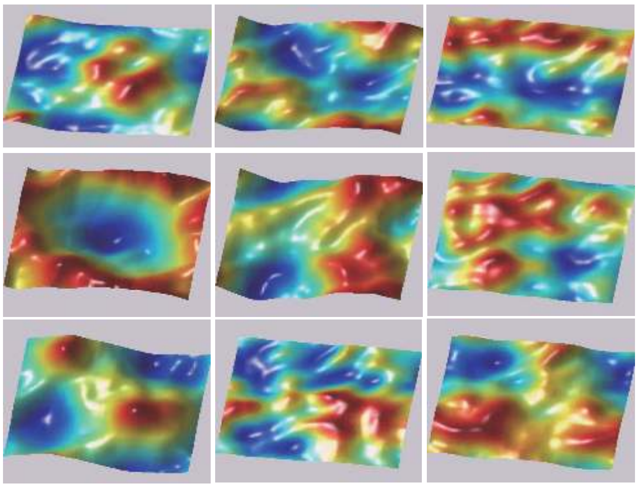


Fig. 16 Reconstructed water surfaces (dataset: SmallFonts) by spatially integrating the water distortion (Best viewed in color).

9.4 Quantitative Evaluation

In addition to visual comparisons, we also do quantitative comparisons to further verify our approach.

9.4.1 Reprojection error on images

Without groundtruth deformation fields, a convenient evaluation is to check whether the rectified frames concide well with the template, which is the image reprojection error. We compare our method with B-spline registration [23] and Lucas-Kanade registration using the same water bases (Fig. 13) and a coarse-to-fine strategy to avoid possible local minima. To measure the distance between a rectified image I and the template T, we compute the root-mean-square reprojection error $RMS_{\text{intensity}}$ as follows:

$$RMS_{\text{intensity}} = \sqrt{\frac{1}{n}\sum_{\mathbf{x}}(I(\mathbf{x}) - T(\mathbf{x}))^2} \tag{28}$$

where n is the number of pixels in each image. Note the image intensity is normalized into $[0, 1]$ before different algorithms are formally applied. For a video sequence, we compute RMS for each frame and take the mean value over time. Table 3 shows the result. We can see even with the same bases, Lucas-Kanade still gets trapped into the local minima and fails to give a low reprojection error. B-spline works better yet our method performs the best.

Dataset	Distorted video	Lucas-Kanade	B-spline [23]	Our method
TinyFonts	0.0720	0.0618	0.0553	**0.0444**
SmallFonts	0.1029	0.0624	0.0512	**0.0461**
MiddleFonts	0.1551	0.1092	0.0640	**0.0597**
Fish	0.0995	0.0831	0.0584	**0.0527**

Table 3 Comparison of the image reprojection error on different methods. All the errors are computed using RMS (See Eqn. (28)) and the mean RMS over the entire video sequence (500 frames) is shown in the table. Note the pixel intensity is normalized into $[0, 1]$ before different algorithms are applied. Thus the maximal possible reprojection error is 1 (black versus white images).

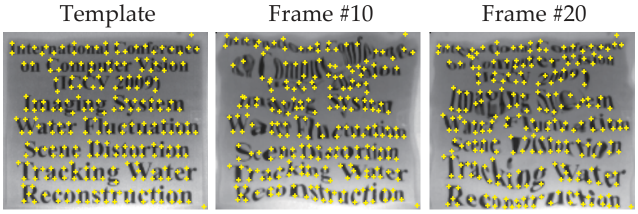


Fig. 17 Samples of landmark-labeled frames in dataset MiddleFonts. Note the video frames and the template are 253 by 293. The first 30 frames are manually labeled, each with 232 landmarks.

9.4.2 Reprojection error on landmarks

The image reprojection error is not a perfect performance measure; a distortion estimation algorithm may result in lower errors by arbitrarily rearranging the pixels without considering the spatial smoothness constraints. To further verify our method, we manually label $m = 232$ landmarks on the first 30 frames of one of the underwater dataset, MiddleFonts (See Fig. 17 for sample labels), and compute root-mean-square error RMS_{spatial} between the landmark positions $\{\mathbf{x}_i^{\text{t}}\}$ transformed from the template to the distorted frame using the estimated deformation field, and the landmark positions $\{\mathbf{x}_i^{\text{d}}\}$ that are labeled on the distorted frame:

$$RMS_{\text{spatial}} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{x}_i^{\text{t}} - \mathbf{x}_i^{\text{d}}\|_2^2} \tag{29}$$

Similarly, we compute mean RMS over 30 labeled distorted frames. Table 4 shows the results. We can see our method gives the smallest errors (measured in pixel), while other generative approaches, such as Lucas-Kanade (with the same set of bases) and B-spline, gives at least 60% higher errors. Since the landmark correspondence is sparse, we also test the performance of feature matching using HOG descriptor. To minimize the matching ambiguity and using the prior knowledge that the landmark positions are fluctuated around

	Distorted video	Lucas-Kanade	Feature matching	B-spline [23]	Our method
mean RMS	6.3404	5.2040	3.9282	3.8212	**2.5142**

Table 4 Comparison of the landmarks reprojection error on different methods. All the errors are computed using Eqn. (29) and in the table the mean error over the 30 labeled video frames of the Middle-Font dataset is shown. See Section 9.4.2 for detailed descriptions of each listed method.

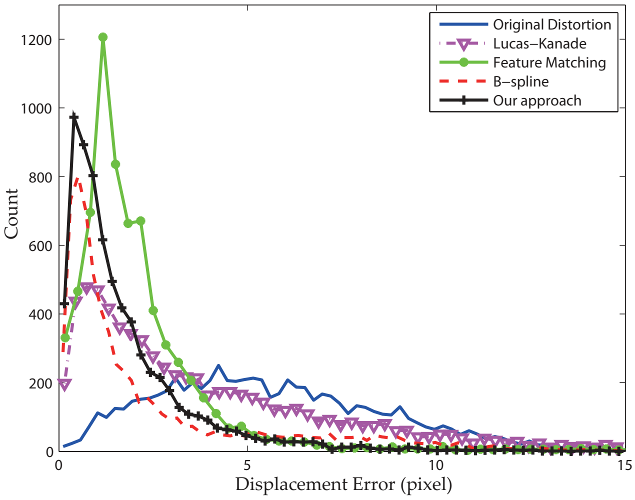


Fig. 18 Histograms of landmark displacement errors using different methods over 30 labeled frames, each with 232 landmarks. The displacements in the distorted images (blue solid line) follow a flat and Gaussian-like distribution. All the methods aim to push the distribution towards the origin. The Lucas-Kanade method (magenta line with triangle) produces a error distribution with a heavy tail, indicating that it often converges to local optima and many landmarks fail to align well. Local dense feature matching (green line with circle) works better, but the local ambiguity of HOG features leads to inaccuracy in the alignment, as indicated by the sharp peak of the distribution located at a region of positive errors. B-spline registration [23] (dashed red line) works even better using a more powerful optimization technique (BFGS) but still not as good as our method (black line with cross) whose error distribution is more concentrated near the origin and with a thinner tail.

their positions in the template, we match each HOG descriptor located at $\mathbf{x}$ in the template with all the *densely* extracted descriptors located in the vicinity of 11 pixels in the distorted frame, and pick the best one as the matching result. This approach yields better results than Lucas-Kanade and comparable to B-spline, yet is still not as good as our approach. Finally, Fig. 18 gives a more detailed analysis of the error distribution of different methods.

10 Application II: Cloth Deformation

Another interesting application of Algorithm 1 is to estimate nonrigid cloth deformation. Given a video sequence with deforming cloth, the goal is to estimate a dense and time-varying deformation field between different frames, which can be used for video tracking and 3D reconstruction.

10.1 Global motion and local deformation

In general, since cloth deformation behaves more globally than water distortion, we use the following two-stage approach. In the first stage, we downsample the original video (720 × 480) by a factor of 2, apply local affine bases of size 200 × 200 and estimate its 6 parameters using our method. This gives a coarsely undistorted video sequence. In the second stage, we apply local random bases (100 × 100) with 40 dimensions to the undistorted sequence, and obtain the final distortion estimation by distortion composition. We build our own dataset acquired by manually perturbing a piece of silk cloth with repetitive heart patterns.

In addition, we apply our method on the dataset offered by the authors of [33] to obtain the dense deformation field, which is used to reconstruct the 3D shape of the cloth. For their datasets, we use a slightly different approach. We begin by first using local trackers (mentioned below) to track reliable interest points over time and manually pick the correct trackers to obtain a *coarse* dense deformation field with thin-plate interpolation. Then local random bases are again applied on the coarsely rectified video sequences for refined estimation.

The local tracker. The local tracker we used is also based on Algorithm 1. Given an interest point in the template (usually is the first frame of the deforming cloth sequence), a local patch around it is cropped and 200 samples are generated using affine warps. During tracking, we initialize the position of the tracker as its position in the previous frame and extract the patch around it, on which Algorithm 1 applies to obtain the local deformation field that gives the position of the tracker in the current frame. With an illumination-invariant metric, this local tracker is robust to the shading effects in the cloth video sequence. As a result, many of the tracking trajectories are reliable and useful throughout the video sequence. By manually picking the good ones, a coarse yet representative deformation field can be built.

10.2 Results

Fig. 20 shows some sample frames of a rectified video sequence produced by our method on a piece of cloth with

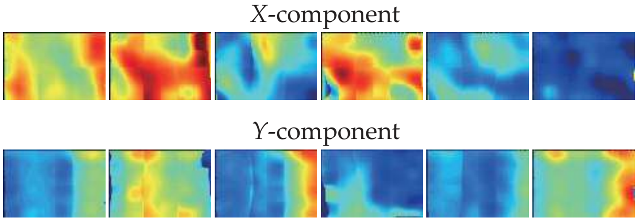


Fig. 19 Estimated deformation fields for the cloth sequence with repetitive heart patterns. The first row shows the x-component while the second row shows the y-component. The linear part in distortion fields is the affine component, while the nonlinear part is the nonrigid component. (Best viewed in color)

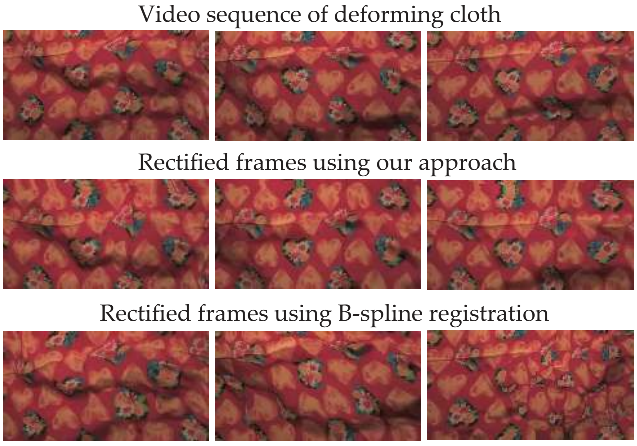

Video sequence of deforming cloth
Rectified frames using our approach
Rectified frames using B-spline registration

Fig. 20 Rectification of cloth deformation using different methods. The first row shows the original video frames, the second row shows the rectified video frames by our approach, and the last row shows the rectification by B-spline registration [23]. As a generative approach, B-spline registration converges to local minima; while our approach gives good distortion estimation and rectifies the deformation correctly.

repetitive heart patterns. B-spline registration [23], as a generative approach, goes into local minima in multiple frames, while our approach does not. Please watch the entire video sequence on our website for a more thorough comparison. Fig. 19 shows the estimated deformation fields. The affine components are shown as the linear part of the deformation fields, while the nonrigid components are shown as the nonlinear part, as clearly illustrated in this figure.

Fig. 21 shows the established correspondence on the dataset from [33]. Our method captures the wavy structure on the cloth in the first dataset and the bending structure in the second dataset throughout the video sequence. The 3D reconstruction of the dataset can be found in [33].

11 Conceptual comparisons with previous methods

As mentioned before, our method is conceptually different from many existing methods. In the following, we describe this difference in a case-by-case study. To make the comparison and illustrations clear, we assume one-dimensional parameter space. In such a case, all distorted images generated from the distortion model form a one-dimensional manifold $\mathcal{I}$ (Eqn. (2)), shown as a curve in the image space (Fig. 22(a)). The template ($\mathbf{p} = 0$), the training samples and the distorted test image $I_{\mathbf{p}}$ are identified as points on the curve.

Generative/discrimative approaches. Fig. 22 shows the fundamental difference between our approach and generative and discriminative approaches in the image space. Generative approaches initialized from the template ($\tilde{\mathbf{p}} = 0$) converge to local optimum due to the complicated nonlinear structure of the manifold $\mathcal{I}$, as shown in Fig. 22(b). On the other hand, discriminative approaches can get the global optimum given the condition that the training samples densely cover the manifold $\mathcal{I}$, as shown in Fig. 22(c). This may not be a big deal if the manifold is one-dimensional, but will require enormous number of training samples in the high-dimensional case. Our approach achieves the same accuracy with an iterative strategy and much fewer training samples distributed in a radially decreasing way. The samples, especially those close to the origin, are heavily reused. While the maximum distance of two nearby training samples has to be $O(\epsilon)$-close in the discriminative case, the maximum distance between two training samples in our approach is only required to be smaller than the "gap" of the curve and *independent* of the prediction accuracy. The gap is implicitly encoded in the two universal constants L_1 and L_2 in Eqn. (7).

Combining generative and discriminative approaches. Fig. 23(c)-(d) shows the difference between our method and previous methods combining the two approaches. Fig. 23(c) shows the heuristic that uses the discriminative approach as the initialization of the generative approach still leads to local minima, while our approach converges to the global optimum with the same distribution of training samples, as shown in Fig. 23(d). Although we do not guarantee global convergence with too few training samples, our approach fails only if the nearest-neighbor estimation is globally wrong, for example, predicting large negative values when the true parameter is large positive in 1-D case. In contrast, the way that the previous methods combine both approaches, as a generative approach by nature, is more sensitive to the local bumpy structures of the manifold $\mathcal{I}$.

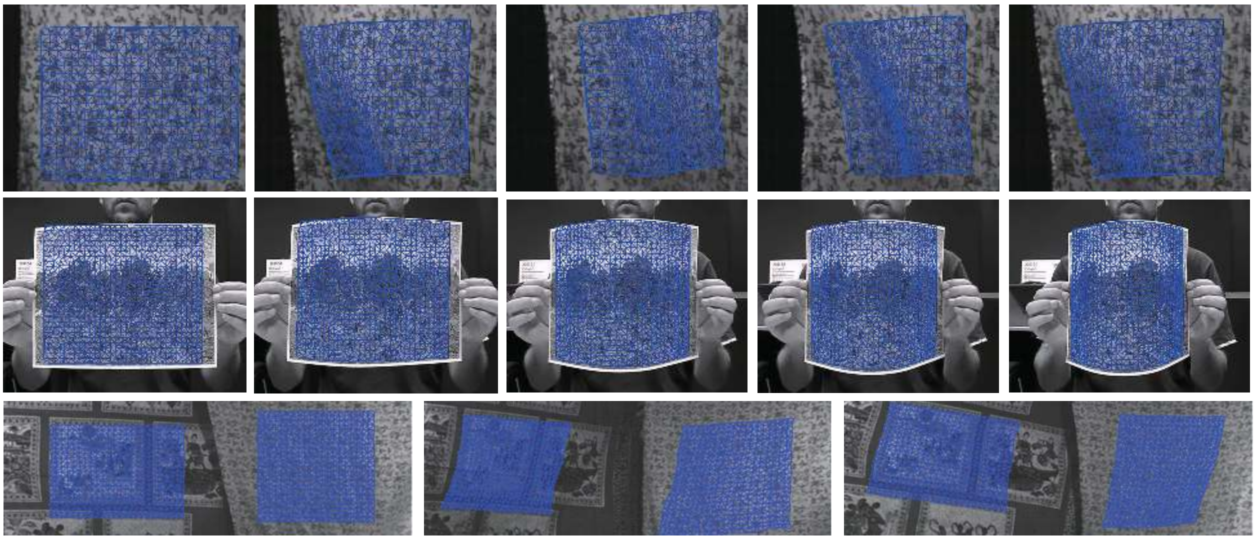


Fig. 21 Estimated 2D mesh on the video sequence of deforming cloth using our approach. The dataset in the first and the last row come from [33], while the dataset in the middle row comes from [25]. (Best viewed in color)

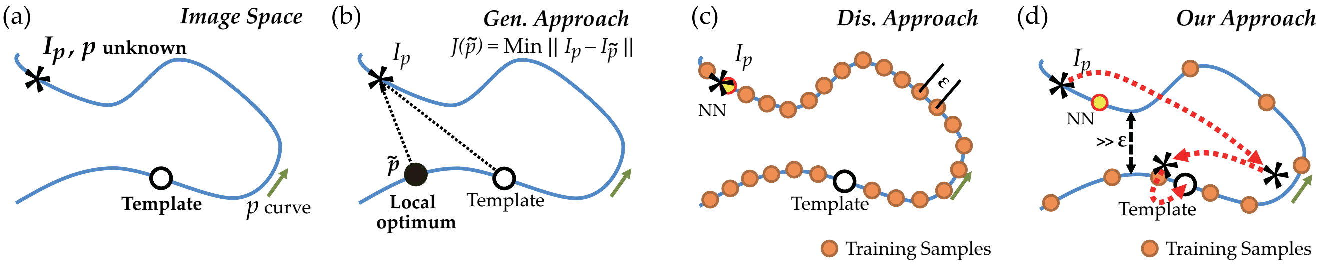


Fig. 22 Comparison with generative/discriminative approaches, illustrated in the image space. **(a)** The image space. The curve parameterized by $\mathbf{p}$ is the set of all the distorted images $\mathcal{I}$ generated from the distortion model (Eqn. (2)), assuming one-dimensional parameter space. **(b)** Generative approaches initialized at the template ($\tilde{\mathbf{p}} = 0$) converge to the local optimum. **(c)** Discriminative approaches obtain an ϵ-accurate estimation, if the training samples densely cover the curve. **(d)** With much fewer samples than the discriminative approaches, our approach obtains the same accuracy by iteratively refining the parameter estimation, as illustrated by the dashed red arrows.

Using warp-back strategies. Fig. 23(a)-(b) shows the fundamental difference between our approach and previous methods [35, 2, 10] with a similar strategy of successively warping-back. An energy minimization framework is commonly used in those methods. The standard gradient descent approach yields a trajectory of less distorted images until it reaches the template. By the formulation, the following two conditions have to be met: **(a)** the warp-back operations are in the warping family; **(b)** all the images on the trajectory have to be on the manifold $\mathcal{I}$, which is the set of all distorted images generated from the distortion model. This is only possible if the warping family forms a group.

For non-invertible distortion, if one condition is met then the other is broken. This is the reason why previous methods cannot handle non-invertible distortion as shown in Fig. 23(a). However, our method can handle it by properly relaxing the condition (b) so that **(1)** the trajectory of less distorted images is allowed to be *off* the manifold yet **(2)** the trajectory converges to the manifold $\mathcal{I}$ when the parameter estimation is close to the true value, and is guaranteed to hit the template if the parameter estimation is perfect, as shown in Fig. 23(b).

Sample distribution. The convergence property of our algorithm is *independent* of the location of the test samples within the sphere $\|\mathbf{p}\| \leq r_0$, if the training samples are distributed as explained in Section 5.2. In other words, we attain the guarantee of the *worst-case* performance. This differs from many previous methods that only work for a given prior distribution. Furthermore, if the distribution of the parameters of real-world deformations of an object is known a priori, then it can be combined with our sampling strategy to reduce the number of training samples even further.

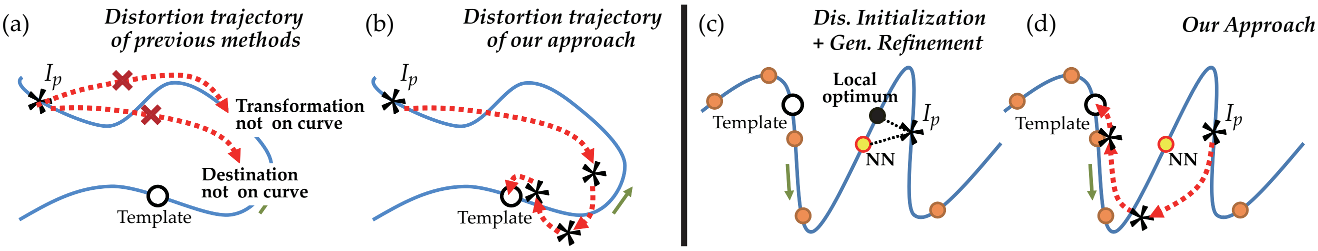

(a)
Distortion trajectory of previous methods
I_p
Transformation not on curve
Destination not on curve
Template
(b)
Distortion trajectory of our approach
I_p
Template
(c)
Dis. Initialization + Gen. Refinement
Local optimum
Template
I_p
NN
(d)
Our Approach
Template
I_p
NN

Fig. 23 **Left:** Comparison with previous works [35,2,10] that also use warp-back strategy, illustrated in the image space. **(a)** Previous methods use a restricted formulation that requires both the intermediate distorted images *on* the curve and the warp-back distortions *in* the warping family, which is only possible for warping families that form a group. **(b)** Our approach allows the undistorted image *off the curve* during iterations and still achieves global convergence. **Right:** Comparison with other methods that combine the generative and discriminative approaches. **(c)** Using the discriminative approach to initialize the generative approach [26,31] still leads to local convergence due to the local irregularity of the curve. **(d)** Using the same training set, our method converges to the global optimum.

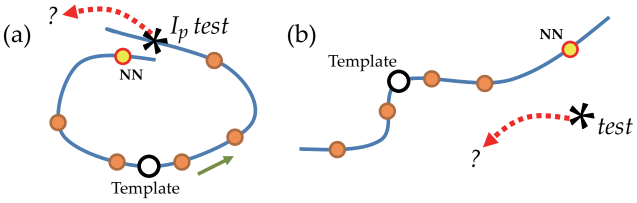


Fig. 24 Two important failure cases. **(a)** One-to-many mapping case. The manifold $\mathcal{I}$ is (almost) self-intersecting. As a result, two similar images have very different parameters, one large positive and the other large negative. If we pull-back the test using the wrong parameter, then Algorithm 1 diverges. Note this does not violate Theorem 2 since in such cases, $L_2 \to +\infty$ and many more train samples are required especially near the ambiguous region to ensure each time the nearest-neighbor procedure picks the correct one. **(b)** The test distorted image is not on the curve. In such a case, the pull-back bound does not hold (Eqn. (17)). As a result, the image sequence of successive warping-back does not approach the manifold $\mathcal{I}$ and Algorithm 1 is not guaranteed to converge. This often happens in the case of occlusion, resampling artifacts or an incomplete distortion model. Yet we empirically show that in such cases, Algorithm 1 still gives decent results.

12 Failure cases

Algorithm 1 works if Eqn. (7) holds universally within the sphere $\|\mathbf{p}\| \le r_0$. In the case of large distortions (r_0 large), the two positive constants (L_1 and L_2) take on their extreme values (0 and $+\infty$) and an infinite number of samples would be required. Eqn. (17) can also fail due to resampling artifacts in large distortions. Although our analysis ignores occlusions, it is possible to handle small occlusions using a more robust image distance metric (e.g., l_1-norm as shown in Section 8.3), but for substantial occlusions, an explicit model would be required. Some of the failure cases are summarized in Fig. 24.

13 Future Work

Although the accuracy $(1/\epsilon)$ is decoupled from the dimension d of the parameter space, in Eqn. (19) there is still a constant term that exponentially varies with d. To further reduce the required number of samples, a local distortion model may be used as in the case of our real experiments. However, better results can be obtained if we consider the correlations of distortions among nearby image regions. Better performance can also be obtained by using more distinctive features instead of raw image pixels for the nearest-neighbor search. In many scenarios, the bases $B(\mathbf{x})$ can be learned rather than be given beforehand. Finally, as a general framework, our method can potentially be used to avoid local minima in optimization tasks.

The algorithm can be used in many more applications, such as optical scanning of text, human pose estimation, marker-less motion capture and air turbulence. Yet in each case, more works need to be done to handle application-specific problems, such as self-occlusion, aliasing, cluttered background and so on.

Acknowledgement: This work was supported in parts by ONR grants N00014-08-1-0330 and DURIP N00014-06-1-0762, an Okawa Research grant and an NSF CAREER Award IIS-0643628.

This paper is the journal version of the conference paper “A Globally Optimal Data-Driven Approach for Image Distortion Estimation” published in Computer Vision and Pattern Recognition (CVPR), 2010.

Data and code are available at our website: http://www.cs.cmu.edu/~ILIM/projects/IM/globalopt/research_globalopt.html.

References

1. A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *PAMI*, 28(1):44–58, 2006. 2, 3, 8
2. S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR*, 2001. 2, 5, 8, 18, 19
3. S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004. 2, 11
4. A. Bissacco, M. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, 2007. 3
5. T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998. 2
6. A. Efros, V. Isler, J. Shi, and M. Visontai. Seeing through water. In *NIPS*, 2004. 1
7. A. Fathi and G. Mori. Human pose estimation using motion exemplars. In *ICCV*, 2007. 2
8. R. Framkot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *PAMI*, pages 439–451, 1988. 14
9. M. Gleicher. Projective registration with difference decomposition. In *CVPR*, 1997. 2, 5
10. G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998. 2, 8, 18, 19
11. R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *CVPR*, 2007. 3
12. X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *CVPR*, 2001. 2
13. F. Jurie and M. Dhome. Hyperplane approximation for template matching. *PAMI*, pages 996–1000, 2002. 3
14. E. Learned-Miller. Data driven image models through continuous joint alignment. *PAMI*, 28(2):236–250, 2006. 1
15. H. Ling and D. Jacobs. Deformation invariant image matching. In *ICCV*, 2005. 1
16. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 3
17. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981. 2
18. I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 2, 3
19. M. Nguyen and F. De la Torre. Local minima free parameterized appearance models. In *CVPR*, 2008. 3
20. J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *IJCV*, 76(2):109–122, 2008. 1
21. G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. Torr. Randomized trees for human pose detection. In *CVPR*, 2008. 3
22. R. Rosales and S. Sclaroff. Learning body pose via specialized maps. In *NIPS*, 2002. 3, 8
23. D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *Medical Imaging*, 18(8):712–721, 1999. 2, 13, 14, 15, 16, 17, 22
24. M. Salzmann and P. Fua. Reconstructing Sharply Folding Surfaces: A Convex Formulation. In *CVPR*, 2009. 3
25. M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *ICCV*, 2007. 18
26. M. Salzmann and R. Urtasun. Combining Discriminative and Generative Methods for 3D Deformable Surface and Articulated Pose Reconstruction. In *CVPR*, 2010. 3, 19
27. S. Sclaroff and J. Isidoro. Active blobs. In *ICCV*, 1998. 2
28. A. Shekhovtsov, I. Kovtun, and V. Hlavác. Efficient MRF deformation model for non-rigid image matching. *Computer Vision and Image Understanding*, 112(1):91–99, 2008. 3
29. A. P. Shon, K. Grochow, A. Hertzmann, and R. P. N. Rao. Learning Shared Latent Structure for Image Synthesis and Robotic Imitation. In *NIPS*, 2006. 3
30. H. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *IJCV*, 36(2):101–130, 2000. 2
31. L. Sigal, A. Balan, and M. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NIPS*, 2007. 3, 19
32. C. Sminchisescu, A. Kanaujia, and D. Metaxas. BM^{3}E: Discriminative Density Propagation for Visual Tracking. *PAMI*, 2007. 3
33. J. Taylor, A. Jepson, and K. Kutulakos. Non-Rigid Structure from Locally-Rigid Motion. In *CVPR*, 2010. 16, 17, 18
34. Y. Tian and S. G. Narasimhan. Seeing through Water: Image Restoration using Model-based Tracking. In *ICCV*, 2009. 1, 2, 13, 14
35. O. Tuzel, F. Porikli, and P. Meer. Learning on Lie Groups for Invariant Detection and Tracking. In *CVPR*, 2008. 2, 5, 8, 18, 19
36. Y. Wang, S. Lucey, and J. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, 2008. 3
37. Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense Non-rigid Surface Registration Using High-Order Graph Matching. In *CVPR*, 2010. 3
38. X. Zhao, H. Ning, Y. Liu, and T. Huang. Discriminative estimation of 3D human pose using gaussian processes. In *ICPR*, 2008. 3, 8

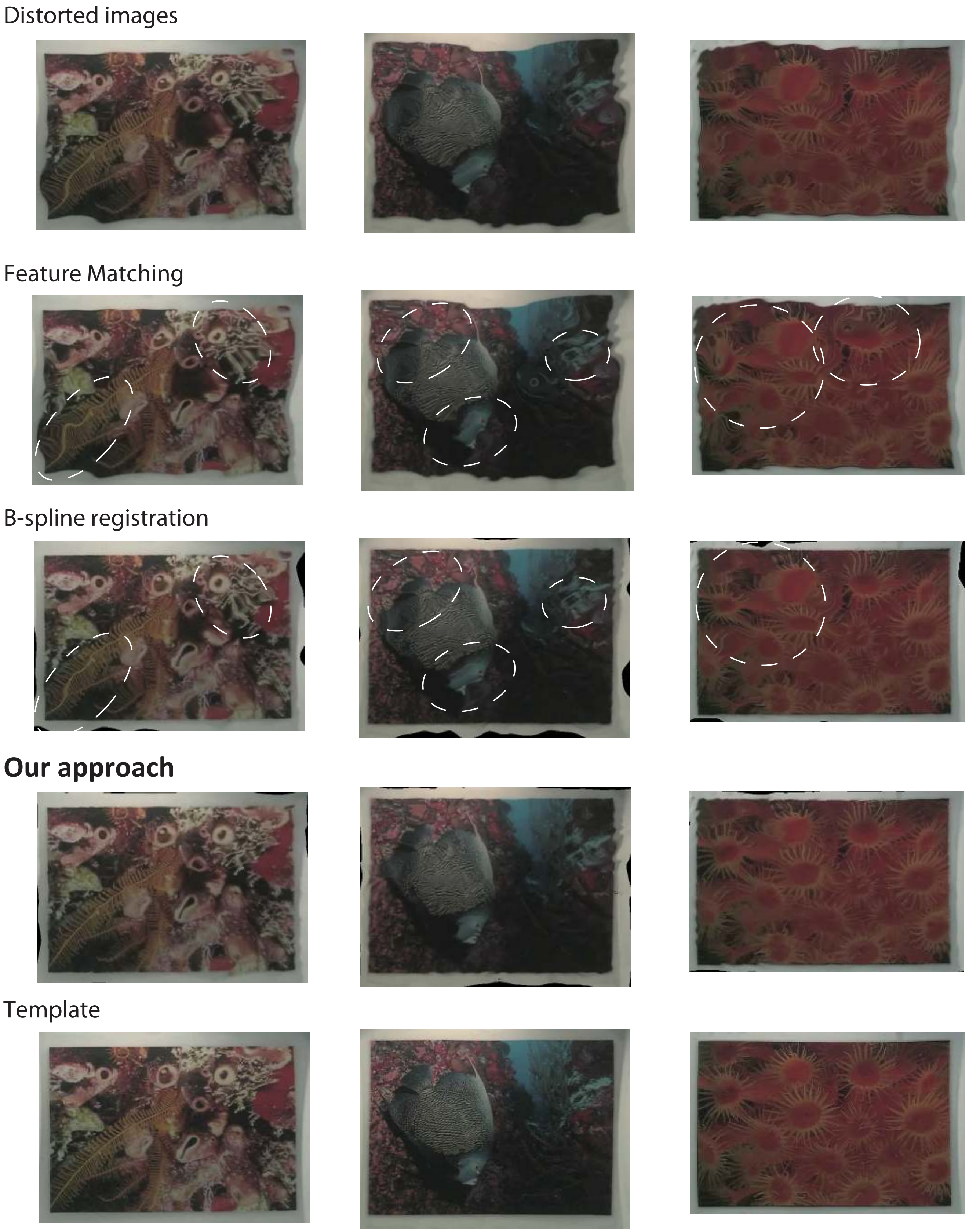


Fig. 25 Rectification of water distortion on 3 different colored texture images. Our method yields the best rectification. Detailed comparison is shown in Fig. 26 (Best viewed in color).

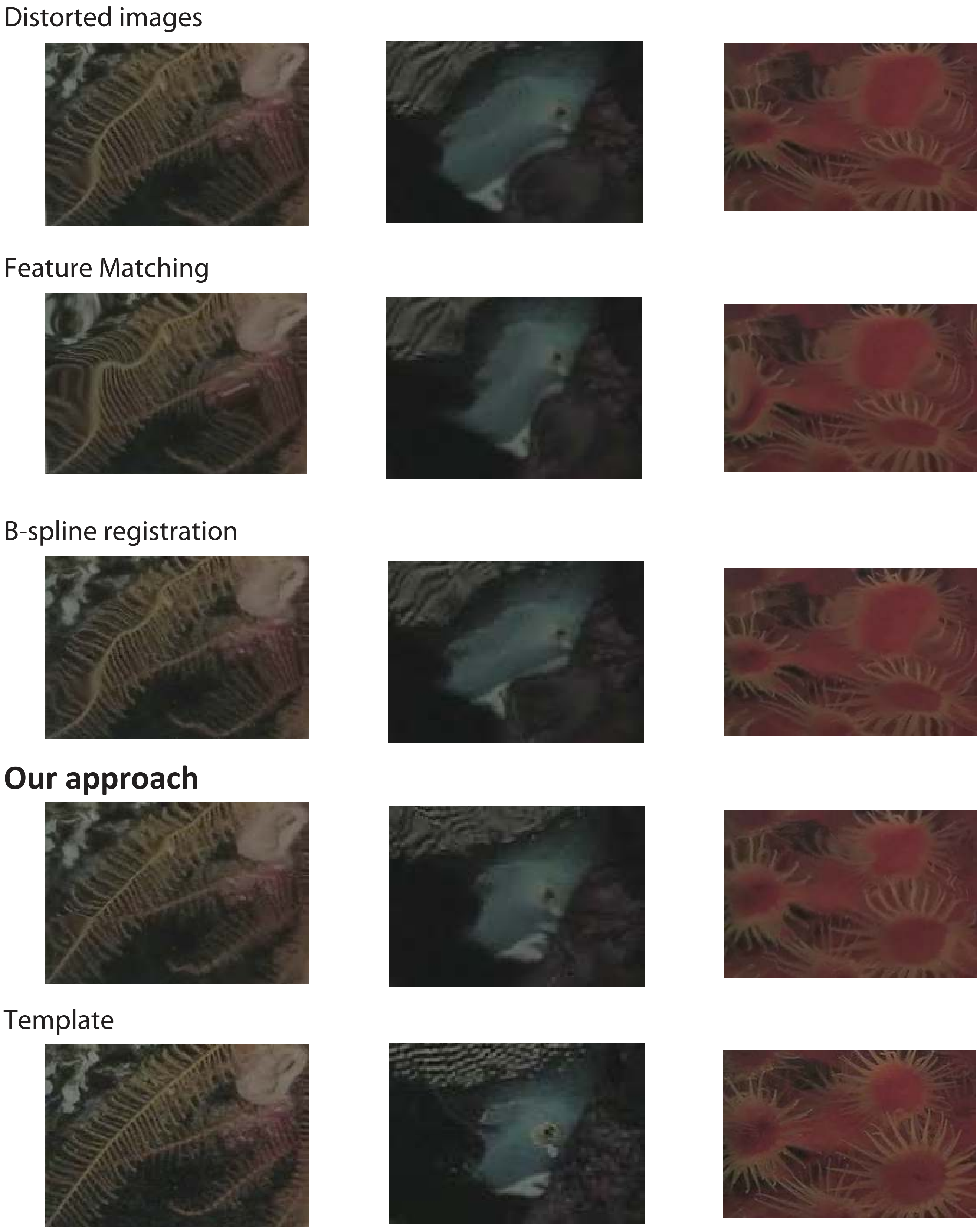


Fig. 26 Detailed comparision between our approach and previous works [23]. (Best viewed in color)

Appendix A

Proof of convergence of Algorithm 1 to the Global Optimum

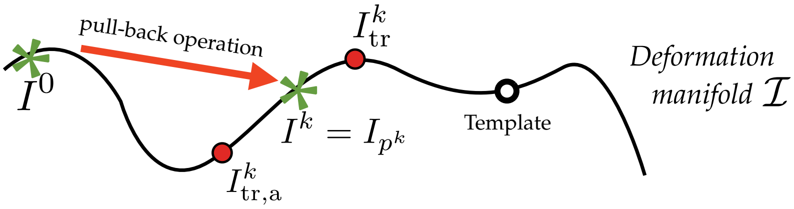


Fig. 27 Illustration of Theorem 3.

Theorem 3 (The global convergence of Algorithm 1 in the invertible warping case.) *If Eqn. (7) and (8) hold and $\beta < 1$, then Algorithm 1 computes an estimated mapping function $M'_K(I) \equiv \tilde{\mathbf{p}}^K_{\text{tr}} = \sum_{k=0}^{K} \mathbf{p}^k_{\text{tr}}$ so that for $\|M(I)\| \leq r_0$:*

$$\|M'_K(I) - M(I)\| \leq \beta^{K+1} r_0 \tag{30}$$

where $1 - \beta$ is the rate of convergence. In particular, $M'_K(I) \to M(I)$ if $K \to +\infty$.

Proof (Proof of Theorem 3) We set $\hat{\mathbf{p}}^k \equiv M(I^k)$, where $\hat{\mathbf{p}}^0 \equiv M(I^0)$ is what we want to know. The estimation residual is $\mathbf{p}^k \equiv \hat{\mathbf{p}}^0 - \tilde{\mathbf{p}}^{k-1}_{\text{tr}} = \hat{\mathbf{p}}^0 - \sum_{j=0}^{k-1} \mathbf{p}^j_{\text{tr}}$, and particularly $\mathbf{p}^0 = \hat{\mathbf{p}}^0$.

In the following, we prove by induction that the norm of the residue $\|\mathbf{p}^k\| \leq r_k \equiv \beta^k r_0$ for any k.

Base case. In the base case, we have $\|\mathbf{p}^0\| = \|\hat{\mathbf{p}}^0\| \leq r_0$ by the condition of this theorem.

Inductive case. Assume those conditions hold for k, in the following we prove they also hold for $k+1$. Since $I^k = H(I_0, \tilde{\mathbf{p}}^{k-1}_{\text{tr}}) = I_{\mathbf{p}^k}$ lies within the manifold $\mathcal{I}$, by the dense condition Eqn. (8), there exists a training sample $I^k_{tr,a} \in \mathcal{I}$ that is close to I^k:

$$\|I^k - I^k_{tr,a}\| \leq \frac{\beta \|\mathbf{p}^k\|}{L_2} \tag{31}$$

That means for rectified image I^k at iteration k, there is at least one training sample that is close to it. Thus, the nearest-neighbor I^k_{tr} of I^k must be even closer:

$$\|I^k - I^k_{tr}\| \leq \|I^k - I^k_{tr,a}\| \leq \frac{\beta}{L_2} \|\mathbf{p}^k\| \tag{32}$$

Thus their parameter is also close according to Eqn. (7):

$$\|M(I^k) - \mathbf{p}^k_{\text{tr}}\| = \|\mathbf{p}^k - \mathbf{p}^k_{\text{tr}}\| \leq \beta \|\mathbf{p}^k\| \tag{33}$$

which means the difference of current residue $\mathbf{p}^k$ and its estimation $\mathbf{p}^k_{\text{tr}}$ is bounded by $\beta\|\mathbf{p}^k\|$. Note such difference $\mathbf{p}^k - \mathbf{p}^k_{\text{tr}}$ is precisely the residue $\mathbf{p}^{k+1}$ in the next iteration. By the induction hypothesis, we have:

$$\|\mathbf{p}^{k+1}\| \leq \beta \|\mathbf{p}^k\| \leq \beta^2 \|\mathbf{p}^{k-1}\| \leq \ldots \leq \beta^{k+1} r_0 \to 0 \tag{34}$$

□

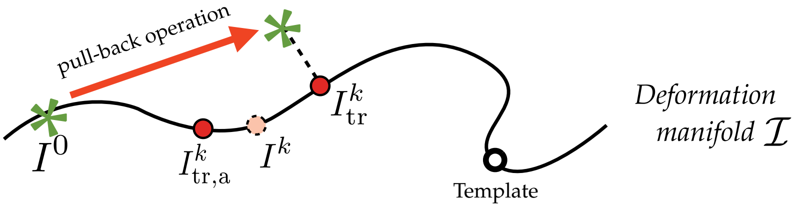


Fig. 28 Illustration of Theorem 4.

Theorem 4 **(The global convergence of Algorithm 1 in the general warping case.)** *If Eqn. (15), Eqn. (16), Eqn. (17) and Eqn. (8) hold and $\gamma \equiv 2\alpha + \beta < 1$ (where $\alpha = RL_2$ and R is defined in Eqn. (17)), then Algorithm 1 computes an estimated mapping function $M'_K(I) \equiv \tilde{\mathbf{p}}^K_{\mathrm{tr}} = \sum_{k=0}^K \mathbf{p}^k_{\mathrm{tr}}$ so that for $\|M(I)\| \leq r_0$:*

$$\|M'_K(I) - M(I)\| \leq \gamma^{K+1} r_0 \tag{35}$$

where $1-\gamma$ is the rate of convergence. In particular, $M'_K(I) \to M(I)$ if $K \to +\infty$.

Proof *(Proof of Theorem 4)* We set $\hat{\mathbf{p}}^k \equiv M(I^k)$, where $\hat{\mathbf{p}}^0 \equiv M(I^0)$ is what we want to know. The estimation residual is $\mathbf{p}^k \equiv \hat{\mathbf{p}}^0 - \tilde{\mathbf{p}}^{k-1}_{\mathrm{tr}} = \hat{\mathbf{p}}^0 - \sum_{j=0}^{k-1} \mathbf{p}^j_{\mathrm{tr}}$, and particularly $\mathbf{p}^0 = \hat{\mathbf{p}}^0$.

In the following, we prove by induction that the norm of the residue $\|\mathbf{p}^k\| \leq r_k \equiv \gamma^k r_0$ for any k.

Base case. In the base case, we have $\|\mathbf{p}^0\| = \|\hat{\mathbf{p}}^0\| \leq r_0$ by the condition of this theorem.

Inductive case. Assume those conditions hold for k, in the following we prove they also hold for $k+1$. By the pull-back bound(Eqn. (17)), we have for $I^k = H(I^0, \tilde{\mathbf{p}}^{k-1}_{\mathrm{tr}})$:

$$\|I^k - I_{\mathbf{p}^k}\| \leq R\|\mathbf{p}^k\| \tag{36}$$

Applying Eqn. (16) and we have

$$\|M(I^k) - \mathbf{p}^k\| \leq RL_2\|\mathbf{p}^k\| = \alpha\|\mathbf{p}^k\| \tag{37}$$

Note that we cannot use the dense condition (Eqn. (8)) directly to show the existence of a training sample that is close to I^k, since I^k is not necessarily lying on the manifold $\mathcal{I}$. Thus, we focus on the image $I_{\mathbf{p}^k}$ instead.

If there happens to be a training sample sitting at $I_{\mathbf{p}^k}$ and we happen to pick it at iteration k, then the algorithm returns the true parameter and terminates immediately with zero error. Without relying on pure luck, by the dense condition Eqn. (8), there exists a training sample $I^k_{tr,a} \in \mathcal{I}$ that is close to $I_{\mathbf{p}^k} \in \mathcal{I}$:

$$\|I_{\mathbf{p}^k} - I^k_{tr,a}\| \leq \frac{\beta\|\mathbf{p}^k\|}{L_2} \tag{38}$$

Using triangle inequality in the image space and we have:

$$\|I^k - I^k_{tr,a}\| \leq \left(R + \frac{\beta}{L_2}\right)\|\mathbf{p}^k\| \tag{39}$$

That means for rectified image I^k at iteration k, there is at least one training sample that is close to it. Thus, the nearest-neighbor I^k_{tr} of I^k must be even closer:

$$\|I^k - I^k_{tr}\| \leq \|I^k - I^k_{tr,a}\| \leq \left(R + \frac{\beta}{L_2}\right)\|\mathbf{p}^k\| \tag{40}$$

Thus their parameter is also close according to Eqn. (16):

$$\|M(I^k) - \mathbf{p}_{\mathrm{tr}}^k\| \leq (RL_2 + \beta)\|\mathbf{p}^k\| = (\alpha + \beta)\|\mathbf{p}^k\| \tag{41}$$

Finally, applying triangle inequality again on Eqn. (41) and Eqn. (42):

$$\|\mathbf{p}^k - \mathbf{p}_{\mathrm{tr}}^k\| \leq (2\alpha + \beta)\|\mathbf{p}^k\| = \gamma\|\mathbf{p}^k\| \tag{42}$$

which means the difference of current residue $\mathbf{p}^k$ and its estimation $\mathbf{p}_{\mathrm{tr}}^k$ is bounded by $\gamma\|\mathbf{p}^k\|$. Note such difference $\mathbf{p}^k - \mathbf{p}_{\mathrm{tr}}^k$ is precisely the residue $\mathbf{p}^{k+1}$ in the next iteration. By the induction hypothesis, we have:

$$\|\mathbf{p}^{k+1}\| \leq \gamma\|\mathbf{p}^k\| \leq \gamma^2\|\mathbf{p}^{k-1}\| \leq \ldots \leq \gamma^{k+1} r_0 \to 0 \tag{43}$$

□

Appendix B

The pull-back operation H

The pull-back operation H is a generalized version of inverse operation for non-invertible warping. Similar to the generating function (Eqn. (3) and Eqn. (4)), the pull-back operation is also an image transform that takes one image I_{input} and one parameter $\mathbf{p}$, and outputs another image I_{output}. The pull-back operation differs from the generating function in the sense that it is operated in the reverse direction.

For example, in the forward case, while the generating function $G_{\mathrm{F}}(I_{\mathrm{input}}, \mathbf{p})$ of warping *pushes* every pixel $\mathbf{x}$ of I_{input} to the destination located at $W(\mathbf{x}, \mathbf{p})$ in I_{output}, the corresponding pull-back function $H_{\mathrm{F}}(I_{\mathrm{input}}, \mathbf{p})$ *pulls* every pixel from location $W(\mathbf{x}, \mathbf{p})$ at the image I_{input} back to $\mathbf{x}$ at I_{output}, as shown in Fig. 29(a).

Similarly, in the backward case, while the generating function $G_{\mathrm{B}}(I_{\mathrm{input}}, \mathbf{p})$ of warping *pulls* every pixel $W(\mathbf{x}, \mathbf{p})$ of image I_{input} to the location $\mathbf{x}$ of I_{output}, the corrsponding pull-back function $H_{\mathrm{B}}(I_{\mathrm{input}}, \mathbf{p})$ *pushes* every pixel from location $\mathbf{x}$ of I_{input} to the location $W(\mathbf{x}, \mathbf{p})$ of I_{output}, as shown in Fig. 29(b).

From these definitions, we can see that $H_{\mathrm{B}} = G_{\mathrm{F}}$ and $H_{\mathrm{F}} = G_{\mathrm{B}}$.

In both cases, an important special case is that for a distorted image $I_{\mathbf{p}} = G(T, \mathbf{p})$, $H(I_{\mathbf{p}}, \mathbf{q}) = T$ for $\mathbf{p} = \mathbf{q}$, i.e., warping a template image T by parameter $\mathbf{p}$, and pulling-back the distorted image using the same parameter, yields exactly the template image T. This is trivial to prove from the definition of the pull-back operation, since each pixel that is pushed forward is exactly the same pixel that is pulled back. However, this fact that "H is a point inverse function" is critical in Algorithm 1 and the convergence analysis in Section 6.

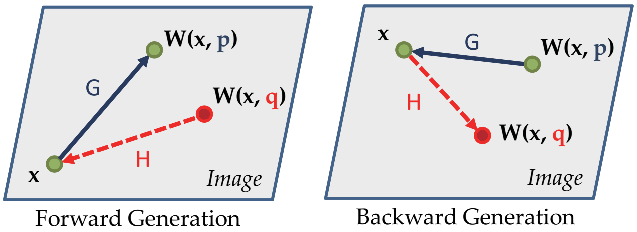


Fig. 29 The mechanism of the pull-back operation H that transform I_{input} to the output I_{output} via a parameter $\mathbf{p}$. In the forward case, a pixel $\mathbf{x}$ in I_{input} is pushed to the position $W(\mathbf{x}, \mathbf{p})$ of the I_{output} by the generating function G_{F}. The corresponding pull-back operation H do the opposite: it takes the pixel value at $W(\mathbf{x}, \mathbf{q})$ in image I_{input}, and stores it at position $\mathbf{x}$ in the resulting image I_{output}. In the case of $\mathbf{p} = \mathbf{q}$, the pixel pushed by G is the same pixel pulled by H, yielding $H(G(I, \mathbf{p}), \mathbf{p}) = T$. A similar reasoning holds in the backward case.

In the general case when $\mathbf{p} \neq \mathbf{q}$, the pull-back operation H behaves not exactly like the inverse function, but is close to it, as characterized by Eqn. (17). The following theorem shows the proof.

Theorem 5 (The Upper bound of the pull-back operation H**)** *Suppose the (backward) distorted image* $I_{\mathbf{p}-\mathbf{q}}$ *maps the pixel at location* L_G *in the template image* T *to the position* $\mathbf{y} \in \mathbb{R}^2$*, and the pulled-back image* $H(I_{\mathbf{p}}, \mathbf{q}) = G_{\mathrm{F}}(I_{\mathbf{p}}, \mathbf{q})$ *maps the pixel at location* L_H *in the template image* T *to the same position* $\mathbf{y}$*. Then we have the following bound if there exists an* $\mathbf{x}$ *so that* $\mathbf{y} = \mathbf{x} + B(\mathbf{x})\mathbf{q}$ *(Or* $W(\mathbf{x}, \mathbf{q})$ *is onto):*

$$\|L_G - L_H\|_1 \leq R' \|\mathbf{p} - \mathbf{q}\|_1 \tag{44}$$

where $R' = 2B_0 \min(B_1 \|\mathbf{q}\|_1, 2)$*,* $B_0 = \|B\|_\infty$ *and* $B_1 = \max_j \max_{\mathbf{x}} \max(\|\nabla b_j^x(\mathbf{x})\|_1, \|\nabla b_j^y(\mathbf{x})\|_1)$ *is the gradient bound of basis* $B(\mathbf{x})$ *(Note:* $\mathbf{b}_j(\mathbf{x}) = [b_j^x(\mathbf{x}); b_j^y(\mathbf{x})]$ *is a column vector at each* $\mathbf{x}$*). Therefore, we have*

$$\|H(I_{\mathbf{p}}, \mathbf{q}) - I_{\mathbf{p}-\mathbf{q}}\|_\infty \leq R \|\mathbf{p} - \mathbf{q}\|_1 \tag{45}$$

where $R = R' Q_1$ *and* $Q_1 = \max_{\mathbf{x}} \|\nabla T(\mathbf{x})\|_1$ *is the gradient bound of the template* T*.*

Proof (Proof of Theorem 5) According to Fig. 29, $H(I_{\mathbf{p}}, \mathbf{q})$ essentially moves the pixel located at $L_H \equiv \mathbf{x} + B(\mathbf{x})\mathbf{p}$ on the *template* T to the position $\mathbf{x} + B(\mathbf{x})\mathbf{q}$:

$$H : L_H \equiv \mathbf{x} + B(\mathbf{x})\mathbf{p} \longrightarrow \mathbf{x} + B(\mathbf{x})\mathbf{q} \tag{46}$$

This is valid for any $\mathbf{x} \in \mathbb{R}^2$. On the other hand, for the pixel $\mathbf{y}$ on the distorted image $I_{\mathbf{p}-\mathbf{q}}$, it comes from the pixel located at $L_G \equiv \mathbf{y} + B(\mathbf{y})(\mathbf{p} - \mathbf{q})$ in the template T:

$$G : L_G \equiv \mathbf{y} + B(\mathbf{y})(\mathbf{p} - \mathbf{q}) \longrightarrow \mathbf{y} \tag{47}$$

Since $W(\mathbf{x}, \mathbf{q})$ is onto, there exists $\mathbf{x}$ so that $\mathbf{y} = \mathbf{x} + B(\mathbf{x})\mathbf{q}$, then Eqn. (47) becomes

$$G : L_G \equiv \mathbf{x} + B(\mathbf{x})\mathbf{q} + B(\mathbf{x} + B(\mathbf{x})\mathbf{q})(\mathbf{p} - \mathbf{q}) \longrightarrow \mathbf{x} + B(\mathbf{x})\mathbf{q} \tag{48}$$

Note the destination(right) part of Eqn. (46) and Eqn. (48) are the same $(\mathbf{y})$, while the difference between the source(left) part of Eqn. (46) and Eqn. (48) is:

$$L_G - L_H = \left[B(\mathbf{x} + B(\mathbf{x})\mathbf{q}) - B(\mathbf{x})\right](\mathbf{p} - \mathbf{q}) \tag{49}$$

so we directly have the bound $\|L_G - L_H\|_1 \leq 4B_0\|\mathbf{p} - \mathbf{q}\|_1$ where $B_0 = \|B\|_\infty = \max_{\mathbf{x}} \max_j \max(|\mathbf{b}_j^x(\mathbf{x})|, |\mathbf{b}_j^y(\mathbf{x})|)$. In addition, using intermediate value theorem, from Eqn. (49) there exists $\{\boldsymbol{\xi}_1^x, \boldsymbol{\xi}_2^x, \ldots, \boldsymbol{\xi}_d^x\}$ and $\{\boldsymbol{\xi}_1^y, \boldsymbol{\xi}_2^y, \ldots, \boldsymbol{\xi}_d^y\}$ on the 2D line segment starting from $\mathbf{x}$ and ending at $\mathbf{x} + B(\mathbf{x})\mathbf{q}$ so that:

$$B^x(\mathbf{x} + B(\mathbf{x})\mathbf{q}) - B^x(\mathbf{x}) = \mathbf{q}^T B(\mathbf{x})^T \left[\nabla b_1^x(\boldsymbol{\xi}_1^x), \nabla b_2^x(\boldsymbol{\xi}_2^x), \ldots, \nabla b_d^x(\boldsymbol{\xi}_d^x)\right] \tag{50}$$

$$B^y(\mathbf{x} + B(\mathbf{x})\mathbf{q}) - B^y(\mathbf{x}) = \mathbf{q}^T B(\mathbf{x})^T \left[\nabla b_1^y(\boldsymbol{\xi}_1^y), \nabla b_2^y(\boldsymbol{\xi}_2^y), \ldots, \nabla b_d^y(\boldsymbol{\xi}_d^y)\right] \tag{51}$$

where $B^x(\mathbf{x}) = [b_1^x(\mathbf{x}), b_2^x(\mathbf{x}), \ldots, b_d^x(\mathbf{x})]$ and $B^y(\mathbf{x}) = [b_1^y(\mathbf{x}), b_2^y(\mathbf{x}), \ldots, b_d^y(\mathbf{x})]$ are the x and y component of $B(\mathbf{x})$. Then:

$$|L_G^x - L_H^x| \leq B_1 B_0 \|\mathbf{q}\|_1 \|\mathbf{p} - \mathbf{q}\|_1 \tag{52}$$

$$|L_G^y - L_H^y| \leq B_1 B_0 \|\mathbf{q}\|_1 \|\mathbf{p} - \mathbf{q}\|_1 \tag{53}$$

where $B_1 = \max_j \max_{\mathbf{x}} \max(\|\nabla b_j^x(\mathbf{x})\|_1, \|\nabla b_j^y(\mathbf{x})\|_1)$. Hence the bound. □