A Globally Optimal Data-Driven Approach for Image Distortion Estimation

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

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 the desired accuracy. 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 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

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. The first category of techniques match a set of sparse local features in the distorted image with those in the template [13, 12, 17]. Then, the parameters of a distortion model are estimated. These methods work well when the dimension d of the parameter space is low (eg., 6 for affine), but often fail in the presence of repetitive textures or high dimensional non-rigid distortions. Template matching techniques, such as Lucas-Kanade [14], Active-Appearance Models [5, 15] and free-form medical image registration [20], obtain dense correspondence between a distorted 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 techniques [3] that often converge to local minima. A convex approximation to the objective function can be learned [16, 26], but whether it remains faithful under large distortions is unclear.

Figure 1. Typical image distortions including water distortion, cloth deformation and text distortion (OCR or Captcha). Given a distorted image and an undistorted one (template), the goal is to estimate a dense deformation field between them. Images are adopted from [23, 21].

On the other hand, discriminative approaches [7, 1] learn a mapping M that directly predicts the distortion parameters $\mathbf{p}$ given 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. However, the well-known *curse of dimensionality* shows that an exponential number of 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 curse remains even in more advanced techniques including Relevant Vector Regression [1], Gaussian Processes [27], Boosting [4] or cluster-based regression [18].

The factor of $(1/\epsilon)^d$ is generally unavoidable, since for an arbitrary function $\mathbf{f}$, $\mathbf{f}(\mathbf{x}_1)$ and $\mathbf{f}(\mathbf{x}_2)$ are generally uncorrelated if $\mathbf{x}_1$ and $\mathbf{x}_2$ are far apart in high-dimensional space. However, two images that are distorted with very different distortion parameters still can share a large portion of the image content (albeit with different permutations of pixels). As a result, the prediction of the test image can be

[1] Other works [23, 11, 6] use a set of distorted images or videos as the input and compute distortions and/or the template.

made from the training images not in its neighborhood.

In this work, we draw upon the above insight to develop a novel data-driven iterative algorithm that combines the best of the generative and discriminative approaches for distortion estimation. 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). The algorithm is based on the notion of a "pull-back" operator that reuses training samples far away from the test image. We show under mild conditions that our algorithm converges to the global optimum using a significantly lower number of training samples $N = O(C^d \log 1/\epsilon)$ that grows only logarithmically with the desired accuracy $1/\epsilon$ (C is independent of ϵ). More importantly, the dimension d is decoupled from required accuracy $1/\epsilon$, breaking the curse of dimensionality[2]. Our approach is similar to [10] in terms of using randomly generated samples as training; however, [10] uses a spatially linear distortion model along with a linear estimator (hyperplane) that does not guarantee global optimality.

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 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 [23, 20]. In the future, we will explore broader applications such as face alignment, 3D registration of CT and range scans.

2. The Pull-back operator for Images

2.1. Problem formulation

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, $T = I_0 = G(T, 0)$. The function G can be implemented using an image warp $W(\mathbf{x}, \mathbf{p})$ (that maps a pixel $\mathbf{x}$ to the position $W(\mathbf{x}, \mathbf{p})$ and typically $W(\mathbf{x}, 0) = \mathbf{x}$) applied in either the forward or backward directions:

$$G_\mathrm{F}(T, \mathbf{p}): \quad I_\mathbf{p}(W(\mathbf{x}, \mathbf{p})) = T(\mathbf{x}) \tag{2}$$

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

[2]Other works [19, 22] have combined generative and discriminative approaches but without the desirable theoretical properties of our work.

Then, the main task of image registration is to estimate the distortion parameters $\mathbf{p}$ given $I_\mathbf{p}$, T and G (or warping function W). In particular, we will focus on occlusion-free warps in the 2D image space, which can cover not only affine transformations but also more complex non-rigid distortions due to water fluctuation and cloth deformation.

2.2. The Pull-back operation

Our work is based on the following key insight: two distorted images $I_\mathbf{p}$ and $I_\mathbf{q}$ share a significant amount of information, even if their parameters $\mathbf{p}$ and $\mathbf{q}$ are far apart. We introduce the notion of a pull-back operation that relates the two distorted images through their parameters and the generating function G. More specifically[3], the operation warps the image $I_\mathbf{p}$ using the parameter $\mathbf{q}$ to obtain a new image $G_\mathrm{B}(I_\mathbf{p}, \mathbf{q})$. In [24], we prove that $G_\mathrm{B}(I_\mathbf{p}, \mathbf{q})$ is close to a *less* distorted image $I_{\mathbf{p}-\mathbf{q}}$:

$$\|G_\mathrm{B}(I_\mathbf{p}, \mathbf{q}) - I_{\mathbf{p}-\mathbf{q}}\| \leq R\|\mathbf{p} - \mathbf{q}\| \tag{4}$$

for a broad class of warping functions of the form:

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

Here, R is a constant independent of $\mathbf{p}$ and $\mathbf{q}$ and $B(\mathbf{x}) = [\mathbf{b}_1(\mathbf{x}), \ldots, \mathbf{b}_d(\mathbf{x})]$ are the warping bases that can be obtained a priori using measured data or physical simulation.

Using Eqn. 4, in Section 3 we show that each successive pull-back operation gives a lesser and lesser distorted image until it reaches the template and the estimated parameters converge to the global optimum. This result significantly broadens the types of warps our algorithm can be applied to and sets our work apart from several previous works [25, 2, 9] that compute possibly local optima for a restricted set of warps. In particular, warps $W(\mathbf{x}, \mathbf{p})$ that form a group, such as affine and projective transform [8], are special cases in Eqn. 4 with $R = 0$.

3. Algorithm for distortion estimation

Based on the pull-back operation, we now present an iterative algorithm for distortion estimation. We start with the distorted test image I^0 and distorted training images $\{I_\mathrm{tr}\}$ with known parameters $\{\mathbf{p}_\mathrm{tr}\}$. In each iteration k, the algorithm finds the nearest training image $(I^k_\mathrm{tr}, \mathbf{p}^k_\mathrm{tr})$ to the distorted image I^k and performs a pull-back operation using $\mathbf{p}^k_\mathrm{tr}$ to get a new image I^{k+1}, that is less distorted compared to I^k. Then, the nearest training sample to I^{k+1} is found, the parameter estimation is updated and the procedure is iterated until convergence. To alleviate the possible error accumulation with successive resampling (interpolation), we

[3]This definition is for forward direction. For the backward direction, the pull-back operation is defined as the forward generating function G_F and the upper bound Eqn. 4 is still valid.

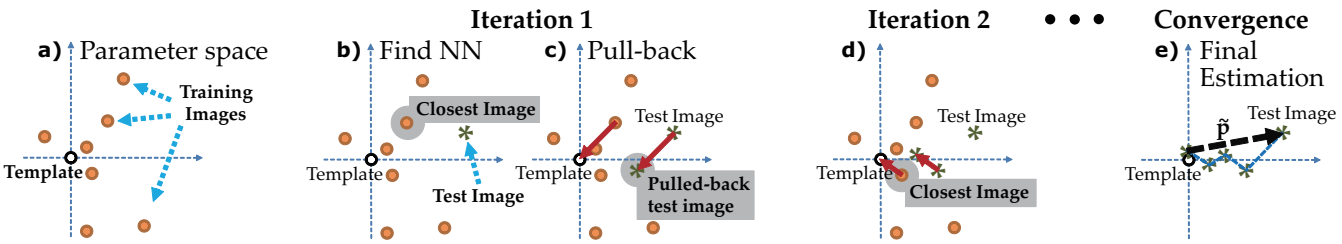

Figure 2. Algorithm for distortion estimation. **(a)** The template (origin) T and distorted training images $\{I_{\text{tr}}\}$ with known parameters $\{\mathbf{p}_{\text{tr}}\}$ are shown in the parameter space. **(b)** Given a distorted test image, its nearest training image $(I_{\text{tr}}, \mathbf{p}_{\text{tr}})$ is found. **(c)** The test image is "pulled-back" using $\mathbf{p}_{\text{tr}}$ 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.

obtain I^k by pulling-back the original test image I^0 using the cumulative estimation $\tilde{\mathbf{p}}^k_{\text{tr}}$ in each iteration. This is summarized in the algorithm below and is illustrated in Fig. 2.

The intuition behind this algorithm is that, in each iteration the selected training image need not to be ϵ-close to the test (as in the case of Nearest-Neighbor); it suffices to have the training images guiding the test image for a part of the way toward the goal (template). Then, another training image will continue to guide, and so on until the goal is reached. The reason we can perform this distortion-splitting is due to the existence of the pull-back operation. As a result, the training images which are far away from the test image in parameter space are reused. This observation is crucial to reducing the number of training images and breaking the curse of dimensionality.

Algorithm 1 The algorithm for distortion estimation

INPUT The training images $\{I^k_{\text{tr}}\}$ with known parameters $\{\mathbf{p}^k_{\text{tr}}\}$. The test image I^0.

for $k = 0 : K$ **do**

Find I^k's nearest training image I^k_{tr} with known parameter $\mathbf{p}^k_{\text{tr}}$ i.e., $I^k_{\text{tr}} = \arg\min_i \|I^k - I^i_{\text{tr}}\|$.

Set cumulative estimation $\tilde{\mathbf{p}}^k_{\text{tr}} = \sum_{j=0}^{k} \mathbf{p}^j_{\text{tr}}$.

Set pulled-back test image $I^{k+1} = G_{\text{B}}(I^0, \tilde{\mathbf{p}}^k_{\text{tr}}) = I^0(W(\mathbf{x}, \tilde{\mathbf{p}}^k_{\text{tr}}))$.

end for

OUTPUT $\tilde{\mathbf{p}}^K_{\text{tr}}$ is the final estimation.

3.1. Convergence property of the algorithm

We now prove that the above algorithm converges to the true parameters, given sufficient number of samples and under mild conditions.

Consider the set of all distorted images whose distortion parameters $\mathbf{p}$ are within the sphere S_{r_0}: $\|\mathbf{p}\| \leq r_0$. The origin of this space corresponds to the undistorted template image T. In this section, we will show how to distribute the training images within this sphere such that *any* test image within S_{r_0} will be transformed to the origin (template) by Alg. 1.

Let M be the unknown mapping function that predicts the parameters $\mathbf{p}$ given the image $I_{\mathbf{p}}$. We make the following two assumptions:

1. The mapping M is one-to-one and smooth. Mathematically, there exist two universal constants $0 < L_1 \leq L_2 < +\infty$ so that for two images I and I' within S_{r_0}:

$$L_1\|I - I'\| \leq \|M(I) - M(I')\| \leq L_2\|I - I'\| \quad (6)$$

Note that a one-to-many mapping M corresponds to $L_2 = +\infty$, in which case an infinite number of samples are needed to get an accurate estimation. Using the definition of M: $M(I_{\mathbf{p}-\mathbf{q}}) \equiv \mathbf{p} - \mathbf{q}$ and substituting Eqn. 4 into Eqn. 6, we have:

$$\|M(G_{\text{B}}(I_{\mathbf{p}}, \mathbf{q})) - (\mathbf{p} - \mathbf{q})\| \leq \alpha\|\mathbf{p} - \mathbf{q}\| \quad (7)$$

where $\mathbf{p} = M(I)$ and $\alpha = L_2 R$.

2. Training images are more densely distributed near the origin. Unlike nearest neighbor that places the training images uniformly in the space to achieve best worse-case performance (leading to an exponential number of samples), we place the training images sparsely at the periphery of S_{r_0}, and densely only near the origin. This distribution can be mathematically stated as follows: given I with $\|M(I)\| \leq r$, we assume that we can find a training image I_{tr} so that

$$\|I - I_{\text{tr}}\| \leq \beta r / L_2 \quad (8)$$

where $\beta < 1$.

Then, we have the following Theorem 3.1 that proves the convergence of our algorithm to the global optimum.

Theorem 3.1 *If Eqn. 6, 7 and 8 hold and* $\gamma \equiv \alpha + \beta(1 + \alpha) < 1$, *then Alg. 1 computes an estimated mapping* $M'_K(I) \equiv \tilde{\mathbf{p}}^K_{\text{tr}} = \sum_{k=0}^{K} \mathbf{p}^k_{\text{tr}}$ *so that for* $\|M(I)\| \leq r_0$:

$$\|M'_K(I) - M(I)\| \leq \gamma^{K+1} r_0 \quad (9)$$

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

That is, in each iteration the norm of the residual between the estimated and true parameters is contracted by γ, and thus Alg. 1 converges. We verify that $\gamma < 1$ on synthetic data in Section 4.2. See Appendix for the proof.

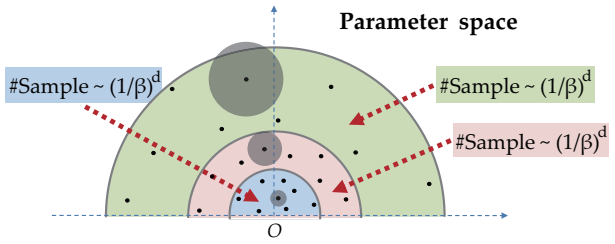

Figure 3. The number of samples needed to fill a give sphere $\|\mathbf{p}\| \leq r$ is independent of r since the allowed prediction uncertainty (shown in gray solid circle) is proportional to r. As a result, only a small neighborhood of the origin O requires dense sampling. This is the key to break the curse of dimensionality.

3.2. The number of training images needed

Using the strategy of Eqn. 8, we now show the number N of required training images grows only logarithmically with respect to the prediction accuracy $1/\epsilon$. Recall that we are interested in populating the samples within a sphere using more samples near the origin than in the periphery. We know that in order to fill a d-dimensional sphere of radius r_1, we require $O((r_1/r_2)^d)$ smaller spheres of radius $r_2 < r_1$. Secondly, in order to fill a sphere $\|M(I)\| \leq r$ in the parameter space, it suffices to fill the sphere of $\|I-T\| \leq r/L_1$ in the image space. This is because we have $r \geq \|M(I)\| = \|M(I)-M(T)\| \geq L_1\|I-T\|$ using the left side of Eqn. 6 and $M(T) = 0$. Thus, only $O((L_2/\beta L_1)^d)$ samples are needed in order to satisfy Eqn. 8. Crucially, this is independent of r (See Fig. 3). Thus for K iterations, $O(K(L_2/\beta L_1)^d)$ samples are needed.

On the other hand, using Eqn. 9, we compute $K = \lceil \log(r_0/\epsilon)/\log(1/\gamma) \rceil - 1$ for a given accuracy $1/\epsilon$. As a result, the total number $N(\epsilon, \alpha, \beta)$ of training images is:

$$N(\epsilon, \alpha, \beta) = O\left[\left(\frac{L_2}{\beta L_1}\right)^d \frac{\log r_0/\epsilon}{\log 1/\gamma}\right] \qquad (10)$$

where, $\gamma \equiv \alpha + \beta(1+\alpha)$ as defined in Theorem 3.1. A large β implies fewer training samples in each iteration but more iterations, and vice versa. The optimal β^*, which is independent of ϵ, can be obtained by minimizing Eqn. 10. As a result, Eqn. 10 grows logarithmically with respect to the accuracy $1/\epsilon$. In contrast, Nearest-Neighbor requires $O((L_2/\epsilon L_1)^d)$ samples for the same accuracy. In Fig. 4(b), we show the drastic differences in performance 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.**

3.3. Extensions of Alg. 1

Sample distribution. The convergence property of our algorithm is *independent* of the distribution of the test samples within the sphere $\|\mathbf{p}\| \leq r_0$, if the training samples are distributed as explained before. This differs from many approaches that only work for a given prior distribution. 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.

K_{NN} nearest neighbors. In practice, due to the constant factor $(L_2/\beta L_1)^d$, the N given by Eqn. 10 can be a large number. Using K_{NN} nearest neighbors with weighted voting (i.e., kernel regression) can further reduce the required samples, as shown in Fig. 4(e).

Incorporating temporal knowledge. Although Alg. 1 does not assume temporal relationship between two distorted images, 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 iteration. 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.

Regressor bag and active sampling. 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 [27]), 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.

4. Analysis of the algorithm using simulations

4.1. 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. 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. The standard derivations of the 1-st and 20-th principle components are 11.63 and 7.95 respectively. For each of the 100 template images, we synthesize $N = 1000$ distorted images for the training set and 10 for the test set. Alg. 1 is applied to each test image to obtain the relative (squared) error $e = \|\mathbf{p}_{\mathrm{true}} - \tilde{\mathbf{p}}\|_2^2 / \|\mathbf{p}_{\mathrm{true}}\|_2^2$.

Fig. 4(a) shows the successful convergence of our algorithm averaged over all the test images. Fig. 5 shows example images warped with different magnitudes of distortion and the computed rectified images. Particularly, notice

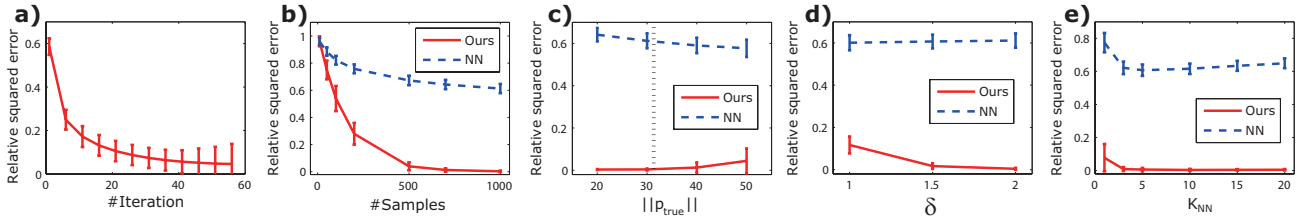

Figure 4. The effects of four different factors on the performance of the algorithm in terms of relative squared error. **(a)** Average convergence behavior computed over all test images. **(b)** The higher the number of training images, the better the performance. Note our performance is much better 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 $\|\mathbf{p}_{\text{tr}}\|$ in the training images). **(e)** Using K_{NN}-nearest neighbor with weighted voting lessens the training samples further.

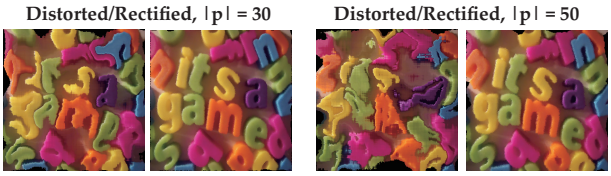

Figure 5. Sample images distorted to various degrees and the recovered rectified images.

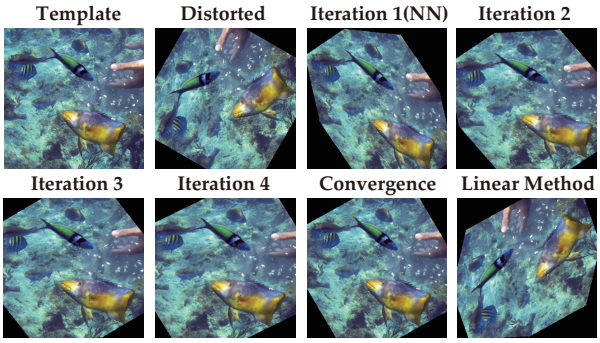

Figure 6. Successful convergence of our algorithm for affine transformed image, given there are at least one training sample reaching that area. In contrast, linear methods (like Lucas-Kanade) get stuck in local minima even by using a coarse-to-fine strategy.

the significant improvement in the most distorted example. Fig. 6 illustrates an image distorted by a 60 degree rotation. Even if a coarse-to-fine strategy is used, linear 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.

4.2. Behavior of the algorithm

Factors that affect the algorithm. There are four major factors that affect the performance of the algorithm, including the number N of training samples used, the number K_{NN} of nearest neighbors for kernel regression, the shape of the distribution of training images, and the magnitude of distortion $\|\mathbf{p}_{\text{true}}\|$ of the test images. We generate the training samples using a sphere-symmetric distribution $g(D^{-1}\mathbf{p})$, where D is a diagonal matrix of standard deviations in each dimension and $g(\mathbf{p}) \propto g(\|\mathbf{p}\|) = \text{Uniform}(0,1)^{\delta}$ where δ is a constant related to the concentration of samples around the origin. For $\delta = 1$ we get a uniform distribution, for $\delta > 1$ we get a distribution peaked around the origin.

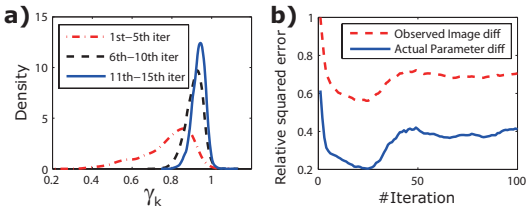

Figure 7. **(a)** The empirical distribution of relative prediction error γ_k on test images in different iterations of the algorithm. 99.2% of the γ_k is small than 1, justifying $\gamma < 1$ in Theorem 3.1; others are due to insufficient samples. **(b)** The U-turn behavior in large distortion ($\|\mathbf{p}_{\text{true}}\| = 50$), when the resampling artifacts are severe.

We set the default values of the four factors to be $N = 1000$, $K_{\text{NN}} = 10$, $\delta = 2$ and $\|\mathbf{p}_{\text{true}}\| = 30$. Fig. 4(b)-(e) shows performance variations when perturbing one factor and keeping the rest constant. Fig. 4(b) shows better performance is obtained with more training images. Although nearest neighbor behaves similarly, its performance is much poorer for the same number of samples. Fig. 4(c) shows that a high accuracy is obtained if training samples are concentrated around the origin given the test image is within their range, as supported by the theoretical analysis. Conversely, the performance drops if a test image is far away from the training set (Fig. 4(d)). Finally, Fig. 4(e) shows that the parameter prediction using multiple neighbors reduces the samples required even further.

Verifying $\gamma < 1$ in Theorem 3.1. Fig. 7(a) shows how the distribution of relative prediction errors $\gamma_k \equiv \|\mathbf{p}^k_{\text{true}} - \tilde{\mathbf{p}}^k_{\text{tr}}\|/\|\mathbf{p}^k_{\text{true}}\|$ on the test images changes over iterations. 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

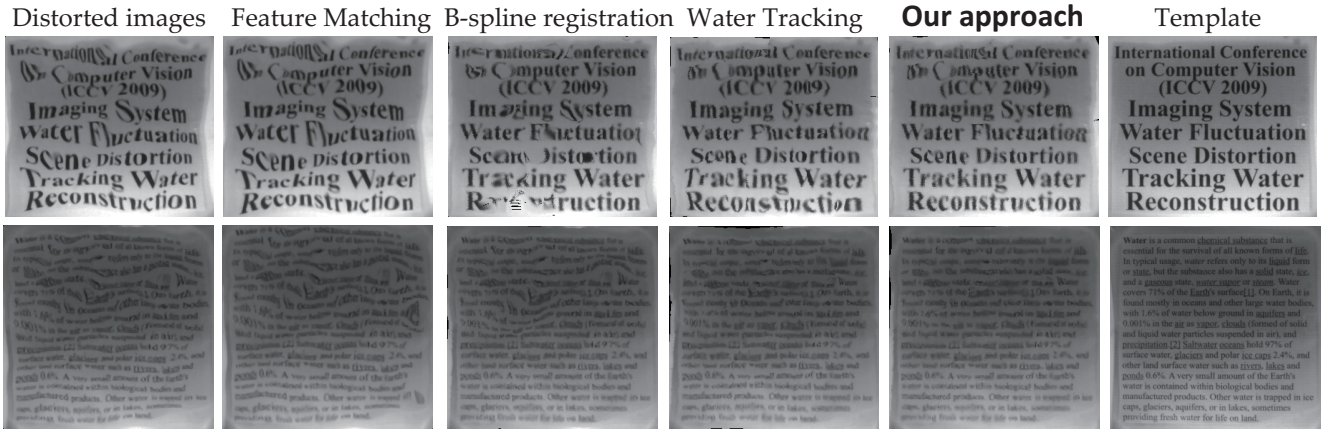

Figure 8. Rectification of water distortion on text images of different font sizes. Our approach outperforms HOG feature matching, b-spline nonrigid registration [20] and yields slightly better results with water tracking [23]. However, water tracking relies on the entire video frames, while ours only needs two images.

that the rate of convergence slows with increasing iterations.

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, even 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. 7(b)). Hence, we conjecture that the solution that produces minimum error among many iterations will be a reasonable one.

5. Real Experiments

We validate our algorithm on real videos, including water distortion induced by the surface refraction and deformations induced by cloth movement. We use $N = 10000$ samples, $\delta = 2$ and $K_{\text{NN}} = 10$ in all the cases. We synthetically generate the training samples from the template using the distortion model in Eqn. 5 where warping bases $B(\mathbf{x})$ are chosen for particular scenes. All the test images (except for texts) are captured with a color video camera and the algorithm is run on gray-scale image patches. Please go to our website for datasets, codes and video results.

Water Distortion. We use the image taken under flat water surface as the template. We use the water bases (57×40) in [23] with $d = 20$ and apply Alg. 1 to their videos (200×300) containing distorted text of various font sizes. We also acquired additional distorted videos (360×240) of underwater scene textures with a setup similar to [23].

We compared our algorithm to three other representative techniques: free form non-rigid image registration using b-splines [20], our previous work of water tracking [23] and a baseline approach where we compute and match HOG features and interpolate the sparse correspondence to create a dense deformation field. Fig. 8 shows the rectified images for a scene with text, and Fig. 9 shows the results for a scene with colored textures. Since only sparse correspondences between two images are used, feature tracking gives an inaccurate interpolated deformation field and fails to align details well. Non-rigid B-spline image registration works better but fails on some parts due to local minima. Water tracking uses a video (61 frames) to produce results better than feature matching and B-spline registration. In contrast, our method yields the best rectification results given only the template and one distorted image at a time.

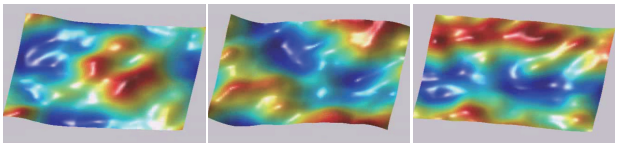

Figure 11. Reconstructed water surface by spatially integrating the water distortion (best viewed in color).

Cloth Deformation. We use a dataset acquired by manually perturbing silk cloth. Since cloth deformation behaves more globally than water distortion, we use the following two-stage approach. First we downsample the original video (720×480) by a factor of 2 and apply local 200×200 affine bases and estimate the 6 parameters using our method. Secondly, we apply local random bases (100×100) with 40 dimensions to the resulting undistorted video sequence, and obtain the final distortion estimation by distortion composition. Fig. 13 shows three accurately tracked frames using estimated distortion.

6. Limitations and Future work

Alg. 1 works if Eqn. 6 holds universally within the sphere $\|\mathbf{p}\| \leq r_0$. In the case of large distortions (r_0 large),

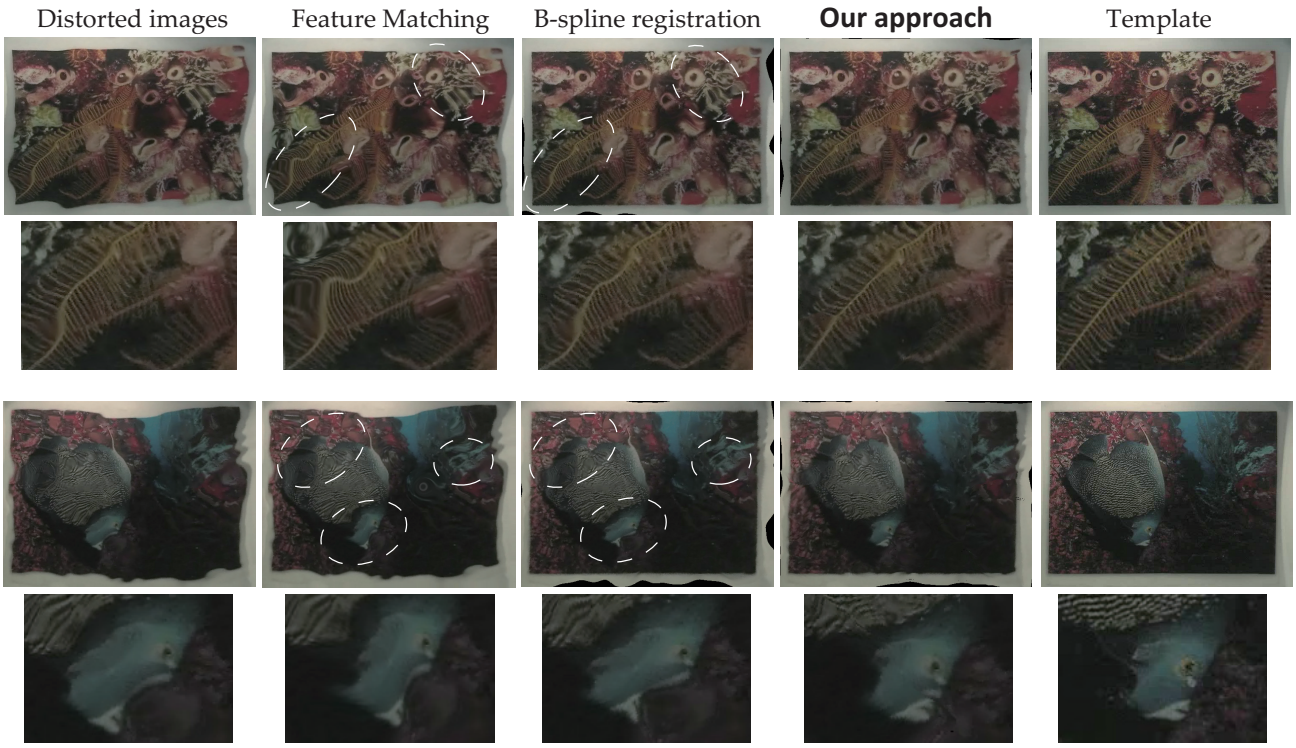

Figure 9. Rectification of water distortion on 2 different colored texture images. Our method yields the best rectification. Note the even rows show the details of the rectified images. (best viewed in color).

Figure 10. Tracking a video after undistortion. 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). See our website for the complete video.

the two positive constants (L_1 and L_2) take on their extreme values (0 and $+\infty$) and an infinite number of samples are required. Eqn. 4 can also fail due to resampling artifacts in large distortions, as shown in Fig. 12. Although our analysis ignores occlusions, we believe it will be possible to handle small occlusions using a more robust image distance metric (e.g., L1-norm), but harder cases will require an explicit model of occlusions.

Although the accuracy of $1/\epsilon$ is decoupled from the dimension d of the parameter space, in Eqn. 10 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 instead of the analytical ones used. Finally, as a general framework, our method can potentially be used to avoid local minima in optimization tasks.

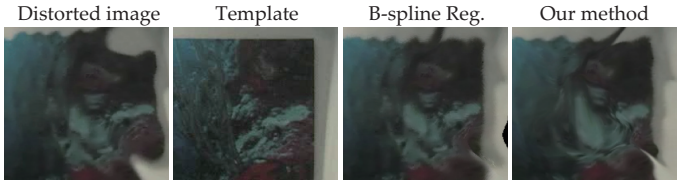

Figure 12. Typical failure case due to severe resampling artifacts. Note all the methods fail in this case.

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.

Appendix

Proof of Theorem 3.1 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 $\mathbf{p}^k \equiv \hat{\mathbf{p}}^0 - \tilde{\mathbf{p}}_{\text{tr}}^{k-1}$, and particularly $\mathbf{p}^0 = \hat{\mathbf{p}}^0$.

We prove by induction that the norm of the residue

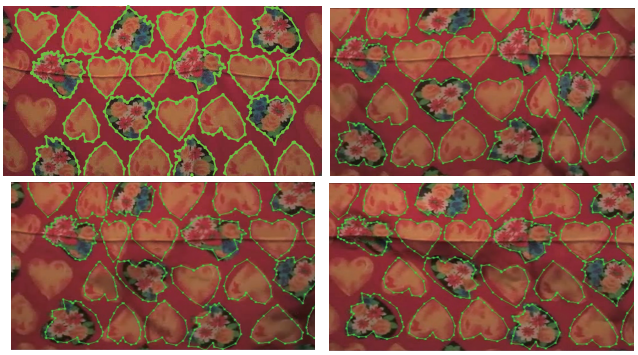

Figure 13. Tracking results of cloth deformation. Top-left is the template with manually-labeled shapes. The rest are the tracking results. See our website for the complete videos.

$\|\mathbf{p}^k\| \leq r_k \equiv \gamma^k r_0$ for any k. In the base case we have $\|\mathbf{p}^0\| = \|\hat{\mathbf{p}}^0\| \leq r_0$ by the condition of Theorem 3.1. Assume those conditions hold for k, in the following we prove they also hold for $k+1$. By Eqn. 7, in the forward case we have for $I^k = G_{\mathrm{B}}(I^0, \tilde{\mathbf{p}}_{\mathrm{tr}}^{k-1})$ (backward is similar):

$$\|M(I^k) - (\hat{\mathbf{p}}^0 - \tilde{\mathbf{p}}_{\mathrm{tr}}^{k-1})\| = \|\hat{\mathbf{p}}^k - \mathbf{p}^k\| \leq \alpha \|\mathbf{p}^k\| \leq \alpha r_k \tag{11}$$

where $\hat{\mathbf{p}}^k \equiv M(I^k)$. Moreover, from Eqn. 11 we have

$$\|\hat{\mathbf{p}}^k\| \leq (1+\alpha)\|\mathbf{p}^k\| \leq (1+\alpha) r_k \tag{12}$$

Then using Eqn. 8 and Eqn. 12, we can find I_{tr}^k so that

$$\|I_{\mathrm{tr}}^k - I^k\| \leq \frac{\beta(1+\alpha) r_k}{L_2} \tag{13}$$

By Eqn. 6, we have

$$\|M(I_{\mathrm{tr}}^k) - M(I^k)\| = \|\mathbf{p}_{\mathrm{tr}}^k - \hat{\mathbf{p}}^k\| \leq \beta(1+\alpha) r_k \tag{14}$$

Combine Eqn. 11 and Eqn. 14, we have

$$\|\mathbf{p}^k - \mathbf{p}_{\mathrm{tr}}^k\| \leq \|\mathbf{p}_{\mathrm{tr}}^k - \hat{\mathbf{p}}^k\| + \|\hat{\mathbf{p}}^k - \mathbf{p}^k\| \tag{15}$$

$$\leq [\alpha + \beta(1+\alpha)] r_k = r_{k+1} \tag{16}$$

Since $\mathbf{p}^{k+1} = \mathbf{p}^k - \mathbf{p}_{\mathrm{tr}}^k$, we have $\|\mathbf{p}^{k+1}\| \leq r_{k+1}$. ∎

References

[1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *PAMI*, 28(1):44–58, 2006. 1, 4

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR*, 2001. 2

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004. 1

[4] A. Bissacco, M. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, 2007. 1

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998. 1

[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. 1

[8] M. Gleicher. Projective registration with difference decomposition. In *CVPR*, 1997. 2

[9] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998. 2

[10] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *PAMI*, pages 996–1000, 2002. 2

[11] E. Learned-Miller. Data driven image models through continuous joint alignment. *PAMI*, 28(2):236–250, 2006. 1

[12] H. Ling and D. Jacobs. Deformation invariant image matching. In *ICCV*, 2005. 1

[13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[14] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981. 1

[15] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 1

[16] M. Nguyen and F. De la Torre. Local minima free parameterized appearance models. In *CVPR*, 2008. 1

[17] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *IJCV*, 76(2):109–122, 2008. 1

[18] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. Torr. Randomized trees for human pose detection. In *CVPR*, 2008. 1

[19] R. Rosales and S. Sclaroff. Learning body pose via specialized maps. In *NIPS*, 2002. 2

[20] 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. 1, 2, 6

[21] M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3-d tracking. In *ICCV*, 2007. 1

[22] L. Sigal, A. Balan, and M. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NIPS*, 2007. 2

[23] Y. Tian and S. G. Narasimhan. Seeing through Water: Image Restoration using Model-based Tracking. In *ICCV*, 2009. 1, 2, 6

[24] Y. Tian and S. G. Narasimhan. Theoretical Bounds for the Distortion Estimation Algorithm. *CMU RI Tech. Report*, 2010. 2

[25] O. Tuzel, F. Porikli, and P. Meer. Learning on Lie Groups for Invariant Detection and Tracking. In *CVPR*, 2008. 2

[26] Y. Wang, S. Lucey, and J. Cohn. Enforcing convexity for improved alignment with constrained local models. In *CVPR*, 2008. 1

[27] X. Zhao, H. Ning, Y. Liu, and T. Huang. Discriminative estimation of 3D human pose using gaussian processes. In *ICPR*, 2008. 1, 4