Int J Comput Vis (2015) 115:44–67
DOI 10.1007/s11263-015-0838-5

Theory and Practice of Hierarchical Data-driven Descent for Optimal Deformation Estimation

Yuandong Tian[1] · **Srinivasa G. Narasimhan**[1]

Received: 4 August 2014 / Accepted: 15 June 2015 / Published online: 14 July 2015
© Springer Science+Business Media New York 2015

Abstract Real-world surfaces such as clothing, water and human body deform in complex ways. Estimating deformation parameters accurately and reliably is hard due to its high-dimensional and non-convex nature. Optimization-based approaches require good initialization while regression-based approaches need a large amount of training data. Recently, to achieve globally optimal estimation, data-driven descent (Tian and Narasimhan in Int J Comput Vis , 98:279–302, 2012) applies nearest neighbor estimators trained on a particular distribution of training samples to obtain a globally optimal and dense deformation field between a template and a distorted image. In this work, we develop a hierarchical structure that first applies nearest neighbor estimators on the entire image iteratively to obtain a rough estimation, and then applies estimators with local image support to refine the estimation. Compared to its non-hierarchical version, our approach has the theoretical guarantees with significantly fewer training samples, is faster by several orders, provides a better metric deciding whether a given image requires more (or fewer) samples, and can handle more complex scenes that include a mixture of global motion and local deformation. We demonstrate in both simulation and real experiments that the proposed algorithm successfully tracks a broad range of non-rigid scenes including water, clothing, and medical images, and compares favorably against several other deformation estimation and tracking approaches that do not provide optimality guarantees.

Keywords Deformation modeling · Globally optimal solutions · Non-rigid deformation · Data-driven approach · Non-linear optimization · Non-convex optimization · Image deformation · High-dimensional regression

Communicated by Phil Torr, Steve Seitz, Yi Ma, and Kiriakos Kutulakos.

Yuandong Tian is now in Facebook AI Research.

✉ Yuandong Tian
yuandong.tian@gmail.com
http://www.yuandong-tian.com

Srinivasa G. Narasimhan
srinivas@cs.cmu.edu
http://www.cs.cmu.edu/~ILIM/

[1] The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

1 Introduction

Accurately finding dense correspondence between images capturing deforming objects is important for many vision tasks, such as 3D reconstruction, image alignment and tracking. However, estimating the parameters of nonrigid deformation is hard due to its high-dimensionality and strong non-convexity. Continuous optimization approaches (e.g. gradient descent or Newton's method) require no training but often suffer from local minima, while regression-based approaches (e.g., nearest neighbor) have guaranteed solutions (i.e., the prediction $\hat{\mathbf{p}}$ satisfies $\|\hat{\mathbf{p}} - \mathbf{p}\| \leq \epsilon$, where $\mathbf{p}$ is the true parameters), only when $O(1/\epsilon^d)$ training samples are available.

Recently, Tian and Narasimhan (2012) proposed data-driven descent which combines the best properties of both continuous optimization and regression. They show that in the presence of a generative model for deformation, the training samples can be generated by simply deforming the template using parameters from a particular distribution. Then a sequence of nearest neighbor predictions will achieve the globally optimal solution, that is, find $\hat{\mathbf{p}}$ so that $\|\hat{\mathbf{p}} - \mathbf{p}\| \leq \epsilon$ for the true solution $\mathbf{p}$. This global solution

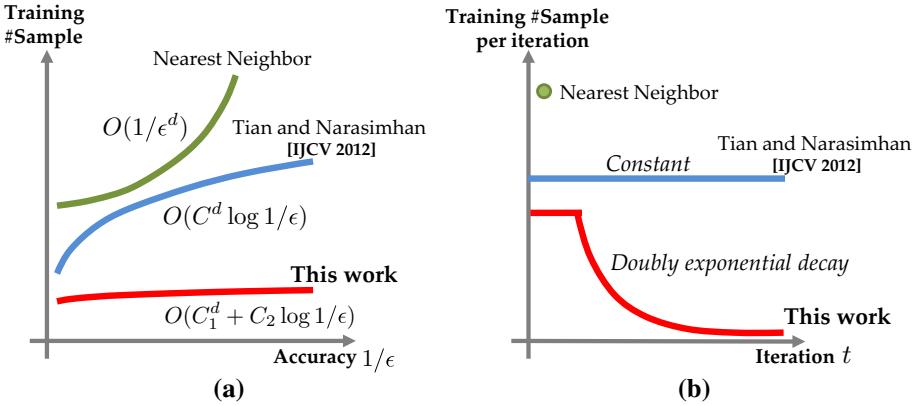

Fig. 1 Illustrations of order of training sample complexity required for estimating d dimensional deformation. **a** To achieve a guaranteed accuracy $1/\epsilon$, traditional regression-based approaches (e.g. Nearest neighbor) require $O(1/\epsilon^d)$ training samples. Data-driven descent (Tian and Narasimhan 2012) requires $O(C^d \log 1/\epsilon)$, decoupling the dimensionality from the accuracy. Our hierarchical framework for deformation estimation achieves $O(C_1^d + C_2 \log 1/\epsilon)$ with constant C_1 much smaller than C and C_2 independent of dimensionality. **b** Sample complexity per iteration. A constant number of samples per iteration is needed in Tian and Narasimhan (2012). The number of samples needed is a constant for the first few iterations, and then decays *double exponentially* for our algorithm

essentially warps the test image to the template.[1] Furthermore, to achieve the accuracy of $1/\epsilon$, the number of samples needed is $O(C^d \log 1/\epsilon)$ for d dimensional warping, much fewer than $O(1/\epsilon^d)$ required for general regressions. Intuitively, this approach captures the group-like structure in deformation and uses the training samples which are far away from the test image for prediction. Their approach shows good empirical results for local deformation, but fails to capture general deformation that includes both global and local components (e.g., cloth moving and deforming).

In this paper, we develop a top-down hierarchical structure for deformation estimation with global optimality guarantee. First, the deformation field is parameterized so that the deformation happening within a local image patch can be predicted by the content of that patch, reducing the dimensionality. Then, we model the nonlinear relationship between the image content and the deformation parameters using a novel Lipschitz criterion. With this criterion, all patches at different locations and scales can be regarded as predictors with guaranteed worst-case precisions. Finally, combining these predictors together in a top-down hierarchical manner leads to an overall predictor that can handle large and high-dimensional deformation with both local and global components.

Our contributions are threefold. *First*, our approach brings down sample complexity to $O(C_1^d + C_2 \log 1/\epsilon)$, which increases very slowly for higher accuracy. In particular, the number of samples required in each iteration stays constant for the first few iterations (layers of hierarchy), and then decays double exponentially (Fig. 1). Practically, our unoptimized Matlab implementation is fast, achieving 3–4 fps on real images. *Second*, compared to data-driven descent (Tian and Narasimhan 2012), our sample complexity guarantee is based on much weaker assumptions that can be verified with an efficient algorithm. As a result, our constant C_1 is much smaller than the constant C in Data-driven descent. Third, our work provides a rigorous theoretical analysis and interesting insights for top-down coarse-to-fine hierarchical structures. We believe that our analysis can be useful to analyze other similar hierarchies proposed in the computer vision community.

Our work not only has strong theoretical foundations, but also demonstrates good quantitative and qualitative results on real video sequences containing different types of deformation, including clothing and water surface deformations as well as medical images of internal organs. Our approach outperforms optimization-based approaches such as Lucas–Kanade (Lucas and Kanade 1981; Baker and Matthews 2004), Free-form registration (Rueckert et al. 1999) and PatchMatch (Barnes et al. 2009) (note all approaches are implemented in a coarse-to-fine manner), regression-based approaches such as nearest neighbor and explicit shape regression (Cao et al. 2012), feature-based approaches such as SIFT, (Lowe 2004), tracking-based approaches such as KLT (Lucas and Kanade 1981; Shi and Tomasi 1994), and hybrid methods such as data-driven descent.

Limitations. Unlike previous approaches (Beauchemin and Barron 1995; Barnes et al. 2009, 2010) that estimate non-rigid pixel correspondences from two images, our approach first requires training samples to build a hierarchical model, and then applies the trained model to other images to obtain pixel correspondence. However, the limitation can be overcome if we know the template image and its deformation model, from which the training samples can be generated. In

[1] Note that here the parameter norm $\|\cdot\|$ can be any norm, since if a certain norm is ϵ-small, so do others.

this case, the proposed algorithm can estimate the correspondences between two images. While the theory is applicable to 3D deformations with self occlusions, the generative model may be complex and require accurate rendering tools to synthesize training data.

2 Related Work

Optimization-based approaches (e.g., Baker and Matthews 2004; Matthews and Baker 2004; Rueckert et al. 1999) usually reach a local minimum using gradient descent or Newton's method. Random initialization is used to improve the quality of solutions on a heuristic basis. Regression-based approaches aim to learn a mapping from the distorted image to the deformation parameters using labeled training samples. The actual form of mapping could be nonparametric like nearest neighbor, or parametric like linear (Matthews and Baker 2004; Tan et al. 2014), random forest (Shotton et al. 2011), boosted random fern (Cao et al. 2012), etc. Feature-based approaches (e.g., SIFT, Lowe 2004) find correspondence by matching local features. Designing these local features needs a balance between feature distinctiveness and invariance under deformation.

Hierarchical structures have been used extensively in vision. Typical scenarios include coarse-to-fine optimization (Rueckert et al. 1999) for a better local solution, interest point detection (Lowe 2004), multi-resolutional feature extraction (Lazebnik et al. 2006), biologically plausible framework for object recognition (Serre et al. 2005) and so on. Recently, it is also used in deep learning, showing the state-of-art performance in image classification (Krizhevsky et al. 2012). However, as far as we know, none of the previous works provides theoretical performance guarantees for hierarchical structures.

Hierarchical optical flow (Beauchemin and Barron 1995) also adopts a top-down coarse-to-fine approach to estimate nonrigid deformation field. However, fundamentally, hierarchical optical flow builds the computational model based on non-convex optimization, and treats the coarse-to-fine approach as a practical heuristic that might help escape from local minima of the objective function with no theoretical guarantees. In this work, by introducing training samples and Lipschitz conditions, we show that the coarse-to-fine approach is more than a heuristic for the optimization, but has its own principles that deserve a different mathematical framework.

Similar to our work, PatchMatch (Barnes et al. 2009, 2010) also estimates a possibly nonrigid deformation field between two images. Initialized by a random deformation field that contains a few good correspondences with high probability, it propagates these good matches by nearest neighbor search over local translation (Barnes et al. 2009) or local scaling and rotation (Barnes et al. 2010). To overcome local solutions, a random search is also used to jump out of local optimal matches. These operations are done interleavingly in a coarse-to-fine manner: the top level correspondence is upsampled to be the initialization of lower level. Multiple random initializations are proposed to improve the quality of solution. Theoretical analysis has also been conducted in Barnes et al. (2009) for finding correspondence of a synthetic image pair, which has two distinctive regions located at different locations. The analysis shows that the expected number of random initializations used to extract the correct correspondences remains a small constant, if the distinctive region occupies a constant fraction of the image, and is independent of the size of image. From our point of view, if a large portion of the image is distinctive from its surroundings, then under deformation the image will undergo substantial change and Lipschitz condition naturally holds. Therefore, our conditions subsume the assumption made in the analysis (Barnes et al. 2009). In fact, our Lipschitz condition is much more general. For example, it can also be applied to synthetic examples containing two or more repetitive patterns deformed in a spatially correlated way, still achieving global convergence. On the other hand, the analysis in Barnes et al. (2009) would fail to show that PatchMatch can get the correct solution. PatchMatch may sample the correct correspondence in a few iterations, or may converge to the wrong but locally similar solution, even using a coarse-to-fine framework.

3 The Image Deformation Model

Denote T as the template image and $I_{\mathbf{p}}$ as the distorted image with deformation parameters $\mathbf{p}$. The deformation field $W(\mathbf{x};\mathbf{p})$ maps the pixel location $\mathbf{x}$ on the template to the pixel location $W(\mathbf{x};\mathbf{p})$ on the distorted image $I_{\mathbf{p}}$:

$$I_{\mathbf{p}}(W(\mathbf{x};\mathbf{p})) = T(\mathbf{x}) \tag{1}$$

Similar to data-driven descent (Tian and Narasimhan 2012), we parameterize the deformation field $W(\mathbf{x};\mathbf{p})$ by a linear combination of a set of bases:

$$W(\mathbf{x};\mathbf{p}) = \mathbf{x} + B(\mathbf{x})\mathbf{p} \tag{2}$$

where $B(\mathbf{x}) = [b_1(\mathbf{x}), b_2(\mathbf{x}), \ldots, b_K(\mathbf{x})]$ are K deformation bases and $\mathbf{p}$ are the coefficients. In general, $I_{\mathbf{p}}$ is *globally* related to the parameter $\mathbf{p}$, since a change of component k in $\mathbf{p}$ may propagate to the entire image via $b_k(\mathbf{x})$.

3.1 Local Over-Parameterization

In this paper, we consider a *local* parameterization of the deformation field $W(\mathbf{x};\mathbf{p})$ by making each basis a *sparse*

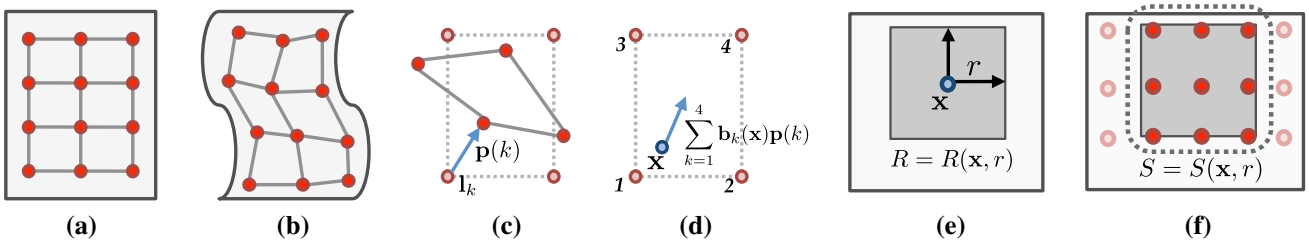

Fig. 2 Local parameterization of deformation. **a, b** The deformation field is controlled by a set of landmarks on the template image. By moving these landmarks, a deformed image is created. **c** Local parameterization. Each parameter $\mathbf{p}(i)$ encodes the 2D displacement of the landmark i. **d** Displacement on any pixel $\mathbf{x}$ is interpolated using displacements of nearby landmarks. **e, f** Image patch $I(R)$, where the image region is parameterized by location $\mathbf{x}$ and scale r, and its associated parameters $\mathbf{p}(S)$. In this paper, we study the nonlinear relationship between the two

vector. At any 2D point $\mathbf{x}$, its 2D deformation is determined by a weighted linear combination of the displacements of K landmarks (or control points), as shown in Fig. 2. Moreover, the farther away the landmark is from the query point $\mathbf{x}$, the smaller its weight. Formally, we write $\mathbf{p}(k)$ as the displacement of k-th landmark, and use a K-by-2 matrix $\mathbf{p} = [\mathbf{p}(1), \mathbf{p}(2), \ldots, \mathbf{p}(K)]^T$ to store $2K$ displacements.

From this setting, it is natural to see that the basis $b_k(\mathbf{x})$ is sparse and its energy is concentrated around the rest location $\mathbf{l}_k$ of landmark k. At any location $\mathbf{x}$, weights $b_k(\mathbf{x})$ are non-negative and normalized (i.e., $\sum_k b_k(\mathbf{x}) = 1$). Furthermore, at the rest location of k-th landmark location $\mathbf{l}_k$, $b_k(\mathbf{l}_k) = 1$ while $b_k(\mathbf{l}_{k'}) = 0$ for $k' \neq k$. In practice, $B(\mathbf{x})$ can be any interpolation function, e.g., Thin-plate Spline (Bookstein 1989), B-spline (Rueckert et al. 1999), local linear interpolation, etc.

In the case that we cover the image with a sufficient number of landmarks, due to strong positive correlations between nearby landmark displacements, the *actual* dimensionality d of the warping field could be much lower than $2K$. Similarly, if we select an image patch $I(R)$ which contains a subset S of landmarks, the local degrees d of freedom of a patch R are also smaller than $2|S|$, where $|S|$ is the number of landmarks within this region. We summarize this observation as follow:

Observation 1 *The local degrees of freedom of a patch are no more than* $\min(d, 2|S|)$.

An interesting question is, if the degrees of freedom of a patch are low, why is it necessary to use so many parameters? Because in such a case, the parameter estimation procedure can be *factorized* and achieves lower sample complexity. Many previous works (Tian and Narasimhan 2012; Matthews and Baker 2004; Salzmann et al. 2008) also assume a similar form of $W(\mathbf{x}; \mathbf{p})$. However, their parameters $\mathbf{p}$, usually given by dimensionality reduction procedures (e.g., PCA), is not localized to spatial landmarks. In comparison, Eq. 2 is both a localized and over-parameterization of the deformation field $W(\mathbf{x}; \mathbf{p})$, which leads to a further reduction of training samples needed.

3.2 Generating Training Samples

From Eq. 1, given the parameter $\mathbf{p}$, one can generate the deformed image $I_\mathbf{p}$ from the template T. This is done by assigning every pixel $\mathbf{y}$ of the deformed image $I_\mathbf{p}$ with the pixel value on location $\mathbf{x} = W^{-1}(\mathbf{y}; \mathbf{p})$ of the template T. Here W^{-1} is the inverse mapping from $\mathbf{y} = W(\mathbf{x}; \mathbf{p})$ back to $\mathbf{x}$ for a fixed $\mathbf{p}$. Note that due to the nonlinearity of $b_k(\mathbf{x})$, W^{-1} may not be representable by the linear model (Eq. 2), but as long as W^{-1} exists, samples can still be generated. Choosing different parameters $\{\mathbf{p}_i\}$ gives many *training samples* $\{(\mathbf{p}_i, I_{\mathbf{p}_i})\}$.

The major contribution of this paper, as described in the next sections, is to properly distribute the training samples and analyze the number of samples needed (i.e., *sample complexity*) to achieve the globally optimal prediction of the unknown parameters for a distorted test image.

4 The Relationship Between Image Evidence and Distortion Parameters

4.1 Lipschitz Conditions

Given any deformed image $I_\mathbf{p}$, to estimate $\mathbf{p}$ with theoretical guarantees, we need to assume a positive correlation between the image difference $\Delta I \equiv \|I_{\mathbf{p}_1} - I_{\mathbf{p}_2}\|$ and the parameter difference $\Delta\mathbf{p} \equiv \|\mathbf{p}_1 - \mathbf{p}_2\|_\infty$. Here ΔI is computed by a certain image metric, while $\Delta\mathbf{p}$ is computed in terms of maximal absolute difference between landmark displacements. Intuitively, if two images are close, so are their parameters and vice versa. Data-driven descent (Tian and Narasimhan 2012) characterizes such a relationship using global *Lipschitz condition* as below:

$$L_1 \Delta I \leq \Delta\mathbf{p} \leq L_2 \Delta I \tag{3}$$

where, L_1 and L_2 are two constants dependent on the template T. Tian and Narasimhan (2012) shows that the ratio of L_2/L_1 is a characteristic for samples complexity for guar-

anteed nearest neighbor prediction (see Eq. 4). For simple images that contain one salient object with a clear background, L_2/L_1 is typically small and a few samples suffice. For difficult images with repetitive patterns, L_2/L_1 is large and a lot of samples are needed to distinguish among locally similar-looking structures.

Note that in Tian and Narasimhan (2012), Eq. 3 is used to model the relationship between *global* image appearance and *global* parameters. However, Eq. 3 can be much broader. Both the parameters and the appearance may cover the entire image, or may only occupy a subregion of the image. In the following, we will show that the fact that local appearance changes with the local parameters will reduce the sample complexity.

Note that throughout the paper, the word "correlation" is used vaguely to represent the dependence between $\Delta\mathbf{p}$ and ΔI. It could take the form of Eq. 3, or take the form of relaxed Lipschitz condition (as will be discussed later). We define it in such a way to give provable bounds on the predicted accuracy if the training samples are sufficiently dense. At the same time, the conditions remain general and plausible in practice.

4.2 Sample Complexity Guarantee for Nearest Neighbor

Suppose we have training samples $\{\mathbf{p}^{(i)}, I_{\mathbf{p}}^{(i)}\}$ and want to predict the parameter for a test image I with an unknown true parameter $\mathbf{p}$. The simplest way is to use the nearest neighbor predictor: find $I_{\mathbf{p}'}^{(i)}$ in the training set that is closest to I, and return the parameter $\mathbf{p}'$ as the prediction $\hat{\mathbf{p}}$.

As a simple approach, nearest neighbor achieves guaranteed solution if we have a sufficient number of samples distributed uniformly in the space. Specifically, with the Lipschitz condition, it is possible to estimate sample complexity required for a prediction to be ϵ-close to the true value.

Theorem 1 (Sample Complexity for Nearest Neighbor) *If Eq. 3 holds, then with*

$$\left\lceil \frac{L_2}{L_1}\frac{r_0}{\epsilon} \right\rceil^d \tag{4}$$

number of training samples, for an image $I_{\mathbf{p}}$ generated from Eq. 1 with $\|\mathbf{p}\|_\infty \le r_0$, the predictor is able to make a prediction $\hat{\mathbf{p}}$ that is ϵ-close to the true parameter $\mathbf{p}$:

$$\|\hat{\mathbf{p}} - \mathbf{p}\|_\infty \le \epsilon \tag{5}$$

Proof By Lemma 4, we can uniformly sample the hypercube $[-r_0, r_0]^d$ in the parameter space so that for any $\|\mathbf{p}\|_\infty \le r_0$, there exists $(\mathbf{p}^{(i)}, I^{(i)})$ so that

$$\|\mathbf{p} - \mathbf{p}^{(i)}\|_\infty = \max_j |\mathbf{p}(j) - \mathbf{p}^{(i)}(j)| \le \frac{L_1}{L_2}\epsilon \tag{6}$$

From the same theorem, the number of samples needed is exactly Eq. 4. On the other hand, a nearest neighbor prediction $(\mathbf{p}_{NN}, I_{NN})$ can only be closer to the test $I_{\mathbf{p}}$ in the image space. Then by Lipschitz condition, we have:

$$\|I_{NN} - I_{\mathbf{p}}\| \le \|I^{(i)} - I_{\mathbf{p}}\| \le \frac{1}{L_1}\|\mathbf{p}^{(i)} - \mathbf{p}\|_\infty \le \frac{1}{L_2}\epsilon \tag{7}$$

Again use Lipschitz condition and notice $\mathbf{p}_{NN}$ is the output prediction, we have:

$$\|\hat{\mathbf{p}} - \mathbf{p}\|_\infty = \|\mathbf{p}_{NN} - \mathbf{p}\|_\infty \le L_2\|I_{NN} - I_{\mathbf{p}}\| \le \epsilon \tag{8}$$

□

4.3 Nyquist Limit

In fact, despite the context of image deformation, Eq. 3 can be used to characterize any input/output mapping and Theorem 1, as a sufficient condition for nearest neighbor to work theoretically, will still hold. Furthermore, without exploiting any domain-specific knowledge, it is likely that $O((1/\epsilon)^d)$ is the best we can do. A substantial reduction of training samples is impossible. The intuition is that the mapping $\mathbf{p} \mapsto I_{\mathbf{p}}$, although locally smooth as required by the Lipschitz condition, could be arbitrary over the long range in the high-dimensional space. To cope with such a flexibility, one needs to densely place the training samples at every location of the space, which naturally leads to the factor $(1/\epsilon)^d$. Therefore, we call $O((1/\epsilon)^d)$ the *Nyquist limit* since it reflects the lower bound of information needed to completely encode an arbitrary high-dimensional mapping, independent of specific prediction algorithms (e.g., nearest neighbor, boosting, etc.).

4.4 Limitations of Global Lipschitz Conditions

The entire analysis in data-driven descent relies on Theorem 1 and the global Lipschitz conditions. However, one shortcoming of the global Lipschitz condition

$$L_1\Delta I \le \Delta\mathbf{p} \le L_2\Delta I \quad \forall \mathbf{p}_1, \mathbf{p}_2 : \|\mathbf{p}_1\|_\infty, \|\mathbf{p}_2\|_\infty \le r_0 \tag{9}$$

is that it must hold for arbitrarily small ΔI and $\Delta\mathbf{p}$. Thus it (and also data-driven descent) fails in the following two situations:

- *Noisy images.* Adding noise to a distorted image I_p changes its appearance but not its parameters. As a result, $\Delta\mathbf{p} \approx 0$ but ΔI is finite. This makes $L_1 \to 0$.
- *Repetitive patterns.* If an image resembles itself after some transformation (e.g. translation/rotation), $\Delta\mathbf{p}$ is finite but $\Delta I \approx 0$. This makes $L_2 \to +\infty$.

Graphically these two cases are illustrated in Fig. 4d, where L_1 and L_2 are inverse slopes of the two boundaries

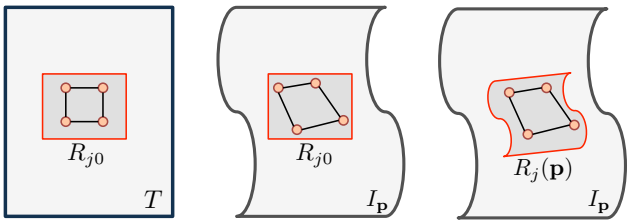

Fig. 3 Regions in template T and deformed image $I_{\mathbf{p}}$. **a** Template region R_{j0} of patch j on template image T. **b** R_{j0} on deformed image $I_{\mathbf{p}}$, the region is not changed while the deformed image changes with respect to $\mathbf{p}$. As a result, the image content $I_{\mathbf{p}}(R_{j0})$ also changes, from which the parameters $\mathbf{p}(S_j)$ can be estimated. **c** On the other hand, the region $R_j(\mathbf{p})$ moves in accordance with $\mathbf{p}$. The image content $I_{\mathbf{p}}(R_j(\mathbf{p}))$ equals $T(R_{j0})$ and remains constant (Eq. 1)

of the cone that covers all data. In both cases, the analysis in Theorem 1 gives a trivial (infinite) bound on sample complexity and global optimality cannot be guaranteed.

5 Relaxed Lipschitz Conditions

In this paper, we overcome these fundamental difficulties by introducing a novel and more general hierarchical framework. First, we introduce *relaxed Lipschitz conditions* that handle the two situations in which global conditions fail. Instead of using L_1 and L_2 to implicitly characterize ΔI-$\Delta\mathbf{p}$ relationship, we model it explicitly with the trade-off between sample density (the number of samples used per dimension) and convergence rate (the relative improvement of prediction upon the initial estimate) to build a guaranteed predictor. High sample density allows higher convergence rate but leads to high computation complexity. On the other hand, low sample rate makes prediction faster, but leads to low convergence rate.

Furthermore, the relaxed conditions are defined only on the patches and assumed to be valid for small deformation that does not move the patch too far away from its original locations. Since the assumptions are weaker than the global one, the conditions are more general.

Following these properties, predictors on large patches can operate on large deformation but produce coarse prediction, while predictors on small patches only work for small deformation but could lead to refined prediction. Therefore, we could stack the deformation predictors together in a coarse-to-fine hierarchy. Critically, the estimation of the coarse predictors push the patches of the fine predictors closer to their true locations, enabling the fine predictors to work properly. This establishes the "chain reaction" between successive layers and yields a much better parameter estimator that handles large deformation with precise prediction.

One main contribution of this paper, is to make these intuitions mathematically rigid and consistent. We will introduce each component of the framework in the following sections.

5.1 Formulation

To overcome the difficulties of the global Lipschitz condition, we introduce a patch-wise *relaxed Lipschitz condition* that characterizes the relation between the change of patch content $\Delta I_{\mathbf{p}}(R_{j0})$ and the subset of parameters $\Delta\mathbf{p}(S_j)$ of that patch. Intuitively, this condition says that similar parameters must yield similar appearance, while different appearance must come from different parameters. If we draw a scatter plot between ΔI and $\Delta\mathbf{p}$ (Fig. 4), then we see such a relationship could not grow too rapidly away from the origin but, after a certain point, must stay well above zero.

To make the intuition concrete, we first introduce some notations. As illustrated in Fig. 3, R_{j0} is the template region for j-th patch and is fixed when $\mathbf{p}$ changes, while $R_j(\mathbf{p})$ is the region that goes with $\mathbf{p}$. S_j is the subset of landmarks within the j-th patch (Fig. 2e), and r_j is the radius (or acceptance range) of patch. See Table. 1 for a notation overview. Then we formally define relaxed Lipschitz conditions independently at each patch j as follows:

Assumption 1 *(Relaxed Lipschitz condition for patch j)* For patch j with scale r_j and pull-back error η_j, there exists 4-tuples $(\alpha_j, \gamma_j, A_j, \Gamma_j)$ with $0 < \alpha_j \leq \gamma_j < 1$ and $A_j + 2\eta_j < \Gamma_j$ so that for any $\mathbf{p}_1$ and $\mathbf{p}_2$ with $\|\mathbf{p}_1\|_\infty \leq r_j$, $\|\mathbf{p}_2\|_\infty \leq r_j$, we have:

$$\Delta\mathbf{p} \leq \alpha_j r_j \implies \Delta I \leq A_j r_j \tag{10}$$

$$\Delta\mathbf{p} \geq \gamma_j r_j \implies \Delta I \geq \Gamma_j r_j \tag{11}$$

for parameter difference $\Delta\mathbf{p} \equiv \|\mathbf{p}_1(S_j) - \mathbf{p}_2(S_j)\|_\infty$ and image difference $\Delta I \equiv \|I_{\mathbf{p}_1}(R_{j0}) - I_{\mathbf{p}_2}(R_{j0})\|$.

Here $\|\mathbf{x}\|_\infty \equiv \max_i |x_i|$ is the max-norm (or L_∞ norm) of a vector. Note that similar to Tian and Narasimhan (2012), for those warps whose inversion cannot be parameterized by the linear model (Eq. 2), the pull-back error η, as an additional error term introduced to compensate the image quality loss when the image is warped to a less distorted version, is greater than 0. For the first read, *one could just treat $\eta = 0$ in all assumptions and theorems*. For a deeper understanding, we suggest the reader to consult the proof in Tian and Narasimhan (2012) and Appendix 2.

Graphical Illustrations. As illustrated in Fig. 4a, Eq. 10 says all $(\Delta\mathbf{p}, \Delta I)$ left to the vertical line $\alpha_j r_j$ must be below $A_j r_j$ (in the red-shaded box); while the second part says all $(\Delta\mathbf{p}, \Delta I)$ right to the vertical line $\gamma_j r_j$ must be above $\Gamma_j r_j$ (in the blue-shaded box). Finally, the condition $A_j + 2\eta_j < \Gamma_j$ suggests that the bottom of blue is

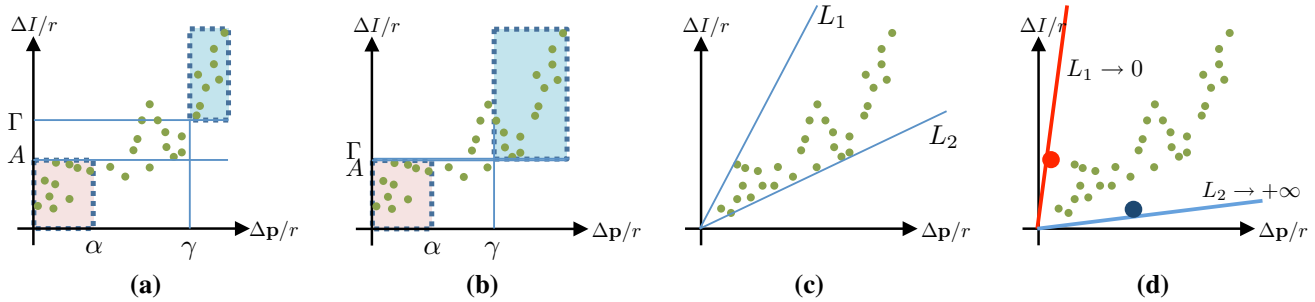

Fig. 4 Relaxed Lipschitz Condition (Eqs. 10 and 11). **a** There are four constants $(\alpha, \gamma, A, \Gamma)$ capturing the correlations between ΔI and $\Delta \mathbf{p}$ when $\Delta \mathbf{p}$ is small ($\leq \alpha r$) or large ($\geq \gamma r$). **b** Minimal γ without violating the condition. **c** Global Lipschitz condition (Eq. 9). The two constants L_1 and L_2 envelope the data points $(\Delta \mathbf{p}, \Delta I)$. **d** A noisy mapping ($\Delta \mathbf{p} = 0$ but $\Delta I \neq 0$) makes $L_1 = 0$ while an one-to-many mapping ($\Delta I = 0$ but $\Delta \mathbf{p} \neq 0$) causes $L_2 = +\infty$. In both cases, global Lipschitz condition fails while relaxed Lipschitz conditions are still valid with finite $(\alpha, \gamma, A, \Gamma)$

always above the top of red (by some margin). Intuitively, The red box enforces that similar parameters must yield similar appearances. Therefore, with sufficient (but finite) sampling we could cover all possible appearances. On the other hand, the blue box enforces two images similar in appearance might have similar parameters. This enables us to infer the parameters of an unknown image from the nearest neighbor in the appearance space.

Figure 5 gives more intuitions in the image/parameter space rather than in the delta space, when both $\mathbf{p}$ and $I_{\mathbf{p}}$ is one-dimensional. In Fig. 5c, all the blue rectangles depict that the vertical oscillation of the function should not exceed the upper/lower bound indicated by the rectangle. On the other hand, in Fig. 5d, the function must go up after a plateau indicated by the rectangle.

Trade-offs. From the definition, we can see that (α_j, γ_j) is not unique. In particular, if (α_j, γ_j) is a valid pair, so does (α'_j, γ'_j) for $\alpha'_j \leq \alpha, \gamma'_j \geq \gamma$. As we shall see in the next section, we want the smallest gap between α and γ. This can be achieved when A_j and Γ_j touch (Fig. 4b). For a given α_j, we define the *minimal*, or tightest γ_j given by the monotonic curve $\gamma_j = \gamma(\alpha_j)$, as shown in Fig. 6c.

The acceptance range r_j. Different from the Lipschitz conditions (Eq. 9), one important aspect of Eqs. 10 and 11 is that ΔI and $\Delta \mathbf{p}$ are only correlated up to the *acceptance range* r_j, i.e., $\|\mathbf{p}\|_\infty \leq r_j$. This *weaker* condition makes it possible to account for noise and parameter changes outside the subset S_j that may influence the patch $I_{\mathbf{p}}(R_{j0})$ without altering $\mathbf{p}(S_j)$. This also accounts for the case in which two slightly different parameters share the same image appearance. In both cases, the pair (α_j, γ_j) is still well-behaved while L_2/L_1 in Eq. 9 is not. Mathematically, r_j should be the maximal radius so that the relaxed Lipschitz condition is satisfied for $\|\mathbf{p}\| < r_j$. Practically we just choose r_j to be the radius of the patch (e.g., if a patch is of size a_1-by-a_2, then choose r_j as $\min(a_1, a_2)/2$), so that the Relaxed Lipschitz Conditions will in general hold and according to Theorem 2, the displacement can be predicted accurately.

5.2 Empirical Estimation of Lipschitz Constants

Empirically, the constants in the relaxed Lipschitz conditions (Eqs. 10 and 11) can be estimated from a set of image differences $\{\Delta I_m\}$ and corresponding parameter differences $\{\Delta \mathbf{p}_m\}$. Both differences can be computed from training samples $\{(\mathbf{p}^{(i)}, I^{(i)})\}$. Since γ, A, Γ is actually a function of α, what we estimate is a set of plausible 4-tuples, and most importantly, the curve $\gamma = \gamma(\alpha)$. Note that we omit subscript j here for clarity.

For M pairs of image and parameters differences $\{(\Delta \mathbf{p}_m, \Delta I_m)\}$, a brute-force search computes all plausible 4-tuples by enumerating all possible (α, γ) to find feasible ones. This takes $O(M^3)$ operations. Here we propose Algorithm 1 which only costs $O(M \log M)$. The algorithm is also illustrated in Fig. 6. Intuitively, we first sort the list of pairs $(\Delta \mathbf{p}, \Delta I)$ in ascending order with respect to $\Delta \mathbf{p}$, then we scan from the smallest $\Delta \mathbf{p}$. For each $\Delta \mathbf{p}_m$, we find the smallest (and tightest) sample index l^* that satisfies Relaxed Lipschitz Conditions. $\Delta \mathbf{p}_{l^*}$ thus gives the minimal γ achievable for $\alpha = \Delta \mathbf{p}_m / r_j$. Hence the curve $\gamma = \gamma(\alpha)$. A nice property of Algorithm 1 is that we only need to keep two pointers (m and l^*) and scan the array once per pointer. It is also an online algorithm: if the final curve is incomplete (e.g., γ is still very small for largest α), due to insufficient number of pairs, simply add more pairs with larger image distances and resume the algorithm from latest l^*. Please see Appendix 1 for the correctness proof.

5.3 Guaranteed Prediction by Nearest Neighbor

Now let us study how the relaxed Lipschitz condition helps nearest neighbor prediction. We wish to know how well patch $(\mathbf{x}, r)$ can predict the deformation $\mathbf{p}(S)$ within its acceptance

Table 1 Notations used in this paper

Images	
I_0	Template image
I	An arbitrary 2D Image
$\mathbf{x}, x, y$	2D pixel location, $\mathbf{x} = (x, y)$ is the vector form, while x and y are components.
Parameters	
K	The number of landmarks
S_j	A subset of landmarks. See Fig. 2(f). $\|S_j\|$ is the number of landmarks
$\mathbf{p}, \mathbf{q}, \hat{\mathbf{p}}, \tilde{\mathbf{p}}$	Deformation parameters as a K-by-2 matrix. $\mathbf{q}$ and $\hat{\mathbf{p}}$ are for training samples parameters (or their summation) (See Theorem 2)
$\mathbf{p}(j), \mathbf{p}^x(j), \mathbf{p}^y(j)$	The j-th landmark displacement (1-by-2 vector) in parameter $\mathbf{p}$. $\mathbf{p}^x(j)$ and $\mathbf{p}^y(j)$ are its components
$\mathbf{p}(S_j)$	A $\|S_j\|$-by-2 matrix that contains the landmark displacements in the subset S_j
$I_{\mathbf{p}}$	A deformed image with ground truth deformation parameter $\mathbf{p}$.
$W(\mathbf{x}; \mathbf{p})$	Deformation field parametrized by $\mathbf{p}$. Given $\mathbf{x}$ and $\mathbf{p}$, $W(\mathbf{x}; \mathbf{p})$ is a 1-by-2 vector
$W^{-1}(\mathbf{x}; \mathbf{p})$	Fixing $\mathbf{p}$, the inverse mapping from $\mathbf{y} = W(\mathbf{x}; \mathbf{p})$ back to $\mathbf{x}$. Not necessarily linear with respect to $\mathbf{p}$
$B(\mathbf{x})$	Deformation bases. $B(\mathbf{x}) = [b_1(\mathbf{x}), b_2(\mathbf{x}), \ldots, b_K(\mathbf{x})]$ where each column is a basis function.
d	Degrees of freedom.
Lipschitz conditions	
L_1, L_2	(Global) Lipschitz Constants proposed in Tian and Narasimhan (2012).
r_j	The radius of patch j
α	Inverse of sample density
γ	(One minus) convergence rate
A, Γ	Constants in Relaxed Lipschitz Conditions
ΔI	Scalar difference between two images I_1 and I_2, $\Delta I = \|I_1 - I_2\|$. $\|\cdot\|$ could be any norm
$\Delta\mathbf{p}$	Scalar difference between two parameters $\mathbf{p}_1$ and $\mathbf{p}_2$, $\Delta\mathbf{p} = \|\mathbf{p}_1 - \mathbf{p}_2\|_\infty$
Samples and algorithms	
N	Number of training samples
T	Total number of iterations
$(\mathbf{p}^{(i)}, I^{(i)})$	A training pair. A deformed image $I^{(i)}$ and its ground truth parameter $\mathbf{p}^{(i)}$
$\{\mathbf{p}^{(i)}, I^{(i)}\}_{i=1}^N$	A collection of training samples.
ϵ	Prediction error. Inverse of prediction accuracy
C	Constant terms in big-O notations
c_{SS}	Constant in Theorem 2 when modeling sampling complexity of deformation in a subspace. See Appendix 3
Miscellaneous	
$\lceil x \rceil$	Ceiling function, $\lceil x \rceil$ is the smallest integer that is greater than or equals to x
$\|\mathbf{p}\|_\infty$	Infinite norm. $\|\mathbf{p}\|_\infty = \max_i \|\mathbf{p}(i)\|_\infty = \max_i \max(\|\mathbf{p}^x(i)\|, \|\mathbf{p}^y(i)\|)$

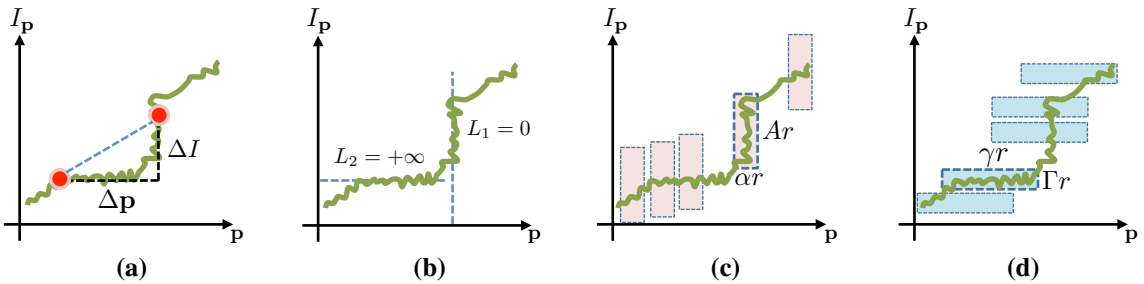

Fig. 5 More conceptual comparison in Lipschitz condition in one-dimensional image/parameters space (Note that Fig. 4 is in the delta space). **a** A (noisy) relationship between the parameters $\mathbf{p}$ and the deformed image $I_{\mathbf{p}}$. **b** Global Lipschitz condition (Eq. 9) cannot capture such relationship due to "plateaus" and "cliffs" in the relationship. On the other hand, relaxed Lipschitz condition is still valid, as long as **c** the cliff is not too tall or **d** the plateau does not extend too much

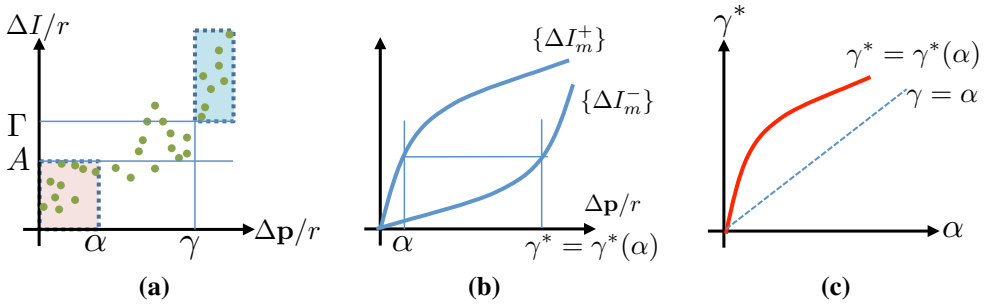

Fig. 6 Empirical estimation of constants in relaxed Lipschitz conditions. **a** Relaxed Lipschitz conditions and four constants $(\alpha, \gamma, A, \Gamma)$. **b** From the data point $(\Delta\mathbf{p}, \Delta I)$, in Algorithm 1, we compute two monotonously increasing function $\{I_m^+\}$ and $\{I_m^-\}$ (Note m is the discretized parameter difference $\Delta\mathbf{p}$). Given each α, we thus obtain its minimal γ^*. **c** The curve $\gamma^* = \gamma^*(\alpha)$, note that the curve is always on top of the straight line $\gamma = \alpha$

Algorithm 1 Find Local Lipschitz Constants

1: **INPUT** Parameter distances $\{\Delta\mathbf{p}_m\}$ with $\Delta\mathbf{p}_m \le \Delta\mathbf{p}_{m+1}$.
2: **INPUT** Image distances $\{\Delta I_m\}$.
3: **INPUT** Scale r and noise η.
4: $\Delta I_m^+ = \max_{1\le l\le m} \Delta I_l$, for $m = 1 \ldots M$.
5: $\Delta I_m^- = \min_{m\le l\le M} \Delta I_l$, for $m = 1 \ldots M$.
6: **for** $m = 1$ to M **do**
7: Find minimal $l^* = l^*(m)$ so that $\Delta I_{l^*}^- > \Delta I_m^+ + 2\eta$.
8: Store the 4-tuples:

$$(\alpha, \gamma, A, \Gamma) = (\Delta\mathbf{p}_m, \Delta\mathbf{p}_{l^*}, \Delta I_m^+, \Delta I_{l^*}^-)/r$$

9: **end for**

range r (i.e., $\|\mathbf{p}\|_\infty \le r$). Without any training samples, a trivial prediction $\hat{\mathbf{p}}(S) = 0$ yields a worst-case guaranteed prediction error of r. Now the problem is: if we want to obtain a slightly better prediction, how many training samples do we need?

Theorem 2 gives the answer. It shows that if the relaxed Lipschitz conditions (Eqs. 10 and 11) hold, then a Nearest Neighbor prediction with $1/\alpha$ samples per dimension will always reduce the error by a factor of $\gamma < 1$. Intuitively, we first fill the $2|S|$-dimensional hypercube $[-r, r]^{2|S|}$ with $(1/\alpha)^{2|S|}$ training samples uniformly. Then, for any test sample I within the hypercube, there is I' whose parameter difference is within αr. By one side of the Lipschitz condition (Eq. 10), $\|I - I'\| \le Ar$. The nearest neighbor of I, namely I_{NN}, is closer to I than I' to I. By the other side of the Lipschitz condition (Eq. 11), the parameter of I_{NN}, which is the prediction, is γr close to the true parameters of I.

Theorem 2 *(Guaranteed nearest neighbor) Suppose we have a distorted image I close to $I_{\mathbf{p}}$ on the region R_{j0}:*

$$\|I(R_{j0}) - I_{\mathbf{p}}(R_{j0})\| \le \eta_j r_j \tag{12}$$

If $\|\mathbf{p}\|_\infty \le r_j$, then with

$$N_j = \min\left(c_{ss}\lceil 1/\alpha\rceil^d, \lceil 1/\alpha\rceil^{2|S|}\right) \tag{13}$$

number of samples uniformly distributed in the hypercube $[-r_j, r_j]^{2|S|}$, we can compute a prediction $\hat{\mathbf{p}}(S_j)$ so that

$$\|\hat{\mathbf{p}}(S_j) - \mathbf{p}(S_j)\| \le \gamma_j r_j \tag{14}$$

using nearest neighbor prediction on the image region $I(R_{j0})$.

Proof Since $\|\mathbf{p}\|_\infty \le r_j$, by definition we have $\|\mathbf{p}(S_j)\|_\infty \le r_j$, that is, $\mathbf{p}(S_j)$ is within the hypercube of radius r_j. Then by applying Lemma 4 and Theorem 8 with $\alpha = \alpha_j$, if the number of samples needed follows Eq. 13, then there exists a training sample whose set of parameters $\mathbf{q}$ is also within the hypercube $\|\mathbf{q}(S_j)\|_\infty \le r_j$, and satisfies:

$$\|\mathbf{p}(S_j) - \mathbf{q}(S_j)\|_\infty \le \alpha_j r_j \tag{15}$$

For $k \notin S_j$, the value of $\mathbf{q}(k)$ is not important as long as $\|\mathbf{q}\|_\infty \le r_j$. This is because by assumption, the relaxed Lipschitz conditions still holds no matter how $\mathbf{q}(S_j)$ is extended to the entire landmark set.

Figure 7 shows the relationship for different quantities involved in the proof. Consider the patch $I_{\mathbf{p}}(R_{j0})$, using Eq. 10 and we have:

$$\|I_{\mathbf{p}}(R_{j0}) - I_{\mathbf{q}}(R_{j0})\| \le A_j r_j \tag{16}$$

Thus we have for the input image I:

$$\begin{aligned} &\|I(R_{j0}) - I_{\mathbf{q}}(R_{j0})\| \\ &\le \|I(R_{j0}) - I_{\mathbf{p}}(R_{j0})\| + \|I_{\mathbf{p}}(R_{j0}) - I_{\mathbf{q}}(R_{j0})\| \\ &\le (\eta_j + A_j) r_j \end{aligned} \tag{17}$$

On the other hand, since $I_{nn}(R_{j0})$ is the Nearest Neighbor image to $I(R_{j0})$, their distance can only be smaller:

$$\|I(R_{j0}) - I_{nn}(R_{j0})\| \le \|I(R_{j0}) - I_{\mathbf{q}}(R_{j0})\| \le (A_j + \eta_j) r_j \tag{18}$$

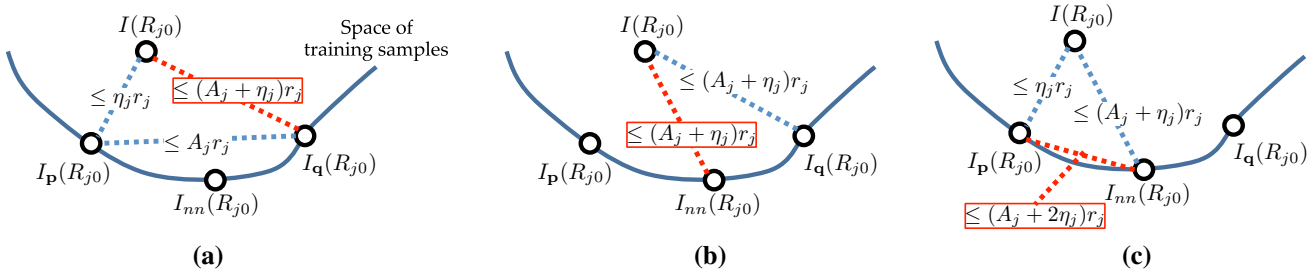

Fig. 7 Illustration of Theorem 2 and its proof. **a** Upper bound the appearance distance between query image $I(R_{j0})$ and image $I_{\mathbf{q}}(R_{j0})$ in the training set. **b** Since I_{nn} is the nearest neighbor in the training set, its distance is even smaller. **c** Upper bound the appearance distances between $I_{\mathbf{p}}(R_{j0})$ and $I_{nn}(R_{j0})$ to determine the precision of nearest neighbor prediction

Thus we have:

$$\|I_{\mathbf{p}}(R_{j0}) - I_{nn}(R_{j0})\| \tag{19}$$
$$\leq \|I_{\mathbf{p}}(R_{j0}) - I(R_{j0})\| + \|I(R_{j0}) - I_{nn}(R_{j0})\| \tag{20}$$
$$\leq (A_j + 2\eta_j)r_j \tag{21}$$

Now we want to prove $\|\mathbf{p}(S_j) - \mathbf{q}_{nn}(S_j)\| \leq \gamma_j r_j$. If not, then from the Lipschitz condition (Eq. 11) we have:

$$\|I_{\mathbf{p}}(R_{j0}) - I_{nn}(R_{j0})\| \geq \Gamma_j r_j > (A_j + 2\eta_j)r_j \tag{22}$$

which contradicts Eq. 21. Thus we have:

$$\|\mathbf{p}(S_j) - \mathbf{q}_{nn}(S_j)\|_\infty \leq \gamma_j r_j \tag{23}$$

Thus, just setting the prediction $\hat{\mathbf{p}}(S_j) = \mathbf{q}_{nn}(S_j)$ suffices. □

From Theorem 2, the exponent of Eq. 13 is the degrees of freedom mentioned in Obs. 1, which demonstrates the curse of dimensionality. Note that the gap η_j helps Theorem 2 work out when the preceived image patch $I(R_{j0})$ contains some noise, compared to the true distortion $I_{\mathbf{p}}(R_{j0})$. From Eqs. 13 and 14, now both α and γ have their physical meanings: α is the inverse of sample complexity per dimension, while γ is the inverse of prediction accuracy. Ideally we want a large α for lower sample complexity, and a small γ for higher accuracy. However, the constraint $\alpha \leq \gamma$ and the minimal curve $\gamma = \gamma(\alpha)$ show there is a trade-off. Like L_2/L_1 in Eq. 3, this trade-off reflects the difficulty level of images for deformation prediction (See Sect. 8 for empirical validation).

6 Construction of Hierarchy

6.1 Intuition

According to Theorem 2, different image patches $(\mathbf{x}, r)$ show different characteristics in their prediction guarantees: large patches (large r) can deal with large deformation but have low precision of prediction, while small patches (small r) only deal with small deformation but enjoy high precision. Therefore, in order to estimate large deformation with high precision, a natural way is to build a coarse-to-fine hierarchy of predictions as follows: the coarse layer (large patch) reduces the prediction residue by a certain extent so that it is within the acceptance range of the fine layer (small patch), where the prediction is refined.

6.2 The algorithm

Following this intuition, we construct the hierarchical structure as follows. Within the same layer t, scale of patches is fixed and denoted as r_t. Let $[t]$ contain all patches at layer t. When going from top to bottom (t becomes large), the scale r_t of patches shrinks towards zero. The shrinking factor $\bar{\gamma} = r_{t+1}/r_t$ is set to be

$$\bar{\gamma} \equiv \max_j \gamma_j < 1 \tag{24}$$

Algorithm 2 Hierarchical Deformation Estimation.

INPUT Training samples $Tr_j \equiv \{(\mathbf{p}^{(i)}, I^{(i)})\}$ for each image patch j.
INPUT Test image I_{test} with unknown parameters $\mathbf{p}$.
Set an initial estimation $\hat{\mathbf{p}}^0 = 0$.
for $t = 1$ to T **do**
  Set the current rectified image $I^t(\mathbf{x}) = I_{\text{test}}(W(\mathbf{x}; \hat{\mathbf{p}}^{t-1}))$.
  for Patch j within layer t **do**
    Find the Nearest Neighbor i^* for patch $I^t(R_{j0})$: $i^* = \arg\min_{i \in Tr_j} \|I^t(R_{j0}) - I^{(i)}(R_{j0})\|$
    Set the estimation $\tilde{\mathbf{p}}^t_j(S_j) = \mathbf{p}^{(i^*)}(S_j)$.
  end for
  Aggregation: $\tilde{\mathbf{p}}^t(i) = \text{mean}_{j:i \in S_j} \tilde{\mathbf{p}}^t_j(i)$.
  Update: $\hat{\mathbf{p}}^t(i) = \hat{\mathbf{p}}^{t-1}(i) + \tilde{\mathbf{p}}^t(i)$ for all landmarks.
end for
Return final predictions $\hat{\mathbf{p}}^T(i)$ for all landmarks.

Figure 8 and Algorithm 2 illustrate the algorithm that estimates the unknown parameter $\mathbf{p}$ given the test image I_{test}.

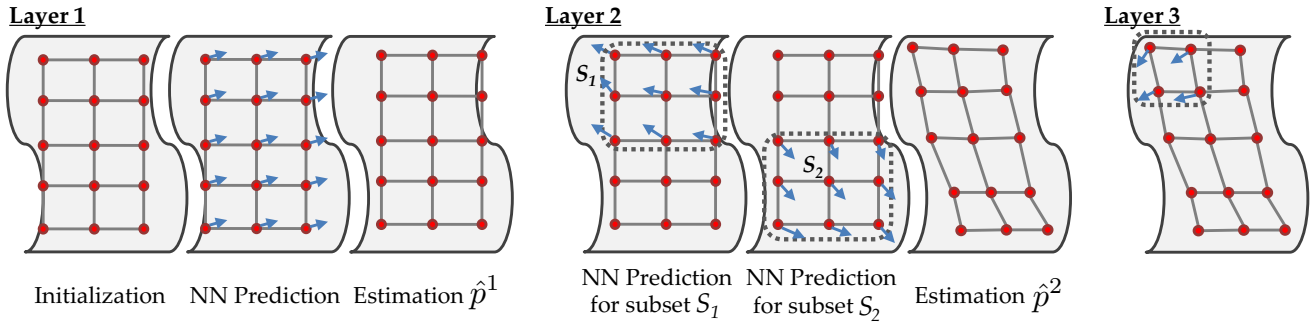

Fig. 8 Work flow of our hierarchical algorithm for deformation estimation. On *Layer 1*, a global prediction is made and the estimation is updated. On *Layer 2*, local deformation is estimated and aggregated. The procedure repeats until the last layer

For the first iteration, the test image I_{test} is directly compared with the training samples generated from the entire image with scale r_1 to obtain the nearest neighbor prediction $\hat{\mathbf{p}}^1$. Then for the second iteration, we have a slightly less distorted image $I_{\text{test}}(W(\mathbf{x}, \hat{\mathbf{p}}^1))$, from which we estimate $\mathbf{p}-\hat{\mathbf{p}}^1$. Since $\|\mathbf{p}-\hat{\mathbf{p}}^1\|_\infty$ is smaller than $\|\mathbf{p}\|_\infty$, its predictions can be localized to smaller patches. Then this procedure is iterated until the lowest layer. Similar to Tian and Narasimhan (2012), Algorithm 2 will converge to the globally optimal solution (Theorem 4), while the required number of samples is $O(C_1^d + C_2 \log 1/\epsilon)$ (Theorem 5).

Note that a less distorted image $I_{\text{test}}(W(\mathbf{x}; \hat{\mathbf{p}}^{t-1}))$, as the input of layer t, is not necessarily the same as a distorted image $I_{\mathbf{p}-\hat{\mathbf{p}}^{t-1}}$ generated directly from the template image. Their difference is the *pull-back error*. Fortunately, similar to Tian and Narasimhan (2012), the pull-back error decreases when $r_t \to 0$ and global convergence guarantee still holds. Theorem 7 summaries this property.

How much patch overlaps is not a critical part of the theoretical framework. However, in practice patch overlaping allows a slower reduction of patch size from top layer to bottom layer, making the entire framework workable in the case of larger γ. In our experiment, patches could be substantially overlapped (note that Fig. 8 is only for illustrative purpose).

Theorem 3 *For jth patch with template region R_{j0} and radius r_j, if $\|\mathbf{p}-\mathbf{q}\|_\infty \le r_j$ and $\|\mathbf{q}\|_\infty \le c_q$, then*

$$\|I_{\mathbf{p}}(R_j(\mathbf{q})) - I_{\mathbf{p}-\mathbf{q}}(R_{j0})\| \le \eta_j r_j \tag{25}$$

where $\eta_j = c_B c_q c_G Area_j$. Note $c_G = \max_{\mathbf{x}} |\nabla I_{\mathbf{p}}(\mathbf{x})|_1$. Note that R_{j0} is the template region while $R_j(\mathbf{q})$ is the deformed region (See Fig. 3).

Proof See Appendix 2. □

7 Analysis of the Algorithm

7.1 Global Convergence

As mentioned in Sect. 6.2, each iteration we reduce the error by $\bar{\gamma}$. Therefore, the convergence is a natural consequence from the intuition.

Theorem 4 (The global convergence theorem) *If $\|\mathbf{p}\|_\infty \le r_1$ and $r_1/(1-\bar{\gamma}) \le c_q$, (where c_q is defined in Theorem 3) then the prediction $\hat{\mathbf{p}}^t$ satisfies:*

$$\|\hat{\mathbf{p}}^t - \mathbf{p}\|_\infty \le \bar{\gamma}^t r_1 \tag{26}$$

As a result, the final prediction $\hat{\mathbf{p}}^T$ satisfies:

$$\|\hat{\mathbf{p}}^T - \mathbf{p}\|_\infty \le \bar{\gamma}^T r_1 \to 0 \tag{27}$$

for a sufficiently deep hierarchy whose number of layer $T \to +\infty$.

Proof First from the proof of Theorem 2, we can see the prediction $\tilde{\mathbf{p}}_j$ given by j-th patch satisfies $\|\tilde{\mathbf{p}}_j\|_\infty \le r_t$ for $j \in [t]$, since the prediction is picked from a hypercube $[-r_t, r_t]^{2|S_j|}$. Then, for the overall prediction $\hat{\mathbf{p}}^t$ at any layer t, we have:

$$\|\hat{\mathbf{p}}^t\|_\infty = \left\|\sum_{l=1}^{t-1} \tilde{\mathbf{p}}^l\right\|_\infty \le \sum_{l=1}^{t-1} \|\tilde{\mathbf{p}}^l\|_\infty \le \sum_{l=1}^{t-1} r_l \le \frac{r_1}{1-\bar{\gamma}} \le c_q \tag{28}$$

Then we prove the main result by induction. Suppose after layer t is processed, the residue $\delta\mathbf{p}^t \equiv \mathbf{p} - \hat{\mathbf{p}}^t$ satisfies:

$$\|\delta\mathbf{p}^t\|_\infty \le r_{t+1} \tag{29}$$

This is trivially true for $t = 0$ by the premise $\|\mathbf{p}\|_\infty \le r_1$ and initialization of the algorithm $\hat{\mathbf{p}}^0 = 0$. Now suppose Eq. 29 is correct for $t-1$, then:

– 1. By local pull-back inequality (Theorem 3), the bound of current estimation $\hat{\mathbf{p}}^t$ (Eq. 28) and $\|\delta\mathbf{p}^{t-1}\| \le r_t$, we have:

$$\begin{aligned}&\|I^t(R_{j0}) - I_{\delta\mathbf{p}^{t-1}}(R_{j0})\|\\ &= \|I_{\text{test}}(W(R_{j0}; \hat{\mathbf{p}}^{t-1})) - I_{\delta\mathbf{p}^{t-1}}(R_{j0})\|\\ &= \|I_{\text{test}}(R_j(\hat{\mathbf{p}}^{t-1})) - I_{\delta\mathbf{p}^{t-1}}(R_{j0})\|\\ &\le \eta_j r_t \end{aligned} \tag{30}$$

– 2. Since Eq. 30 together with the inductive hypothesis $\|\delta\mathbf{p}^{t-1}\|_\infty \le r_t$ satisfies the premise of Theorem 2, a prediction $\tilde{\mathbf{p}}_j(S_j)$ from j-th patch yields:

$$\|\tilde{\mathbf{p}}_j^t(S_j) - \delta\mathbf{p}^{t-1}(S_j)\|_\infty \le \gamma_j r_t \tag{31}$$

– 3. From Eq. 31, we know for every landmark k:

$$\|\tilde{\mathbf{p}}_j^t(k) - \delta\mathbf{p}^{t-1}(k)\|_\infty \le \gamma_j r_t \quad \forall k \in S_j \tag{32}$$

Then for every landmark k, the averaged prediction $\tilde{\mathbf{p}}(k)$ over overlapping patch set $S[k] = \{j : j \in [t], k \in S_j\}$ can also be bounded:

$$\begin{aligned}&\|\tilde{\mathbf{p}}^t(k) - \delta\mathbf{p}^{t-1}(k)\|_\infty\\ &= \left\| \left(\frac{1}{\#S[k]} \sum_{j\in S[k]} \hat{\mathbf{p}}_j^t(k) \right) - \delta\mathbf{p}^{t-1}(k) \right\|_\infty\\ &\le \frac{1}{\#S[k]} \sum_{j\in S[k]} \|\hat{\mathbf{p}}_j^t(k) - \delta\mathbf{p}^{t-1}(k)\|_\infty\\ &\le \frac{1}{\#S[k]} \sum_{j\in S[k]} \gamma_j r_t \le \bar{\gamma} r_t \end{aligned}$$

– 4. Finally, the residue $\delta\mathbf{p}^t$ after adding prediction $\tilde{\mathbf{p}}$ of layer t satisfy:

$$\|\delta\mathbf{p}^t\|_\infty = \|\delta\mathbf{p}^{t-1} - \tilde{\mathbf{p}}\|_\infty \le \bar{\gamma} r_t = r_{t+1} = \bar{\gamma}^t r_1 \tag{33}$$

Therefore, Eq. 29 also holds for t. □

7.2 The Number of Samples Needed

A natural question is, to achieve the condition of Theorem 4, how many samples are needed? Since the areas of patches, as well as the number of landmarks $|S|$, decrease exponentially by a factor of $\bar{\gamma}^2$ from top to bottom, the number of samples needed, exponential to the degrees of freedom $\min(2|S|, d)$ by Observation 1, stays the same until $2|S| \approx d$, and then goes down *double-exponentially*. Therefore, by adding them up, the top level samples dominate. The following theorem makes the intuition rigid:

Theorem 5 (The number of samples needed) *The total number N of samples needed is bounded by:*

$$N \le C_3 C_1^d + C_2 \log_{1/\bar{\gamma}} 1/\epsilon \tag{34}$$

where $C_1 = 1/\min_j \alpha_j$, $C_2 = 2^{1/(1-\bar{\gamma}^2)}$ *and* $C_3 = 2 + c_{SS}(\lceil \frac{1}{2} \log_{1/\bar{\gamma}} 2|S|/d \rceil + 1)$. *See Appendix 3 for the definition of* c_{ss}.

Proof We divide our analysis into two cases: $d = 2|S|$ and $d < 2|S|$, where $|S|$ is the number of landmarks. $d > 2|S|$ is not possible because of over-parameterization.

Case 1: $d = 2|S|$

First let us consider the case that the intrinsic dimensionality of the deformation field d is just $2|S|$. Then the root dimensionality $d_1 = 2|S_1|$ (twice the number of landmarks). By Observation 1, the dimensionality d_t for layer t is proportional to $2|S_t|$ and thus:

$$d_t = \frac{d_1}{r_1^2} r_t^2 = \bar{\gamma}^{2t-2} d_1 \tag{35}$$

since $|S_j|$ goes down with r_t^2. Let $[t]$ be the set of all patch indices in layer t. Any patch $j \in [t]$ has the same degrees of freedom since by Observation 1, d_j only depends on $|S_j|$, which depends on r_j, a constant over layer t.

For any patch $j \in [t]$, from Theorem 2, N_j training samples suffice to ensure the contracting factor is indeed at least $\gamma_j \le \bar{\gamma}$:

$$N_j \le \left(\frac{1}{\alpha_j} \right)^{d_t} \tag{36}$$

Note for patch j, we only need the content within the region R_{j0} as the training samples. Therefore, training samples of different patches in this layer can be stitched together, yielding samples that cover the entire image. For this reason, the number N_t of training samples required for the layer t is:

$$N_t \le \arg\max_{j\in[t]} N_j \le C_1^{d_t} = C_1^{\bar{\gamma}^{2t-2} d_1} \tag{37}$$

for $C_1 = 1/\min_j \alpha_j$. Denote $n_t = C_1^{\bar{\gamma}^{2t-2} d_1}$. Then we have:

$$N \le \sum_{t=1}^{T} N_t \le \sum_{t=1}^{T} n_t \tag{38}$$

To bound this, just cut the summation into half. First, n_t is a decreasing function with respect to t; second, the ratio between n_t and n_{t+1} has the following simple form:

$$\frac{n_t}{n_{t+1}} = \frac{C_1^{\bar{\gamma}^{2t-2} d_1}}{C_1^{\bar{\gamma}^{2t} d_1}} = C_1^{\bar{\gamma}^{2t-2}(1-\bar{\gamma}^2) d_1} = n_t^{1-\bar{\gamma}^2} \tag{39}$$

Given $l > 1$, set T_0 so that

$$\frac{n_{T_0}}{n_{T_0+1}} = n_{T_0}^{1-\bar{\gamma}^2} \geq l, \quad \frac{n_{T_0+1}}{n_{T_0+2}} = n_{T_0+1}^{1-\bar{\gamma}^2} \leq l \tag{40}$$

This means that before T_0, n_t decreases faster than $(1/l)^t$, while after T_0, n_t decreases slower. Thus if we break the summation into two terms:

$$\sum_{t=1}^{T} n_t = \sum_{t=1}^{T_0} n_t + \sum_{t=T_0+1}^{T} n_t \tag{41}$$

Then the first is bounded by a geometric series:

$$\sum_{t=1}^{T_0} n_t \leq C_1^{d_1} \sum_{t=1}^{T_0} \left(\frac{1}{l}\right)^{t-1} \leq \frac{C_1^{d_1}}{1-1/l} = \frac{l}{l-1} C_1^{d_1} \tag{42}$$

On the other hand, since $n_{T_0+1}^{1-\bar{\gamma}^2} \leq l$, each item of the second summation is less than $l^{1/(1-\bar{\gamma}^2)}$. We thus have:

$$\sum_{t=T_0+1}^{T} n_t \leq l^{1/(1-\bar{\gamma}^2)} T \tag{43}$$

Combining the two, we then have:

$$N \leq \frac{l}{l-1} C_1^{d_1} + l^{\frac{1}{1-\bar{\gamma}^2}} T \tag{44}$$

for $T = \lceil \log_{1/\bar{\gamma}} 1/\epsilon \rceil$. Note this bound holds for any l, e.g. 2. In this case, we have

$$N \leq 2C_1^{d_1} + C_2 T \tag{45}$$

for $C_2 = 2^{\frac{1}{1-\bar{\gamma}^2}}$.

Case 2: $d < 2|S|$

In this case, setting $d_1 = 2|S|$, finding T_1 so that $d_{T_1} \geq d$ but $d_{T_1+1} < d$ in Eq. 35, yielding:

$$T_1 = \left\lceil \frac{1}{2} \log_{1/\bar{\gamma}} 2|S|/d \right\rceil + 1 \tag{46}$$

Then, by Observation 1, from layer 1 to layer T_1, their dimensionality is at most d. For any layer between 1 and T_1, N_t is bounded by a constant number:

$$N_t \leq c_{ss} C_1^d \tag{47}$$

The analysis of the layers from T_1 to T follow Case 1, except that we have d as the starting dimension rather than $2|S|$. Thus, from Eq. 45, the total number of samples needed is:

$$N \leq (T_1 c_{ss} + 2) C_1^d + C_2 T \tag{48}$$

□

8 Empirical Upper Bounds for Images

Given a specific template and a specific family of deformation, we can generate many deformed images and their parameters $(\mathbf{p}_i, I_{\mathbf{p}_i})$, compute all-pair image/parameter distances $\{\Delta\mathbf{p}_i, \Delta I_i\}$ and estimate the monotonous curve $\gamma = \gamma(\alpha)$ like Fig. 4. This curve can help predict the theoretical difficulties of images for deformation estimation. For simplicity, we set a constant and convergent contraction factor $\bar{\gamma} = 0.95$ and compute the largest $\alpha_{0.95} = \gamma^{-1}(0.95)$. Therefore, simple images have high $\alpha_{0.95}$, indicating low sample complexity per dimension $(1/\alpha_{0.95})$, and vice versa.

We randomly generate 1000 deformed samples and compute all-pair distances. The deformation is 2D translation and in-plane rotation ($d = 3$) up to $\pm\pi/8$. We propose Algorithm 1 which only costs $O(M \log M)$ to compute the curve $\gamma = \gamma(\alpha)$, while brute-force search costs $O(M^3)$. See Appendix 1 for correctness proof.

Figure. 9 shows each template and its $N_1 \equiv 1/\alpha_{0.95}$. We can clearly see that images with a salient object and uniform background requires fewer samples, while images with repetitive patterns and cluttered backgrounds require more. In contrast, $N_2 \equiv L_2/\gamma L_1$, as suggested in Tian and Narasimhan (2012), is much higher in both cases. Note that N_1 and N_2 are comparable quantities. They both specify the number of samples needed per dimension for nearest neighbor to reduce the prediction error by $\bar{\gamma}$.

With regard to the total sample complexity N, Theorem 2 states that for easy images, $N_1 = 1/\alpha_{0.95} \approx 5$ and $N \approx [5 \cdot (2+\sqrt{2})]^4 = 84926$ (See Appendix 3 for details), while for hard images, $1/\alpha_{0.95} \approx 12$ and $N \approx [12 \cdot (2+\sqrt{2})]^4 = 2817654$. Although in practice N may be much smaller, the theoretical upper bound gives a sense of difficulty levels of images.

9 Experiments on Simulated Data

9.1 Degree of Freedom of a Patch

We start with verification of Observation 1 that specifies the relationship between the size of patch and the deformation degrees of freedom. Figure 10 shows that deformation dimensionality is indeed quadratically related to the patch size, coincides with the observation. Here the deformation field is generated from multi-variate Gaussian distribution with correlation matrix $\Sigma_{\mathbf{x},\mathbf{x}'}$ as follows:

$$\Sigma_{\mathbf{x},\mathbf{x}'} = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}\|^2}{2\sigma^2}\right) \tag{49}$$

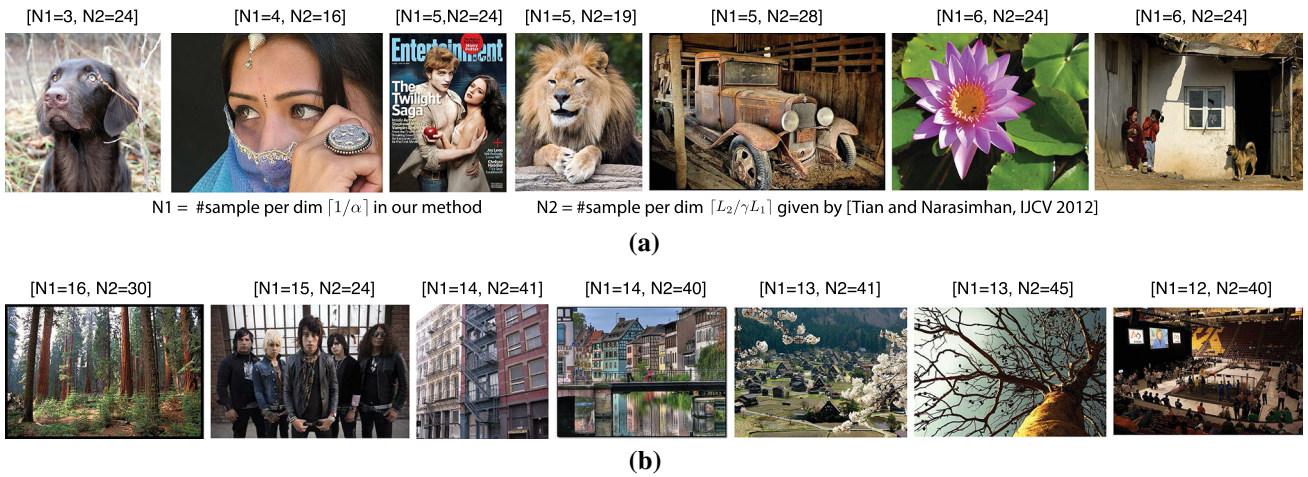

Fig. 9 Exemplar images and the theoretical bounds for the number of samples needed per dimension. For each bracket, $N_1 = 1/\alpha_{0.95}$ is our sample complexity bound per dimension, while $N_2 = L_2/\bar{\gamma} L_1$ is from data-driven descent ($\bar{\gamma} = 0.95$). N_1 and N_2 are comparable quantities that both specify the number of samples required per dimension for nearest neighbor to reduce the prediction error by $\bar{\gamma}$. *Top row* Images with a salient object and clean background require only a few samples per dimension. *Bottom row* Images with repetitive patterns require more samples per dimension. In both cases, our bound is smaller than that given by data-driven descent

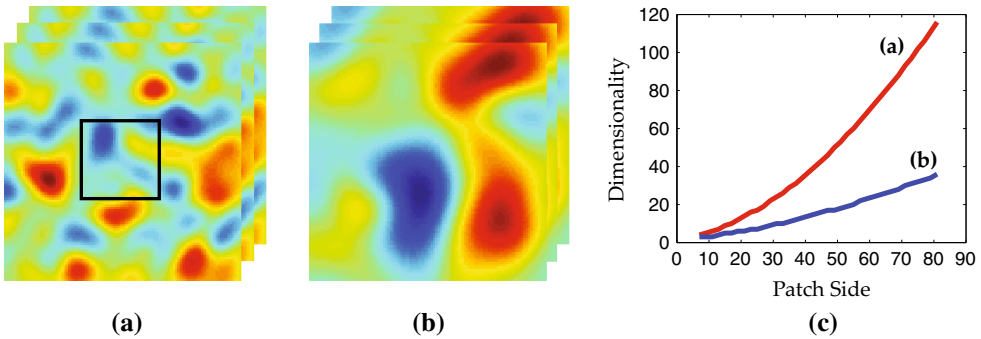

Fig. 10 Verification of the quadratic relationship between patch size and dimensionality of the deformation field $W(\mathbf{x}) - \mathbf{x}$. **a, b** Weakly and strongly correlated deformation field, generated by Gaussian distribution with spatially correlated covariance matrices. **c** The dimension of deformation field within the patch indeed grows quadratically with respect to the size of patch

where σ specifies the smoothness of the deformation field (larger σ gives spatially smoother deformation). To estimate dimensionality, we perform a SVD on 1000 patches of deformation field, and find the number of dimensions whose singular values sum up to 90 % of the total summation.

Figure 10 shows that a weakly-coupled deformation field has higher dimensionality, while a strongly-coupled one has lower dimensionality. In both cases, the dimensionality grows quadratically with respect to the patch side, or linearly with respect to the patch area.

9.2 Convergence Behavior

Now we show our algorithm works well for synthetic data. Our approach adopts a hierarchical structure using a grid of 256 landmarks with $\bar{\gamma} = 0.7$ and $T = 8$ layers. We start from using the entire image for prediction and then reduce the patch size by $\bar{\gamma}$ for the next layer. Once the patch size is reduced, we thus find the subset S_j of landmarks that are inside the patches. This procedure is performed recursively until each patch only covers a single landmark. Once the patch arrangements are done, training follows. Since we use nearest neighbor approach, training is essentially to extract deformed patch samples from the template. We use global affine transform plus Thin-Plate Spline (Bookstein 1989) as bases functions with proper normalization. While our theory gives an upper bound of the sample complexity, practically 350 training samples over all layers suffice for good performance.

We artificially distorted 100 images with a 20-dimensional global warping field specified in Tian and Narasimhan (2012). For each image, its 10 distorted versions are generated with groundtruth random parameters. We thus compare the estimation from Tian and Narasimhan (2012) and from our approach based on groundtruth.

Figure 11 shows the performance comparison. Our algorithm obtains much better performance and lower variance compared to DDD with the same number of training samples. Note that the strong drop in error shows that our method

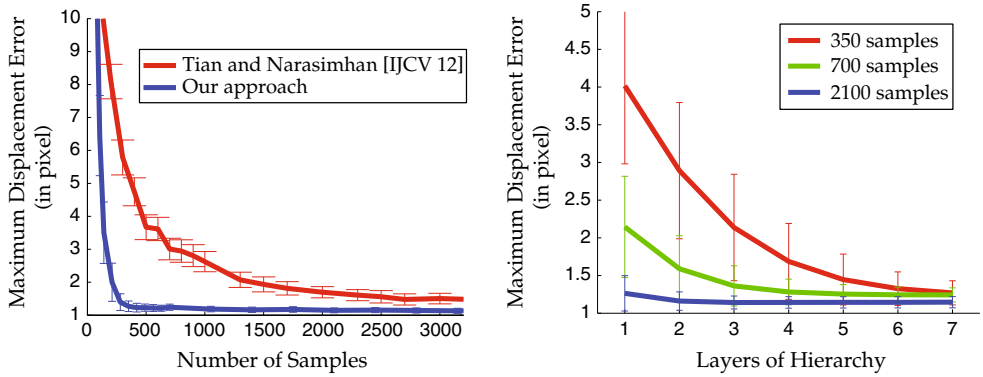

Fig. 11 Performance of the proposed algorithm. *Left* Performance comparison with data-driven descent (Tian and Narasimhan 2012). Accuracy of our approach increases much faster than Tian and Narasimhan (2012) with the same number of samples. To obtain the same level of accuracy of our approach with 400 samples, (Tian and Narasimhan 2012) requires 10,000 samples or more. Our approach also has lower variance in performance. *Right* Convergence behavior of our approach with different number of training samples

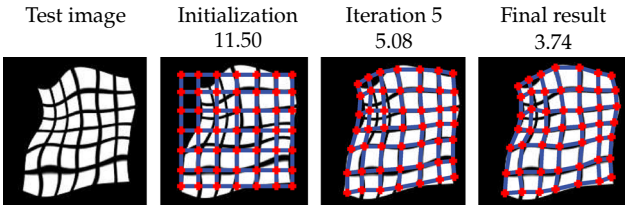

Fig. 12 Demonstration of the iterative procedure of our algorithm. Starting from initialization, the algorithm applies predictors of different layers to estimate the landmark locations. Numbers on top show RMS errors

Table 2 Performance comparison of different approaches, including Lucas–Kanade (LK) (Lucas and Kanade 1981; Baker and Matthews 2004), data-driven descent (TN) (Tian and Narasimhan 2012), Free-form registration (FF) (Rueckert et al. 1999), Explicit Shape Regression (ESR) (Cao et al. 2012) and SIFT matching with outlier removal using RANSAC (SR) (Lowe 2004).

	LK	TN	ESR	FF	SR	Ours
RMS	14.79	6.44	8.98	7.29	98.94	**5.63**
Sec/frame	11	77	**0.012**	35	1.25	0.10

Ours is the best performer and second best in time cost per frame

achieves high accuracy by adding very few samples once it starts to work. This coincides with our sample complexity $O(C_1^d + C_2 \log 1/\epsilon)$ and Fig. 1.

9.3 Deformation Estimation on Repetitive Patterns

We further test our approach on synthetic data containing distorted repetitive patterns (Fig. 12), and compare it with previous methods. From an undistorted template (240-by-240), we generate a dataset of 200 distorted images, each with labeled 49 points. The deformation field is created by random Gaussian noise without temporal continuity.

The overall degree of freedom for this dataset is very high (50 dimensions are needed to achieve <1 pixel reconstruction error). It is in general impossible to have a sufficient number of samples for global optimality conditions to be satisfied. However, practically our method still works well.

We compare our approach to the following previous methods: Lucas–Kanade (LK) (Lucas and Kanade 1981; Baker and Matthews 2004), data-driven descent (DDD) (Tian and Narasimhan 2012), free-form registration (FF) (Rueckert et al. 1999; Tan et al. 2014), explicit shape regression (ESR) (Cao et al. 2012) and SIFT matching with outlier removal using RANSAC (SR) (Lowe 2004). LK and DDD use a locally parametric deformation model. LK uses local affine bases of size 100-by-100, and DDD uses a 20-dimensional smooth bases of size 57-by-40 (Tian and Narasimhan 2012). LK, FF and DDD compute dense deformations. Our hierarchy outputs 256 predicted landmarks on a regular grid. In all cases, estimated deformations are interpolated into 49 landmark locations for a fair comparison. The KLT tracker (Lucas and Kanade 1981; Shi and Tomasi 1994) requires temporal information and will be compared in the real video sequence.

For one image, the RMS error is computed between the estimated landmark locations $\hat{\mathbf{p}}$ and groundtruth locations $\mathbf{p}$ as $RMS = \sqrt{\frac{1}{K}\sum_{i=1}^{K} \|\mathbf{p}(i) - \hat{\mathbf{p}}(i)\|^2}$. For multiple images, averaged RMS is reported.

Table 2 compares the performance. Due to repetitive patterns, previous approaches fail to estimate the landmarks correctly. SIFT matching fails completely. The prediction of ESR is restricted to be on the linear shape subspace spanned by the training samples. Thus, it is insufficient to use the template to capture the subspace of a complex deformation field. LK and FF are stuck in local maxima despite their coarse-to-fine implementations. Our approach obtains the best performance. Figure 12 shows the progression of our algorithm. In terms of speed, our approach is the second best only to ESR, which uses a fast boosting framework.

Table 3 Performance on synthetic data if the first L layer of predictors are switched off, showing the bottom layers play a critical role for performance

L	0	1	2	3	4	5	6
RMS	5.63	5.20	5.14	5.83	6.72	7.95	8.74

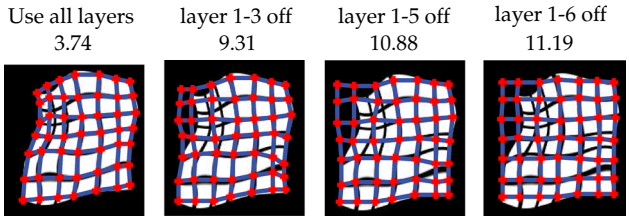

Fig. 13 Performance changes if the first K layers are switched off. When more layers are switched off, the algorithm is unable to identify global deformation and is essentially the same as local template matching at each landmark

9.4 Influence of Multiple Layers

It is interesting to see how the performance changes if we switch off the first L layers of predictors. As shown in Table 3, the first two layers have less contribution on the performance than the rest of the layers. On the other hand, the lower 6 layers indeed help the performance. Figure 13 demonstrates how prediction from coarse layers (large patch) help the lower layer (small patch) find correct correspondences in repetitive patterns, justifying the hierarchy.

10 Real Experiments

We also apply our framework to real world scenarios such as water distortion, cloth deformation and registration of medical images. For real experiments, we still use $\bar{\gamma} = 0.7$ and $T = 8$. However, the number of landmarks might be different for different video sequences, depending on their dimensions and aspect ratios. In Fig. 14, contour tracking is achieved by interpolating contour points from frame correspondences, while the contour of the first frame is manually labeled. In Fig. 15, the tracked mesh is shown.

The three water distortion sequences (Row 1–2 in Fig. 14, Row 1 in Fig. 15) and one cloth sequence (Row 3 in Fig. 14) are from Tian and Narasimhan (2012). Two cloth sequences (Row 2–3 in Fig. 15) are from Taylor et al. (2010) and Moll and Gool (2012). The medical sequence of cardiac magnetic resonance images (4th row in Fig. 14) is from Zhang et al. (2012). We also captured our own cloth sequence (Fig. 14), 5th row.

For the sequences on the 4th row of Fig. 14 and the 1st row of Fig. 15, we use temporal information by adding training samples generated from estimation of the previous frame. These additional training samples help capture the drifting appearance of the object/scene over time. This procedure slows down the processing to 0.3–0.5 fps, yet is still faster than previous approaches. For other sequences, our algorithm runs at around 3–4 fps. Note that our method successfully estimates the deformations. In comparison, SIFT with RANSAC only obtains a sparse set of distinctive matches, not enough for estimating a nonrigid deformation (even if we are using Thin-Plate Spline). Data-driven descent can capture detailed local deformations but not global shifts of the cloth without modeling the relationship between local patches. KLT trackers lose the target quickly and localize contour inaccurately.

We also quantitatively measure the landmark localization error using the densely labeled dataset provided in Tian and Narasimhan (2012), which contains 30 labeled frames, each with 232 landmarks. In terms of RMS, LK gives 5.20, FF gives 3.93, DDD gives 2.51 while our approach gives 3.29. Our framework is only second to DDD, which is much slower. The reason why the proposed approach is worse than DDD in this particular dataset is three-fold. First, the groundtruth landmarks of this dataset are not on a regular grid (but on the corners), so we first place a regular grid, estimate the deformation field on the grid and then interpolate back to the groundtruth landmarks. As a result, interpolation errors may be introduced. Also, since the datasets are text underwater, there are quite a few uniform regions and landmarks on the regular grid might not correspond to any locally distinctive features. Finally, we stop the construction of hierarchy when each patch only covers one landmark. Although it is good for landmark prediction, local deformation happening close to the landmarks might still not be estimated accurately.

We have tested our algorithm on existing datasets of deformable objects proposed by Salzmann et al. (2007, 2008). Although no ground-truth is available, our performance is close to their published results. For example, we achieve 4.10 mean pixel distance difference in cushion video (Salzmann et al. 2007) and 4.43 in bed-sheet video (Salzmann et al. 2008). All video sequences are 404-by-504.

10.1 A Comparison with PatchMatch

We also compare our approach with PatchMatch (Barnes et al. 2009) using the official coarse-to-fine implementation[2]. PatchMatch is a randomized algorithm that finds dense correspondences between two images. Starting with a random guess, in each iteration correspondences are propagated to local neighbors until convergence. The intuition is that in each random guess, with high probability a few matches are

[2] Please check http://gfx.cs.princeton.edu/pubs/Barnes_2009_PAR/index.php for their source code.

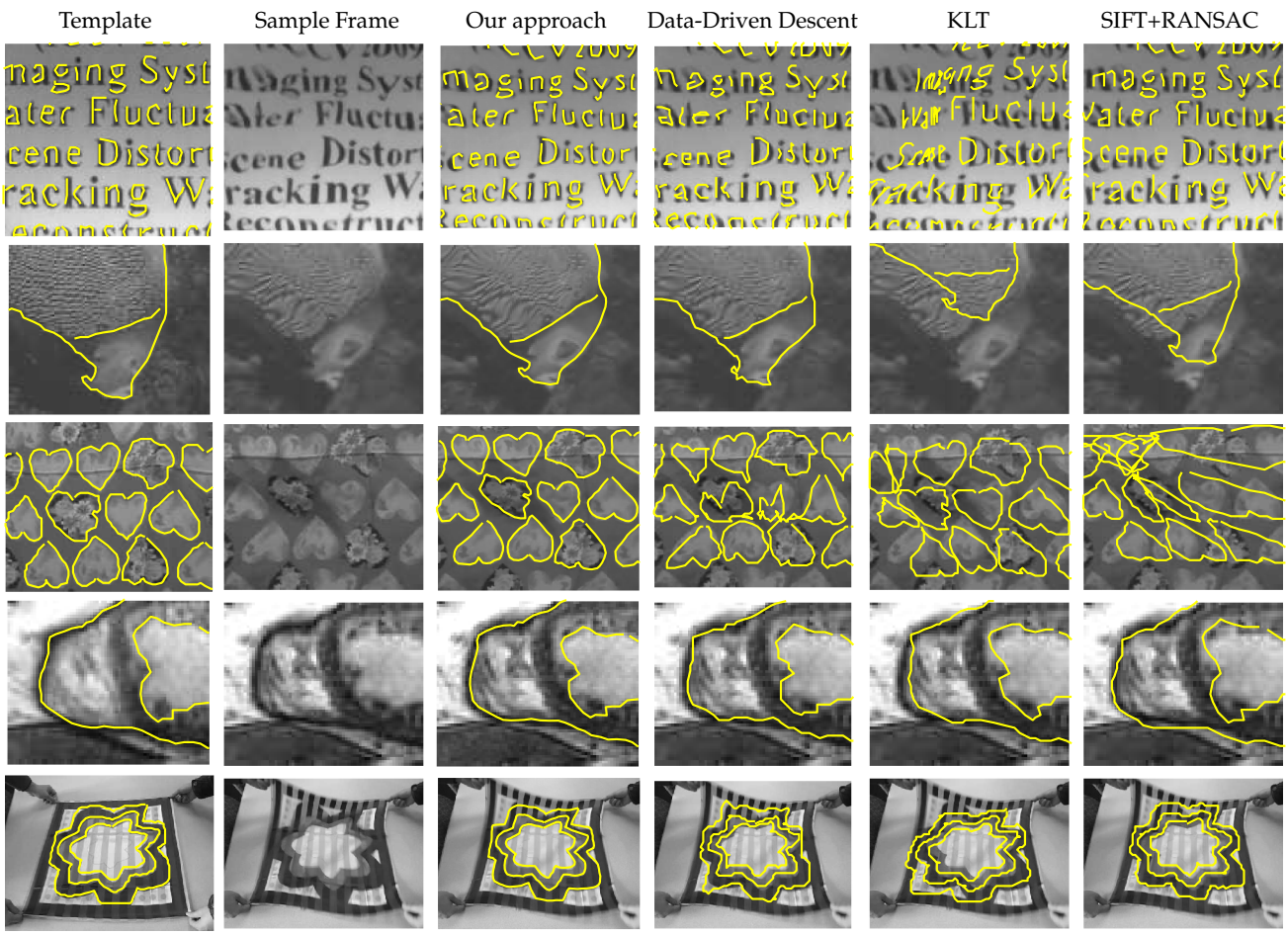

Fig. 14 Example contour localization results given by our approach, data-driven descent (Tian and Narasimhan 2012), KLT (Lucas and Kanade 1981; Shi and Tomasi 1994), and SIFT matching with RANSAC (Lowe 2004). Each *row* is a video sequence, two from underwater imaging, two from cloth deformation and the final one is from medical imaging. For each dataset, one sample frame is shown. The contours are drawn manually for the template image (*1st column*), and are transferred to every video frame after the correspondence was found. Our approach is stable and better than other approaches. (Best viewed in *color*)

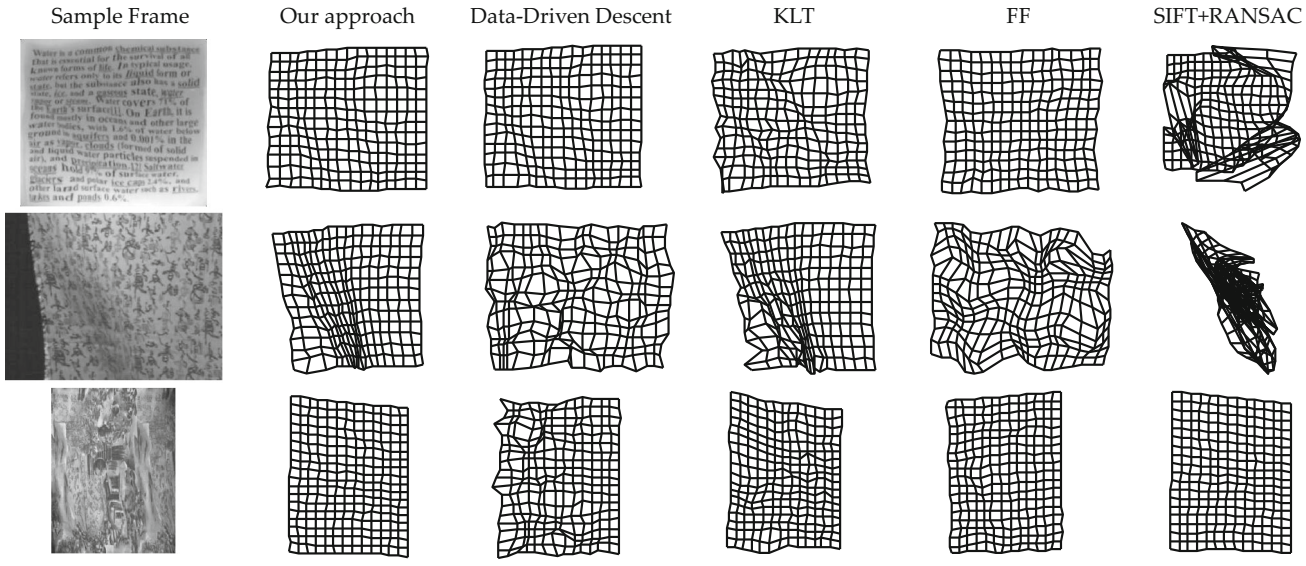

Fig. 15 Example dense correspondence results given by our approach, data-driven descent, KLT, free-form registration and SIFT matching with RANSAC. Each row is a video, two from cloth deformation and one from underwater imaging. The mesh is a regular grid on the template

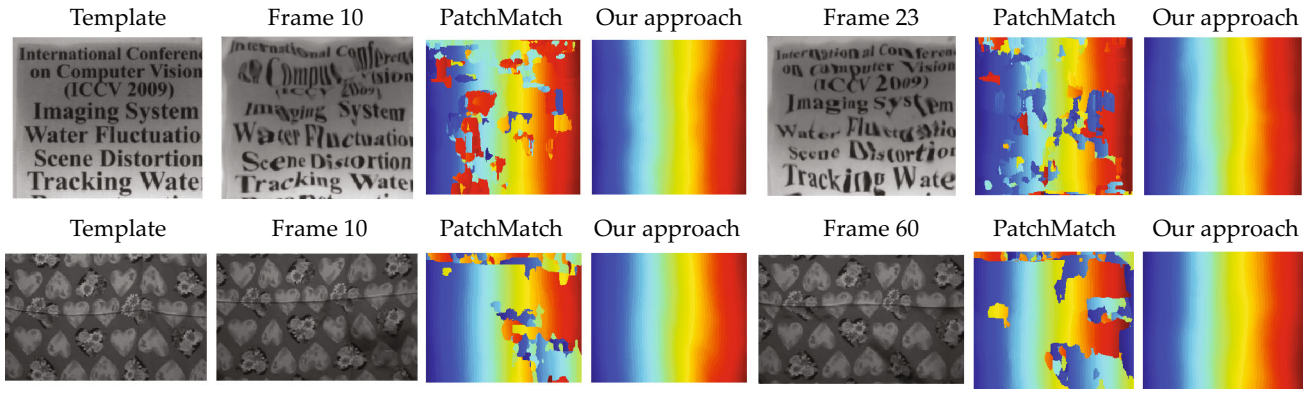

Fig. 16 Comparison with PatchMatch (Barnes et al. 2009) in water distortion and cloth deformation. *Column 1* (and 4) shows the deformed image and *Column 2* and 3 (or 5 and 6) show the x component of warping field $W(\mathbf{x})$ given by PatchMatch and our approach, respectively. PatchMatch gives roughly the correct estimation. However, details are not estimated correctly due to repetitive patterns, as illustrated by the discontinuous boundaries. On the other hand, our approach gives a much smoother solution

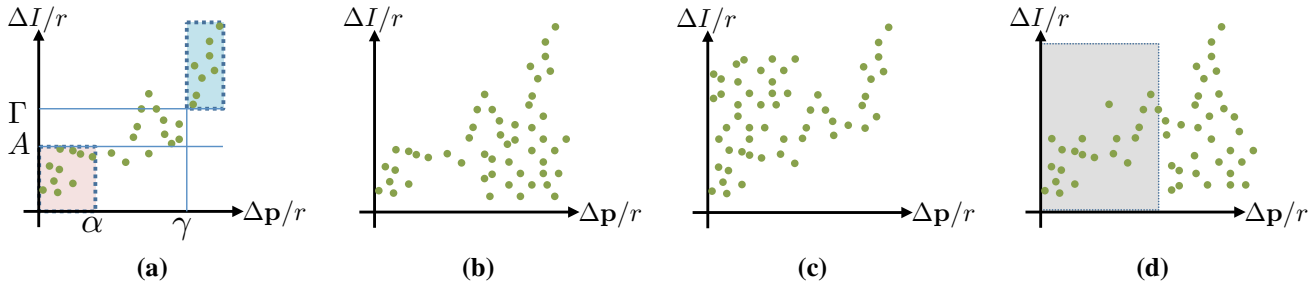

Fig. 17 Failure cases of relaxed Lipschitz conditions. **a** Relaxed Lipschitz conditions (Eqs. 10 and 11). **b, c** Failure cases, when $\Delta\mathbf{p}$ is large (or small), there is no correlation between ΔI and $\Delta\mathbf{p}$. **d** By restricting the parameter difference within some range, again the Relaxed Lipschitz Condition can be used

correct. These correct matches will gradually propagate to the entire image. To alleviate local maxima issues, correspondences computed in the coarse level are thus used as an initialization for the finer level, also in each iteration there is a random search procedure that allows correspondences to take a big step.

Figure 16 illustrates the warping field $W(\mathbf{x})$ estimated from PatchMatch and from our approach on deforming repetitive patterns. PatchMatch successfully estimates the global deformation, but fails to correct matches between a local pattern and a spatially incorrect but similar local pattern. Also random search does not help. On the other hand, our approach gives smooth and consistent matches.

11 Limitations and Future Work

The relaxed Lipschitz condition is very general. It captures the characteristics of a large range of relationships, no matter it is linear or not, it is a mapping or not. However, there are still mappings that cannot be model by the conditions. One such example is illustrated in Fig. 17b: when the parameter difference is smaller, the image difference is also small; however, the image difference no longer makes sense when the parameter difference is large. In this case, any legal tuple (α, γ) has $\gamma > 1$, and a NN predictor does not gives any improvement over a trivial prediction. Indeed, the motivation of hierarchical structure in this paper, is to make sure each NN predictor operates in the region that relaxed Lipschitz conditions make sense (Fig. 17d).

In this paper, all relaxed Lipschitz conditions are independently assumed at each layer (Fig. 18a). Since all the conditions are related to overlapping subsets of parameters and subregions of images, these conditions themselves must be related. For example, the 4-tuple $(\alpha_j, \gamma_j, A_j, \Gamma_j)$ at patch j may be a function of the 4-tuple at its children. Such a relationship leads to a reduction of the number of assumptions (Fig. 18b). In its extreme, it might be the case that all such conditions on the higher level can be derived from the condition in the lowest level.

Second, instead of using just image pyramid as in this paper, we may learn to build the image representations for each layer to make Lipschitz conditions better. The relaxed Lipschitz conditions resemble the optimization goal for distance learning and locality sensitive hashing. Therefore, it may be possible to find better image representations or bet-

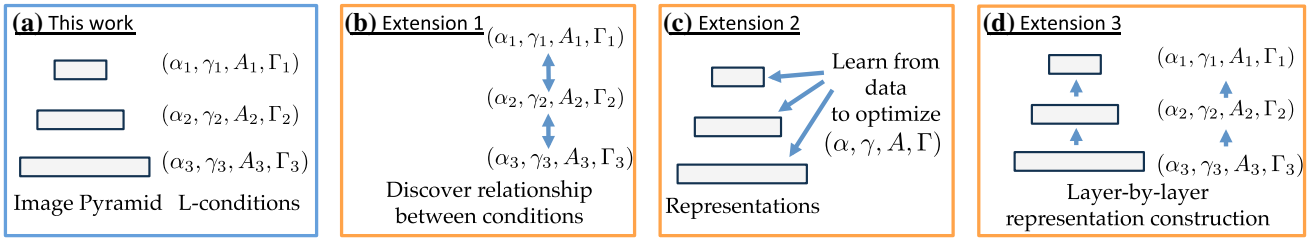

Fig. 18 Current work and possible extensions. **a** This work independently assumes Lipschitz conditions at each layer. **b** Since all conditions are related to sub-region of deformed image and subset of its control points, there is a relationship between conditions of different layers. **c** Instead of using image pyramid, we learn representations from the data to optimize the conditions. **d** Layer-by-layer construction of representations achieves both **(b, c)**. Relationships are established between consecutive layers, and representation can be tuned with data

ter distance metric on image appearance to maximize the gap between Lipschitz constants α and γ, and thus make deformation estimation easier with substantial fewer samples (Fig. 18c).

Even further, if we use the representation at layer t to compute the representation of the higher layer $t-1$, then layers are now explicitly connected and their Lipschitz conditions are directly related. This architecture (Fig. 18d) naturally combines the previous two cases (Fig. 18b, c) and resembles Deep Learning structures (e.g. Krizhevsky et al. 2012). It would be interesting to explore the theoretical properties suggested by Lipschitz conditions, and find theorems that show the efficiency and effectiveness of deep structures.

This hierarchy works if the degrees of freedom are evenly distributed in the image. If most degrees of freedom are concentrated within a small region of the image, then this hierarchy may not reduce the number of samples needed. This might not happen for smooth deformation on textured template, however for irregular deformation where some regions carry deformations of substantially high frequency than other regions, a set of non-uniformly distributed landmarks may be needed.

This paper mainly addresses the estimation of 2D deformation. The theory can be applied to 3D deformations with self occlusions, however the associated complicated generative model may require accurate rendering tools for training data synthesis.

For 3D deformation with self-occlusion and substantial viewpoint changes, substantial works need to be done. Also, we assume that the training distribution be specific so that theoretical assertions can be made. When we only have arbitrarily distributed training samples and no deformation model, as a future work, a learning algorithm should be designed to construct the hierarchical model automatically.

12 Conclusion

In this paper, we propose a novel hierarchical framework that systematically explains how the coarse-to-fine approach works in image deformation estimation. Based on our finding, we construct an algorithm that achieve worst-case guarantees for nonrigid deformation estimation, which from optimization point of view, is nonlinear, high-dimensional and non-convex. Compared to previous work (Tian and Narasimhan 2012), the sample complexity of our approach is $O(C_1^d + C_2 \log 1/\epsilon)$, substantially smaller than that required for brute-force nearest neighbor ($O(1/\epsilon^d)$) and also data-driven descent (Tian and Narasimhan 2012) ($O(C^d \log 1/\epsilon)$) that does not take hierarchy into consideration. To achieve this goal, we propose relaxed Lipschitz conditions (Eqs. 10 and 11) at each patch and each level of hierarchy, which is weaker and more general compared to the global Lipschitz conditions.

Practically, unlike previous approaches (Beauchemin and Barron 1995; Barnes et al. 2009, 2010), ours does need training samples to build a hierarchical model. However, the training samples can be generated from the template and its specific deformation model, enabling the algorithm to be used for finding correspondence of two images. We have done extensive experiments on real and synthetic data to verify our framework.

Acknowledgments This research was supported in parts by ONR grant N00014-11-1-0295, a Microsoft Research PhD fellowship, a University Transportation Center T-SET grant and a gift from TONBO Imaging.

Appendix 1: Correctness of Algorithm 1

Without loss of generality and for notation simplicity, we omit the subscript j and set $r_j = 1$. We first define the following quantities:

Definition 1 (*Allowable set of A and* Γ) Given α, allowable set $\tilde{A}(\alpha)$ is defined as:

$$\tilde{A}(\alpha) = \{A : \forall l \ \Delta \mathbf{p}_l \leq \alpha \implies \Delta I_l \leq A\} \tag{50}$$

Intuitively, $\tilde{A}(\alpha)$ captures all plausible As that satisfy Eq. 10 for a given α. Similarly, given γ, the allowable set $\tilde{\Gamma}(\gamma)$ is

defined as:

$$\tilde{\Gamma}(\gamma) = \{\Gamma : \forall l \ \Delta\mathbf{p}_l \geq \gamma \implies \Delta I_l \geq \Gamma\} \tag{51}$$

Intuitively, $\tilde{\Gamma}(\gamma)$ captures all plausible Γs that satisfy Eq. 11.

The two allowable sets have the following properties:

Lemma 1 *If $\alpha' > \alpha$, then $\tilde{A}(\alpha') \subset \tilde{A}(\alpha)$. Similarly, if $\gamma' < \gamma$, then $\tilde{\Gamma}(\gamma') \subset \tilde{\Gamma}(\gamma)$.*

Proof The proof is simply by definition of the two sets. □

Then we proceed to analyze the two arrays $\Delta I_m^+ = \max_{1\leq l\leq m} \Delta I_l$ and $\Delta I_m^- = \min_{m\leq l\leq M} \Delta I_l$ constructed in Algorithm 1.

Lemma 2 (Properties of ΔI^+ and ΔI^-) The two arrays ΔI^+ and ΔI^- constructed in Algorithm 1 are monotonously increasing functions with respect to m, and $\Delta I_m^- \leq \Delta I_m^+$ for every $1 \leq m \leq M$ (Fig. 6b). Moreover, we have:

$$\Delta I_m^+ = \min \tilde{A}(\Delta\mathbf{p}_m) \tag{52}$$
$$\Delta I_m^- = \max \tilde{\Gamma}(\Delta\mathbf{p}_m) \tag{53}$$

Proof Both ΔI_m^+ and ΔI_m^- are monotonously increasing since when m increases, ΔI_m^+ is the maximal value over a larger set and ΔI_m^- is the minimal value over a smaller set. Also $\Delta I_m^- \leq \Delta I_m \leq \Delta I_m^+$.

Prove $\Delta I_m^+ \in +\tilde{A}(\Delta\mathbf{p}_m)$: For any $\Delta\mathbf{p}_l \leq \Delta\mathbf{p}_m$, since the list $\{\Delta\mathbf{p}_m\}$ was ordered, we have $l \leq m$. By the definition of ΔI_m^+, we have $\Delta I_l \leq \Delta I_m^+$. Thus $\Delta I_m^+ \in \tilde{A}(\Delta\mathbf{p}_m)$.

Prove $A \in \tilde{A}(\Delta\mathbf{p}_m)$, $\Delta I_m^+ \leq A$: For any $1 \leq l \leq m$, since $\Delta\mathbf{p}_l \leq \Delta\mathbf{p}_m$, by the definition of A, we have $\Delta I_l \leq A$, and thus $\Delta I_m^+ = \max_{1\leq l\leq m} \Delta I_l \leq A$.

Therefore, $\Delta I_m^+ = \min \tilde{A}(\Delta\mathbf{p}_m)$. Similarly we can prove $\Delta I_m^- = \max \tilde{\Gamma}(\Delta\mathbf{p}_m)$. □

Theorem 6 *For each m and $\alpha = \Delta\mathbf{p}_m$, Algorithm 1 always gives the globally optimal solution to the following linear programming:*

$$\min \ \gamma \tag{54}$$
$$\text{s.t.} \ \ A \in \tilde{A}(\alpha)) \tag{55}$$
$$\Gamma \in \tilde{\Gamma}(\gamma)) \tag{56}$$
$$A + 2\eta < \Gamma \tag{57}$$

which has at least one feasible solution ($A \to +\infty, \gamma \to -\infty, \Gamma \to -\infty$) for any α.

Proof (a) First we prove every solution given by Algorithm 1 is a feasible solution to the optimization (Eq. 54). Indeed, for any $\alpha = \Delta\mathbf{p}_m$, according to Lemma 2, If we set the solution to be the output of Algorithm 1:

$$(\alpha, \gamma, A, \Gamma) = (\Delta\mathbf{p}_m, \Delta\mathbf{p}_{l^*}, \Delta I_m^+, \Delta I_{l^*}^-) \tag{58}$$

Then since $A = \Delta I_m^+ \in \tilde{A}(\alpha)$ and $\Gamma = \Delta I_{l^*}^- \in \tilde{\Gamma}(\gamma)$, such a tuple satisfies Eqs. 55 and 56. From the construction of Algorithm 1, $A + 2\eta < \Gamma$. Thus, Algorithm 1 gives a feasible solution to Eq. 54.

(b) Then we prove Algorithm 1 gives the optimal solution. Suppose there is a better solution $(\alpha, \gamma', A', \Gamma')$. Obviously $A' = A = \min \tilde{A}(\alpha)$. Note that any optimal solution of γ must align with some $\Delta\mathbf{p}_l$. If there exists $l' < l^*$ so that $\gamma' = \Delta\mathbf{p}_{l'} < \Delta\mathbf{p}_{l^*} = \gamma$ is part of a better solution, then we have:

$$A' + 2\eta < \Gamma' \leq \max \tilde{\Gamma}(\gamma') \leq \max \tilde{\Gamma}(\gamma) = \Delta I_{l^*}^- \tag{59}$$

Therefore, we have $\Delta I_{l'}^- = \max \tilde{\Gamma}(\gamma') \leq \max \tilde{\Gamma}(\gamma) = \Delta I_{l^*}^-$. Since $\Delta I_{l'}^- \leq \Delta I_{l^*}^-$, there are two cases:

- $A \leq \Delta I_m^+ + 2\eta < \Delta I_{l'}^- < \Delta I_{l^*}^-$. This is not possible since the algorithm searching from m will stop at the minimal l^* that satisfies $\Delta I_m^+ + 2\eta < \Delta I_{l^*}^-$.
- $A \leq \Delta I_m^+ + 2\eta < \Delta I_{l'}^- = \Delta I_{l^*}^-$. Then according to the algorithm and monotonicity of ΔI^-, $l' = l^*$.

There fore, $l' = l^*$ and $(\alpha, \gamma', A', \Gamma')$ is given by Algorithm 1. □

From Theorem 6, for every α, Algorithm 1 always outputs the smallest γ that satisfies the Relaxed Lipschitz Conditions (Eqs. 10 and 11). Therefore, it outputs the curve $\gamma = \gamma^*(\alpha)$.

Appendix 2: Local Pullback Operation

Similar to pull-back operation introduced in data-driven descent (Tian and Narasimhan 2012), we can also introduce *local pull-back operation*:

$$I(R_j(\mathbf{q})) \equiv I(W(R_{j0}; \mathbf{q})) \tag{60}$$

In particular, for deformed image $I_\mathbf{q}$ and the moving region $R_j = R_j(\mathbf{q})$, we have

$$I_\mathbf{q}(R_j(\mathbf{q})) = I_\mathbf{q}(W(R_{j0}; \mathbf{q})) = I_0(R_{j0}) \tag{61}$$

which gives back the template content. Similar to pull-back inequality, we also have the following local pull-back inequality:

Theorem 7 *For jth patch with template region R_{j0} and radius r_j, if $\|\mathbf{p} - \mathbf{q}\|_\infty \leq r_j$ and $\|\mathbf{q}\|_\infty \leq c_q$, then*

$$\|I_\mathbf{p}(R_j(\mathbf{q})) - I_{\mathbf{p}-\mathbf{q}}(R_{j0})\| \leq \eta_j r_j \tag{62}$$

where $\eta_j = c_B c_q c_G Area_j$. *Note* $c_G = \max_{\mathbf{x}} |\nabla I_{\mathbf{p}}(\mathbf{x})|_1$ *and* c_B *is a smoothness constant so that:*

$$\|(B(\mathbf{x}) - B(\mathbf{y}))\mathbf{p}\|_\infty \leq c_B \|\mathbf{x} - \mathbf{y}\|_\infty \|\mathbf{p}\|_\infty \tag{63}$$

To prove this, we start with the following lemma.

Lemma 3 (Unity bound) *For any* $\mathbf{x}$ *and any* $\mathbf{p}$*, we have* $\|B(\mathbf{x})\mathbf{p}\|_\infty \leq \|\mathbf{p}\|_\infty$.

Proof

$$\begin{aligned}\|B(\mathbf{x})\mathbf{p}\|_\infty &= \max_{\mathbf{x}} \left\{ \sum_i b_i(\mathbf{x})\mathbf{p}^x(i), \sum_i \mathbf{b}_i(\mathbf{x})\mathbf{p}^y(i) \right\} \\ &\leq \max \left\{ \max_i |\mathbf{p}^x(i)| \sum_i b_i(\mathbf{x}), \max_i |\mathbf{p}^y(i)| \sum_i b_i(\mathbf{x}) \right\} \\ &= \|\mathbf{p}\|_\infty \end{aligned} \tag{64}$$

using the fact that $\sum_i b_i(\mathbf{x}) = 1$ and $b_i(\mathbf{x}) \geq 0$ for any $\mathbf{x}$. □

We now show Theorem 7 is correct.

Proof For any $\mathbf{y} \in R_{j0}$, by definitions of Eqs. 60 and 1, we have:

$$I_{\mathbf{p}}(R_j(\mathbf{q}))(\mathbf{y}) = I_{\mathbf{p}}(W(\mathbf{y}; \mathbf{q})) \tag{65}$$

$$\begin{aligned} I_{\mathbf{p}-\mathbf{q}}(\mathbf{y}) &= T(W^{-1}(\mathbf{y}; \mathbf{p} - \mathbf{q})) \\ &= I_{\mathbf{p}}(W(W^{-1}(\mathbf{y}; \mathbf{p} - \mathbf{q}), \mathbf{p})) \end{aligned} \tag{66}$$

Now we need to check the pixel distance between $\mathbf{u} = W(\mathbf{y}; \mathbf{q})$ and $\mathbf{v} = W(W^{-1}(\mathbf{y}; \mathbf{p} - \mathbf{q}), \mathbf{p})$. Note that both are pixel locations on distorted image $I_{\mathbf{p}}$. If we can bound $\|\mathbf{u} - \mathbf{v}\|_\infty$, then from $I_{\mathbf{p}}$'s appearance, we can obtain the bound for $|I_{\mathbf{p}}(R_j(\mathbf{q}))(\mathbf{y}) - I_{\mathbf{p}-\mathbf{q}}(\mathbf{y})|$.

Denote $\mathbf{z} = W^{-1}(\mathbf{y}; \mathbf{p} - \mathbf{q})$ which is a pixel on the template. By definition we have:

$$\mathbf{y} = W(\mathbf{z}; \mathbf{p} - \mathbf{q}) = \mathbf{z} + B(\mathbf{z})(\mathbf{p} - \mathbf{q}) \tag{67}$$

then by Lemma 3 we have:

$$\|\mathbf{y} - \mathbf{z}\|_\infty = \|B(\mathbf{z})(\mathbf{p} - \mathbf{q})\|_\infty \leq \|\mathbf{p} - \mathbf{q}\|_\infty \leq r_j \tag{68}$$

On the other hand, the difference between $\mathbf{u}$ and $\mathbf{v}$ has the following simple form:

$$\mathbf{u} - \mathbf{v} = W(\mathbf{y}, \mathbf{q}) - W(\mathbf{z}, \mathbf{p}) = \mathbf{y} + B(\mathbf{y})\mathbf{q} - \mathbf{z} - B(\mathbf{z})\mathbf{p} \tag{69}$$

$$= B(\mathbf{z})(\mathbf{p} - \mathbf{q}) - B(\mathbf{z})\mathbf{p} + B(\mathbf{y})\mathbf{q} = (B(\mathbf{y}) - B(\mathbf{z}))\mathbf{q} \tag{70}$$

Thus, by the definition of c_B (Eq. 63), we have:

$$\|\mathbf{u} - \mathbf{v}\|_\infty \leq c_B \|\mathbf{y} - \mathbf{z}\|_\infty \|\mathbf{q}\|_\infty \leq (c_B \|\mathbf{q}\|_\infty) r_j \tag{71}$$

Thus:

$$\begin{aligned} &|I_{\mathbf{p}}(R_j(\mathbf{q}))(\mathbf{y}) - I_{\mathbf{p}-\mathbf{q}}(\mathbf{y})| \\ &= |I_{\mathbf{p}}(W(\mathbf{y}; \mathbf{q})) - I_{\mathbf{p}}(W(W^{-1}(\mathbf{y}; \mathbf{p} - \mathbf{q}), \mathbf{p}))| \\ &= |I_{\mathbf{p}}(\mathbf{u}) - I_{\mathbf{p}}(\mathbf{v})| \qquad (72) \\ &\leq |\nabla I_{\mathbf{p}}(\xi)|_1 \|\mathbf{u} - \mathbf{v}\|_\infty \leq c_B |\nabla I_{\mathbf{p}}(\xi)|_1 \|\mathbf{q}\|_\infty r_j \qquad (73) \end{aligned}$$

where ξ lies on the line segment connecting $\mathbf{u}$ and $\mathbf{v}$. Collecting Eq. 73 over the entire region $R_j(\mathbf{p})$ gives the bound. □

Practically, η_j is very small and can be neglected.

From Eq. 60, there is a relationship between the (global) pull-back operation $H(I, \mathbf{q}) \equiv I(W(\mathbf{x}; \mathbf{p}))$ defined in Tian and Narasimhan (2012) and the local pull-back operation $I(R_j(\mathbf{q}))$:

$$H(I, \mathbf{q})(R_{j0}) = I(W(R_{j0}; \mathbf{p})) = I(R_j(\mathbf{q})) \tag{74}$$

Therefore, to compute $I(R_j(\mathbf{q}))$ for all patches, just compute the global pull-back image $H(I, \mathbf{q})$ once and extract region R_{j0} for every j-th patch on the pulled-back image.

Appendix 3: Sampling in High-dimensional Subspace

Here we show how to count the number of ϵ-ball required (i.e., sample complexity) to cover a hypercube $[-r, r]^D$ in a D-dimensional parameter space. Then we discuss how to compute sample complexity if the parameters are on a d-dimensional subspace within the hypercube. Both cases are shown in Fig. 19.

Covering a D-dimensional Space

Lemma 4 (Sampling theorem, sufficient conditions) *With* $\lceil 1/\alpha \rceil^D$ *number of samples* ($\alpha < 1$)*, for any* $\mathbf{p}$ *in the hypercube* $[-r, r]^D$*, there exists at least one sample* $\hat{\mathbf{p}}$ *so that* $\|\hat{\mathbf{p}} - \mathbf{p}\|_\infty \leq \alpha r$.

Proof A uniform distribution of the training samples within the hypercube suffices. In particular, let

$$n = \left\lceil \frac{1}{\alpha} \right\rceil \tag{75}$$

Thus we have $1/n = 1/\lceil 1/\alpha \rceil \leq 1/(1/\alpha) = \alpha$. For every multi-index $(i_1, i_2, \ldots, i_D)$ with $1 \leq i_k \leq n$, we put one

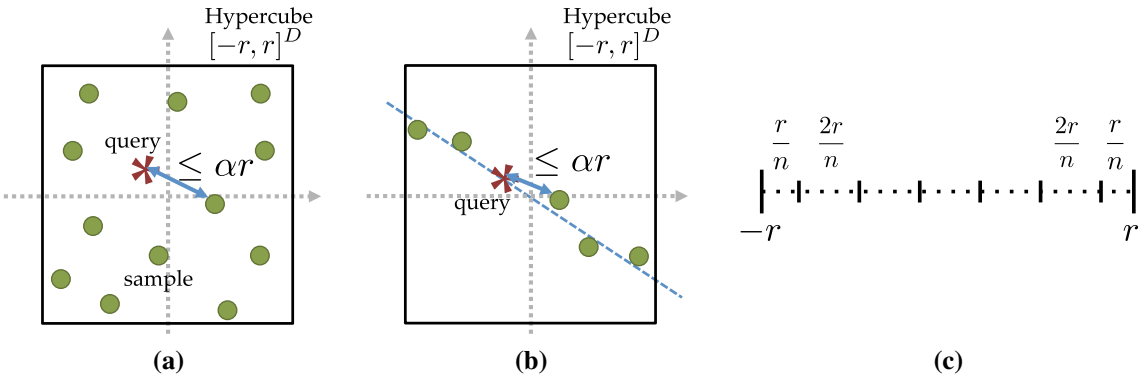

Fig. 19 Sampling strategies in D-dimensional space. **a** Uniform sampling within a hypercube $[-r, r]^D$ so that for any $\mathbf{p}(S) \in [-r, r]^{2|S|}$, there exists at least one training sample that is αr close to $\mathbf{p}(S)$. **b** If $d < D$, then just sampling the subspace within the hypercube suffices. **c** Uniform sampling per dimension (See Lemma 4)

training sample on D-dimensional coordinates:

$$\hat{\mathbf{p}}_{i_1,i_2,\ldots,i_d} = r\left[-1 + \frac{2i_1 - 1}{n}, -1 + \frac{2i_2 - 1}{n}, \ldots, -1 + \frac{2i_D - 1}{n}\right] \tag{76}$$

Therefore, along each dimension, the first sample is r/n distance away from $-r$, then the second sample is $2r/n$ distance to the first sample, until the last sample that is r/n distance away from the boundary r (Fig. 19c). Then for any $\mathbf{p} \in [-r, r]^D$, there exists i_k so that

$$\left|\mathbf{p}(k) - r\left(-1 + \frac{2i_k - 1}{n}\right)\right| \leq \frac{1}{n}r \leq \alpha r \tag{77}$$

This holds for $1 \leq k \leq D$. As a result, we have

$$\|\mathbf{p} - \hat{\mathbf{p}}_{i_1,i_2,\ldots,i_D}\|_\infty \leq \alpha r \tag{78}$$

and the total number of samples needed is $n^D = \lceil 1/\alpha \rceil^D$. □

Covering a Manifold in D-dimensional Space

Now we consider the case that $\mathbf{p}$ lies on a manifold $\mathcal{M}$ embedded in D-dimensional space. This means that there exists a function f (linear or nonlinear) so that for every $\mathbf{p}$ on the manifold and within the hypercube $[-r, r]^D$, there exists a d-dimensional vector $\mathbf{v} \in [-r, r]^d$ with $\mathbf{p} = f(\mathbf{v})$. For example, this happens if we use over-complete local bases to represent the deformation. Note that the function f is onto:

$$\left([-r, r]^D \cap \mathcal{M}\right) \subset f([-r, r]^d) \tag{79}$$

In this case, we do not need to fill the entire hypercube $[-r, r]^D$, but rather fill the d-dimensional hypercube $[-r, r]^d$, which requires the number of samples to be exponential with respect to only d rather than D. To prove this, we first define the expanding factor c regarding to the mapping:

Definition 2 ($Expandingfactorc$) The expanding factor c for a mapping f is defined as:

$$c \equiv \sup_{\mathbf{v}_1, \mathbf{v}_2 \in [-r,r]^d} \frac{\|f(\mathbf{v}_1) - f(\mathbf{v}_2)\|_\infty}{\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty} \tag{80}$$

We thus have the following sampling theorem for deformation parameters $\mathbf{p}$ on a manifold:

Theorem 8 (Sampling theorem, sufficient condition in the manifold case) *With $c_{SS}\lceil 1/\alpha \rceil^d$ samples distributed in the hypercube $[-r, r]^d$, for any $\mathbf{p} \in \mathcal{M}$, there exists at least one sample $\hat{\mathbf{p}} = f(\hat{\mathbf{v}})$ so that $\|\hat{\mathbf{p}} - \mathbf{p}\|_\infty \leq \alpha r$. Note $c_{SS} = \lceil c \rceil^d$.*

Proof We first apply Theorem 4 to the hypercube $[-r, r]^d$. Then with $\lceil \frac{c}{\alpha} \rceil^d$ samples, for any $\mathbf{v} \in [-r, r]^d$, there exists a training sample $\mathbf{v}^{(i)}$ so that

$$\|\mathbf{v} - \mathbf{v}^{(i)}\|_\infty \leq \frac{\alpha r}{c} \tag{81}$$

We then build the training samples $\{\mathbf{p}^{(i)}\}$ by setting $\mathbf{p}^{(i)} = f(\mathbf{v}^{(i)})$. For any $\mathbf{p} \in [-r, r]^D$, there exists an $\mathbf{v} \in [-r, r]^d$ so that $\mathbf{p} = f(\mathbf{v})$. By the sampling procedure, there exists $\mathbf{v}^{(i)}$ so that $\|\mathbf{v} - \mathbf{v}^{(i)}\|_\infty \leq \frac{\alpha}{c}r$, and therefore:

$$\|\mathbf{p} - \mathbf{p}^{(i)}\|_\infty \leq c\|\mathbf{v} - \mathbf{v}^{(i)}\|_\infty \leq \alpha r \tag{82}$$

setting $\hat{\mathbf{p}} = \mathbf{p}^{(i)}$ thus suffices. Finally, since $\lceil ab \rceil \leq \lceil a \rceil \lceil b \rceil$, we have:

$$\left\lceil \frac{c}{\alpha} \right\rceil^d \leq \lceil c \rceil^d \left\lceil \frac{1}{\alpha} \right\rceil^d \tag{83}$$

and the conclusion follows. □

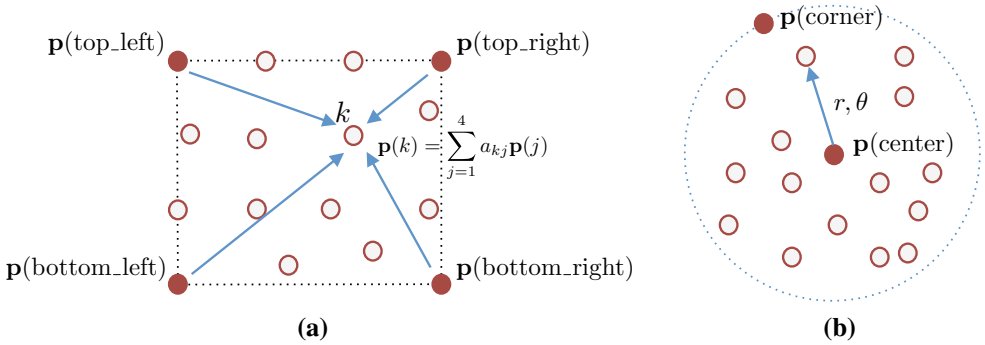

Fig. 20 Finding (meta)-control points to parameterize the deformation manifold. **a** Affine deformation within a *rectangle*. The intrinsic dimension is 6 while we pick four points ($d = 8$) to represent the landmark displacement with expanding factor $c = 1$. Note that picking just three points will also characterize affine, but with a much larger expanding factor c. **b** Deformation that includes translation and rotation. We pick two points ($d = 4$), leading to an expanding factor $c \leq 2 + \sqrt{2}$

We can see from the proof that c plays a quite important role for the number of samples needed. To make c smaller, d is not necessarily the intrinsic dimension of the manifold, but can be slightly more. Theorem 8 applies to any parameterization of the manifold $\mathcal{M}$.

Let us compute c for some global deformation fields. For example, an affine deformation field within a rectangle parameterized by $D/2$ landmarks is always 6-dimensional. To make c smaller, we (meta)-parameterize the field by 4 meta control points ($d = 8$) sitting at the four corner of the rectangle (Fig. 20a). In this case, any landmark displacement $\mathbf{p}(k)$ within this rectangle can be linearly represented by the locations of four corners in a convex manner:

$$\mathbf{p}(k) = A_k \mathbf{v} = \sum_{j=1}^{4} a_{kj} \mathbf{v}(j) \tag{84}$$

Here $\mathbf{v}$ is the concatenation of four deformation vectors [$\mathbf{p}$(top_left), $\mathbf{p}$(bottom_left), $\mathbf{p}$(top_right), $\mathbf{p}$(bottom_right)] from the four corners, $0 \leq a_{kj} \leq 1$ and $\sum_j a_{kj} = 1$. For any $\mathbf{p} \in [-r, r]^D$, $\mathbf{v}$ can be found by just picking up the deformation of its four corners, and thus $\|\mathbf{v}\|_\infty \leq r$. Furthermore, we have for $\mathbf{v}_1, \mathbf{v}_2 \in [-r, r]^k$:

$$\begin{aligned} &\|\mathbf{p}_1 - \mathbf{p}_2\|_\infty = \|f(\mathbf{v}_1) - f(\mathbf{v}_2)\|_\infty \\ &\leq \max_k \sum_{j=1}^{4} a_{kj} \|\mathbf{v}_1(j) - \mathbf{v}_2(j)\|_\infty \leq \|\mathbf{v}_1 - \mathbf{v}_2\|_\infty \end{aligned} \tag{85}$$

Therefore, $c = 1$. The reason why we pick four corners, is to ensure all weights in linear combinations are between 0 and 1.

Similarly, for 3-dimensional deformation with pure translation and rotation, as used in Sect. 8, we can just pick $\mathbf{v}$ as the concatenation of two landmark displacements: the rotation center $\mathbf{p}$(center) and the corner point $\mathbf{p}$(corner) whose rest location is the most distant to the center than other landmarks (Fig. 20b). Denote $r_{\mathbf{corner}}$ as the distance. For other landmark displacement $\mathbf{p}(k)$ whose index k can be parametrized by polar coordinate (r, θ), we have:

$$\mathbf{p}(r, \theta) = \mathbf{p}(\text{center}) + \frac{r}{r_{\mathbf{corner}}} R(\theta)(\mathbf{p}(\text{corner}) - \mathbf{p}(\text{center})) \tag{86}$$

$$= (Id - \frac{r}{r_{\mathbf{corner}}} R(\theta))\mathbf{p}(\text{center}) + \frac{r}{r_{\mathbf{corner}}} R(\theta)\mathbf{p}(\text{corner}) \tag{87}$$

where Id is the identity matrix, $R(\theta)$ is the 2D rotational matrix and $r_{\mathbf{corner}}$ is the distance between the rest location of the center to that of the corner. Therefore, for two different $\mathbf{v}_1$ and $\mathbf{v}_2$, since $r \leq r_{\mathbf{corner}}$, we have:

$$\begin{aligned} &\|\mathbf{p}_1(r, \theta) - \mathbf{p}_2(r, \theta)\|_\infty \\ &\leq \left\| \left(I - \frac{r}{r_{\mathbf{corner}}} R(\theta) \right) (\mathbf{p}_1(\text{center}) - \mathbf{p}_2(\text{center})) \right\|_\infty \\ &\quad + \left\| \frac{r}{r_{\mathbf{corner}}} R(\theta)(\mathbf{p}_1(\text{corner}) - \mathbf{p}_2(\text{corner})) \right\|_\infty \\ &\leq 2\|\mathbf{p}_1(\text{center}) - \mathbf{p}_2(\text{center}))\|_\infty \\ &\quad + \sqrt{2}\|\mathbf{p}_1(\text{corner}) - \mathbf{p}_2(\text{corner}))\|_\infty \qquad (88) \\ &\leq (2 + \sqrt{2})\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty \qquad (89) \end{aligned}$$

since $|\cos(\theta)| + |\sin(\theta)| \leq \sqrt{2}$. Therefore,

$$\begin{aligned} &\|\mathbf{p}_1 - \mathbf{p}_2\|_\infty = \max_{r,\theta} \|\mathbf{p}_1(r, \theta) - \mathbf{p}_2(r, \theta)\|_\infty \\ &\leq (2 + \sqrt{2})\|\mathbf{v}_1 - \mathbf{v}_2\|_\infty \end{aligned} \tag{90}$$

So $c = 2 + \sqrt{2} \leq 3.5$. This constant is used in Sect. 8 to compute the number of samples required by the theory.

References

Baker, S., & Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, *56*, 221–255.

Barnes, C., Shechtman, E., Finkelstein, A., & Goldman, D. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, *28*(3), 24.

Barnes, C., Shechtman, E., Goldman, D. B., & Finkelstein, A. (2010). The generalized patchmatch correspondence algorithm. In *ECCV*, 2010 (pp. 29–43). Berlin: Springer.

Beauchemin, S. S., & Barron, J. L. (1995). The computation of optical flow. *ACM Computing Surveys (CSUR)*, *27*(3), 433–466.

Bookstein, F. L. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *6*, 567–585.

Cao, X., Wei, Y., Wen, F., & Sun, J. (2012). Face alignment by explicit shape regression. In *CVPR*.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI* (pp. 674–679).

Matthews, I., & Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, *60*, 135–164.

Moll, M., & Gool, L. V. (2012). Optimal templates for non-rigid surface reconstruction. In *ECCV*.

Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., & Hawkes, D. (1999). Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Transactions on Medical Imaging*, *18*, 712–721.

Salzmann, M., Hartley, R., & Fua, P. (2007). Convex optimization for deformable surface 3-d tracking. In *ICCV*.

Salzmann, M., Moreno-Noguer, F., Lepetit, V., & Fua, P. (2008). Closed-form solution to non-rigid 3d surface registration. In *ECCV*.

Serre, T., Wolf, L., & Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *CVPR* (Vol. 2, pp. 994–1000).

Shi, J., & Tomasi, C. (1994). Good features to track. In *CVPR*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR*.

Tan, D. J., Holzer, S., Navab, N., & Ilic, S. (2014). Deformable template tracking in 1 ms. In *ECCV*.

Taylor, J., Jepson, A., & Kutulakos, K. (2010). Non-rigid structure from locally-rigid motion. In *CVPR*.

Tian, Y., & Narasimhan, S. G. (2012). Globally optimal estimation of nonrigid image distortion. *International Journal of Computer Vision*, *98*, 279–302.

Zhang, S., Zhan, Y., Zhou, Y., Uzunbas, M., & Metaxas, D. (2012). Shape prior modeling using sparse representation and online dictionary learning. *Medical image computing and computer-assisted intervention* (Vol. 7512, pp. 435–442)., Lecture notes in computer science Berlin: Springer.