The Problem: Computing distances between data vectors is an essential part of many machine learning algorithms. Most image representations do not allow distance between images to be measured in a way that is robust with respect to common spatial variations, such as object deformation, changes in object pose, and misalignment.

Given a pair of similar images, we deform one to the other, allowing pixels to be compared in common spatial coordinates. We present a simple hierarchical algorithm for finding a smooth warp from one image to another and demonstrate that the method improves the performance of a variety of learning algorithms.

Previous Work: Several prior algorithms attempt to provide a deformation-invariant distance measure. Tangent distance [2] uses a planar approximation to a local image manifold, allowing small spatial transformations to be modeled in pixel space. Because of the planar approximation, tangent distance does not work well for large deformations. The “shape context” method [1] is good at obtaining deformation invariance, but loses information by using only samples from image edges.

The most closely related prior work is in the automatic alignment of faces. One algorithm [3] computes deformation fields that are used to warp faces in order to compute image distance. We use a similar concept, but provide an alignment algorithm that is simpler and faster. More important, we demonstrate that this approach is of broad usefulness by considering a wider range of applications in learning-based vision.

Approach: To find a smooth vector field that maps one image to another, we make use of various techniques developed for optical flow. However, unlike optical flow computation, we are comparing arbitrary image pairs, not consecutive video frames. We seek smooth flow fields to avoid the optical flow difficulties related to edge discontinuities.

Our approach also differs from optical flow in that we desire an efficient way to compute all-pairs distances. We pre-process each of the N images so that the $N^2$ comparisons are as fast as possible. This pre-processing consists of computing a pyramid of motion confidence values, based on motion estimation matrices for image patches at multiple scales.

To perform a comparison between two images, we estimate motion (via the methods of [4]) only for patches that contain sufficient motion confidence. For patches without sufficient confidence, we interpolate values from a lower resolution. Once we have warped one image to the other, we can measure distance using any image distance metric (for example, Euclidean pixel distance or a measure that normalizes local brightness).

Impact: Our approach provides a substantial improvement over traditional pixel-based image comparison in several learning domains: image regression, non-linear dimensionality reduction, and object classification.

We use Gaussian-kernel-weighted nearest neighbors to predict the direction of lighting in a labeled dataset (from [5]). The dataset contains two degrees of spatial variation in addition to the changes in lighting direction. Our method is able to reduce the influence of these spatial variations. The 10-fold cross-validated error values (with a 95% confidence interval) are given in Table 1.

We demonstrate that our distance measure can result in better dimensionality reduction using IsoMap [5]. To evaluate the quality of the dimensionality reduction, we use the low-dimensional data to predict the lighting direction, via linear regression and regression trees (using the M5’ algorithm [6]).

We select a classification dataset that is difficult because the intra-class pose variation has a much larger effect in pixel space than the inter-class texture variation (the images—similar-looking athletic shoes—lie on a tangle of pixel manifolds). The backgrounds are removed, allowing automatic centering and rescaling. This gives standard pixel
Table 2: RMS error for light source direction (in degrees)

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>PIXEL-BASED</th>
<th>WARP-BASED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel-Weighted Neighbors ($\sigma = 5$)</td>
<td>10.0270 ± 0.5379</td>
<td>4.6341 ± 0.3192</td>
</tr>
<tr>
<td>Kernel-Weighted Neighbors ($\sigma = 10$)</td>
<td>8.7734 ± 0.5702</td>
<td>5.5043 ± 0.6859</td>
</tr>
</tbody>
</table>

Table 3: RMS error for light source direction (in degrees)

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>PIXEL-BASED</th>
<th>WARP-BASED</th>
</tr>
</thead>
<tbody>
<tr>
<td>LinReg in 3 dims</td>
<td>14.9736 ± 0.8622</td>
<td>10.6249 ± 0.7130</td>
</tr>
<tr>
<td>LinReg in 10 dims</td>
<td>13.9441 ± 1.0549</td>
<td>9.5208 ± 0.6298</td>
</tr>
<tr>
<td>M5’ in 3 dims</td>
<td>12.0287 ± 0.4321</td>
<td>7.3468 ± 0.4602</td>
</tr>
<tr>
<td>M5’ in 10 dims</td>
<td>11.1636 ± 1.0387</td>
<td>6.9142 ± 0.6224</td>
</tr>
</tbody>
</table>

distance some hope of succeeding, but leaves fine-scale alignment and local deformation for the warping approach.

Table 4: Classification error rates

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>PIXEL-BASED</th>
<th>WARP-BASED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Nearest Neighbor</td>
<td>0.3033</td>
<td>0.1367</td>
</tr>
<tr>
<td>Kernel-Weighted Neighbors</td>
<td>0.3100</td>
<td>0.1383</td>
</tr>
<tr>
<td>SVM ($\sigma = 5$)</td>
<td>0.2300</td>
<td>0.0683</td>
</tr>
<tr>
<td>SVM ($\sigma = 10$)</td>
<td>0.1883</td>
<td>0.0400</td>
</tr>
<tr>
<td>SVM ($\sigma = 50$)</td>
<td>0.2233</td>
<td>0.0600</td>
</tr>
</tbody>
</table>

**Future Work:** We present one efficient way of computing a warp-based distance between images, but there are certainly many other ways of doing this and many other kinds of invariance one could attempt to capture in a distance measure. For example, we would like to extend this algorithm to provide an invariance to background clutter by detecting regions for which the warp error is large and treating these separately from foreground regions. One long term goal would be to create a commonly-used image distance measure that can be automatically configured to provide different kinds of invariance.

**References:**


