# A Fast Illumination and Deformation Insensitive Image Comparison Algorithm Using Wavelet-Based Geodesics

Anne Jorstad[1], David Jacobs[1], Alain Trouvé[2]

[1] University of Maryland, College Park, MD, USA
[2] Ecole Normale Supérieure de Cachan, France

**Abstract.** We present a fast image comparison algorithm for handling variations in illumination and moderate amounts of deformation using an efficient geodesic framework. As the geodesic is the shortest path between two images on a manifold, it is a natural choice to use the length of the geodesic to determine the image similarity. Distances on the manifold are defined by a metric that is insensitive to changes in scene lighting. This metric is described in the wavelet domain where it is able to handle moderate amounts of deformation, and can be calculated extremely fast (less than 3ms per image comparison). We demonstrate the similarity between our method and the illumination insensitivity achieved by the Gradient Direction. Strong results are presented on the AR Face Database.

## 1 Introduction

The presence of lighting changes and deformations complicates the task of general image comparison. We present a fast algorithm that can handle illumination variation and moderate amounts of deformation in an efficient wavelet-based geodesic framework. Expressing the image matching cost in the wavelet domain allows us to derive an algorithm where the complete cost calculation requires only $O(n)$ table lookups, for $n$ the number of pixels in one image.

Considering images as points on a high dimensional image manifold, defining a metric to give local structure to the manifold allows paths to be calculated between images along the manifold; see Fig. 1. Computer Vision literature frequently uses geodesics in a Manifold Learning framework, where many given images are assumed to lie on a manifold and paths are defined by edges through sets of known images. In this work, we are given only two images, and we consider the geometry of the manifold, as induced by the chosen metric, to calculate the length of the path between them. It is natural to use the length of the geodesic, or locally shortest path, to define the similarity between two images, and geodesics provide significant information about the ways in which images differ. Points along a geodesic curve are images that have morphed part way from the first image to the second, and changes such as lighting and deformations can be introduced gradually through time. Being able to construct and manipulate geodesics has many applications, including accurate image interpolation [17], the ability to extract nonlinear statistics from a set of images on a manifold [18], and image registration [1]. In this work we aim to measure geodesic lengths on an image manifold, and provide a framework that can then be extended for further applications.

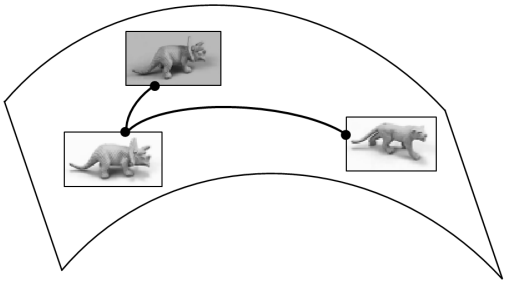


**Fig. 1.** A high-dimensional manifold, where each point on the manifold is an $M \times N$-dimensional image, and geodesics connecting more similar images are shorter (images from [8]).

Due to their high dimensionality, calculating geodesic distances can be a very expensive task directly, but we show that by working in the wavelet domain with a well-chosen metric, we can solve this problem very efficiently. To define an appropriate manifold of images, we will use a metric that is insensitive to changes in lighting and moderate amounts of deformation. The metric depends on image gradients, as gradients are less sensitive to changes in lighting than are direct pixel intensities. We will achieve results similar to those from the illumination-insensitive Gradient Direction, but here we also have a meaningful geodesic in addition to a simple difference value. We will show that our lighting cost is insensitive to moderate amounts of deformation when accumulated over several scales.

A geodesic-based image comparison framework has been considered in the past in works including [1]. Wavelets have been used to obtain insensitivity to group actions in works such as [2], and an efficient approximation of the somewhat deformation-insensitive Earth Mover's Distance has been calculated in the wavelet domain in [15]. The insensitivity of the gradient to lighting change has been shown numerous times such as in [10], where normalized gradients are used as features so that the SIFT descriptors are invariant to affine changes in illumination. We modify the gradient-based lighting-insensitive metric presented in [7]. Handling illumination changes and deformations together in an Optical Flow framework has been studied in works such as [13], but our proposed algorithm can be computed several orders of magnitude faster than these previous methods, while still providing accurate matching costs.

The primary contributions of this work are threefold: 1) a method using geodesics to calculate an illumination-insensitive image comparison cost similar to the Gradient Direction, but useful for applications where manipulating geodesics is required; 2) the insight that local dependencies can be removed by using an appropriate wavelet domain to express an image matching cost function based on gradient terms, allowing the cost computation to be separated into independent problems at every point location in wavelet space; and 3) a very fast calculation of this image comparison cost.

## 2 Geodesics for Object Identification

Identifying objects in pairs of images is made challenging by changes in pose, lighting, deformations, and occlusions. If these changes could be introduced gradually over the

course of several images, they would be much easier to handle. If we consider the manifold of images of a single class of object, where every point on the manifold is some instance of that object, then paths through the manifold connecting two images would consist of a continuum of images morphing from the first image to the second, like a video playing over time. The similarity of two instances of an object could then be defined by the length of the geodesic connecting them on the manifold, where shorter paths imply more similar objects; see Fig. 1.

Given a manifold of $M \times N$-dimensional images, we define a metric on this manifold so that it has a quantifiable structure, making it a Riemannian manifold [4]. The metric defines how costly it is to take an infinitesimal step in any given direction from any given point, and can be thought of as an $M \times N$-dimensional topographical map, where walking up a hill in one direction costs more than walking downwards in a different direction. On the Euclidean plane, the metric is $d(p_1, p_2) = \|p_2 - p_1\|_2$, but a metric can be defined in many ways as long as it is a locally linear metric. The metric chosen to define the manifold can be constructed to heavily penalize certain types of image variations, while allowing other variations to have low costs. For example, we would like an image metric that allows scene lighting to change at little cost, while object instance changes should come with a very high cost.

The length $L$ of a path $I(t)$ from $t = 0$ to $t = 1$ on a manifold is defined, for any given metric $\|\cdot\|$, to be

$$L(I(0), I(1)) = \int_0^1 \left\| \frac{dI}{dt} \right\| dt. \tag{1}$$

In order to calculate the geodesic path connecting $I(0)$ and $I(1)$, we must find the minimum cost path $I(t)$ along the manifold. This becomes an optimization problem, where we want to solve $I_{\text{geod}}(t) = \arg\min_{I(t)} \; L(I(0), I(1))$. Geometrically, a geodesic is a curve whose tangent vectors $\frac{dI}{dt}$ have constant length [4]. It can be shown that the length of the geodesic $I_{\text{geod}}(t)$ is also equal to

$$L_{\text{geod}}(I(0), I(1)) = \min_{I(t)} \sqrt{2E(I(t))}, \tag{2}$$

a function of the energy $E$ of the curve [19], where energy is defined as

$$E(I(t)) = \frac{1}{2} \int_0^1 \left\| \frac{dI}{dt} \right\|^2 dt, \tag{3}$$

which is familiar from classical mechanics where kinetic energy is $\frac{1}{2}mv^2$. The relation (2) can be understood intuitively because the tangent vectors all have constant length $c$, and so if $\int_0^1 \|c\| dt$ is minimal, then $\frac{1}{2}\int_0^1 \|c\|^2 dt$ must also be minimal, as squaring is a monotonic function. Therefore,

$$I_{\text{geod}}(t) = \arg\min_{I(t)} \int_0^1 \left\| \frac{dI}{dt} \right\| dt = \arg\min_{I(t)} \frac{1}{2} \int_0^1 \left\| \frac{dI}{dt} \right\|^2 dt. \tag{4}$$

We will choose an appropriate energy function and use the relation from (2) to help us calculate geodesic distances on the image manifold.

The metric defining the manifold on which the geodesics live can be adjusted for various applications, making this an elegant framework to handle an often messy problem, allowing images to update gradually and continuously through time. In the next sections we will discuss the metric and optimization schemes chosen to efficiently solve this problem.

## 3 A Lighting-Insensitive Metric

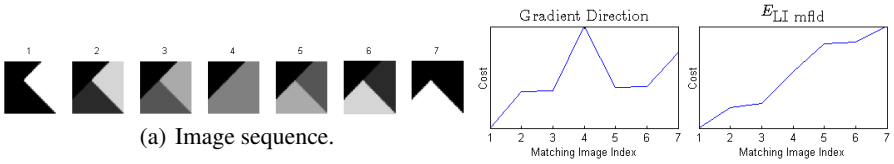
A pixel-based metric proposed in [7] was designed to be insensitive to changes in scene illumination, which the authors combined with a regularization term to handle deformations in an Optical Flow-like framework, calling the combined method the Deformation and Lighting Insensitive (DLI) metric. The lighting-insensitive (LI) term relating two images $I_1$ and $I_2$ was presented as

$$E_{\text{LI}}(I_1, I_2) = \frac{1}{2} \sum_{x,y} \frac{\|\nabla \delta I(x,y)\|^2}{\|\nabla I(x,y)\|^2 + \epsilon^2}, \tag{5}$$

where $\nabla \delta I$ and $\nabla I$ are defined in terms of $I_1$ and a second image $\hat{I_2}$ that is $I_2$ warped to match $I_1$ as closely as possible under certain constraints, so $\delta I = \hat{I_2} - I_1$ and $\nabla I = \nabla I_1$. The small constant $\epsilon$ is of the order of the noise in the image, and ensures that the denominator is never zero.

Using image gradients instead of intensities directly is known to be less sensitive to changes in lighting, for example from [10]. The Gradient Direction is a cost function commonly used when insensitivity to illumination change is desired. The direction of the image gradient $\theta = \tan^{-1}\left(\frac{I_y}{I_x}\right)$ is calculated at each pixel, then used in a sum-of-squared-differences image comparison, defining a cost between a given pair of images. This measure is invariant to adding a constant value to the image, or multiplying the image by a scalar, desirable properties for being insensitive to changes in scene illumination. However, it can be argued that a small change in illumination should be penalized less harshly than a large change in illumination.

The metric $E_{\text{LI}}$ has similar properties to the Gradient Direction, but is able to respond to different gradient relations appropriately, scaling the gradient of the image change $\delta I$ by the norm of the image gradient. Changing from a small to a medium gradient norm will be penalized more severely than changing from a medium to a large gradient norm. Comparing two smooth image regions should have a low cost, while comparing a smooth region to a jagged region should have a high cost. The image gradient is small at pixels that correspond to smooth regions of an object, and although a change in scene lighting will result in different pixel intensities, the relative intensities of the pixels will remain similar, and the gradient will remain small, so both the numerator and the denominator will be small, resulting in a low matching cost, and the desired property holds. In an image region where there is a geometric boundary, such as at the edge of a building, a change in scene lighting could affect the distinct surfaces in very different ways, but as the gradient is likely to be large across this boundary, matching a larger $\nabla \delta I$ is permissible at a lower cost as it will be weighted by the image gradient in the denominator. In an image region where there is an albedo change but little geometric change, for example a colored stripe on a white wall, the gradient across this




(a) Image sequence.

(b) Image matching costs.

**Fig. 2.** (a) Image sequence, where each image is compared to image 1, the leftmost image. (b) Gradient Direction and $E_{\text{LI mfld}}$ costs for each image pair in the image sequence.

boundary may be large, but as the scene lighting changes, $\nabla \delta I$ will scale with $\nabla I$, so as long as the pixels being compared correspond to the same points in the scene, the matching cost will remain low.

To understand the difference in behavior between the Gradient Direction and our new cost $E_{\text{LI mfld}}$, we provide a simple toy example Fig. 2, which could represent a series of images captured as a lighting source moves from one side of a building to another across a corner. Costs are calculated from the leftmost image in Fig. 2(a) to all images in the sequence, and these costs are presented in Fig. 2(b). As the change in intensity gets larger, the cost of $E_{\text{LI mfld}}$ steadily increases, and when the order of the intensity magnitudes reverses (from image 3 to image 5), this causes a jump in the costs. With Gradient Direction (mod $\pi$), the cases where two image regions have the same intensity (images 4 and 7) cause the comparison cost value to blow up, while otherwise the direction of the gradients and hence the costs are not discriminative.

We will use this metric to define a manifold that is insensitive to changes in illumination. Along any curve $I(t)$ (a continuum of images) on the manifold, for small step $\delta t > 0$, $\delta I(t) = I(t + \delta t) - I(t)$. The relation between two images from (5) defines a Riemannian structure on images using the infinitesimal norm

$$\|\delta I\|^2_{\text{LI}} = \frac{1}{2} \sum_{x,y} \frac{\|\nabla \delta I(x,y)\|^2}{\|\nabla I(x,y)\|^2 + \epsilon^2}. \tag{6}$$

Using this term in the energy function from (3), the energy of a curve $I(t)$ on this manifold is

$$E_{\text{LI mfld}}(I(t)) = \lim_{\delta t \to 0^+} \frac{1}{2} \int_0^{1-\delta t} \frac{\|\delta I\|^2_{\text{LI}}}{(\delta t)^2} dt. \tag{7}$$

We search for geodesics on this manifold in order to determine the distance $L_{\text{geod}}(I(0), I(1))$ between any given pair of input images $I(0)$ and $I(1)$. To calculate the geodesic from (4) we must therefore solve

$$I_{\text{geod}}(t) = \arg\min_{I(t)} \lim_{\delta t \to 0^+} \frac{1}{2} \int_0^{1-\delta t} \sum_{x,y} \frac{\|\nabla \delta I(x,y,t)\|^2}{\|\nabla I(x,y,t)\|^2 + \epsilon^2} \frac{1}{(\delta t)^2} dt. \tag{8}$$

### 3.1 Behavior of the Metric

In this section we will discuss the behavior of the geodesics defined by this lighting-insensitive metric at a single point location $(x, y)$. When the image gradient is near

zero, the metric is dominated by the $\frac{1}{\epsilon^2}$ term, and the cost scales nearly linearly with the change in the gradient.

In regions where the image gradients are large, the behavior is more exponential. This can be seen analytically without loss of generality if we consider the case where the gradient is zero in the $y$ dimension in both images, so that there is no change in gradient direction and $\nabla I = I_x$. For clarity let $\epsilon^2 = 0$, and take $I' = \lim_{\delta t \to 0^+} \frac{\delta I}{\delta t}$. This reduces (8) to

$$\arg\min_{I(t)} \frac{1}{2} \int_0^1 \left( \frac{I'_x}{I_x} \right)^2 dt, \tag{9}$$

which can be solved using the Euler-Lagrange equation [3], a technique that converts a functional to be minimized into a differential equation describing the minimizing function. Specifically, given a functional $J$ of the form $J(f) = \min_{f(t)} \int_0^1 F(t, f(t), f'(t))dt$, the function $f(t)$ that minimizes $J(f)$ is described by the equation $\frac{\partial F}{\partial f} - \frac{d}{dt}\frac{\partial F}{\partial f'} = 0$. Applying the Euler-Lagrange equation to (9), the resulting differential equation can be simplified to

$$(I'_x)^2 - I_x I''_x = 0. \tag{10}$$

It can be shown that $I_x(t) = ce^{rt}$ satisfies this equation for $c, r \in \mathbb{R}$, and any set of boundary conditions $I(0)$ and $I(1)$ will determine the specific values of these variables. We therefore see that when the value of $\epsilon$ is small with respect to the magnitudes of $\nabla \delta I$ and $\nabla I$, the gradient of $I$ behaves like an exponential, meaning that $I$ changes exponentially with time. So the cost function we seek to minimize is near linear when the image gradients are near zero, and near exponential when the image gradients are larger, which penalizes scene lighting variation as desired.

### 3.2 Disadvantages of Direct Optimization

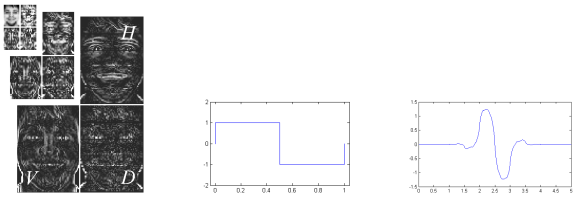
The most straightforward way to minimize (8) is to use a gradient descent optimization scheme. However, for input images of size $M \times N$, the geodesic path $I(t)$ has dimension $M \times N \times T$, for $T$ the number of time steps used to discretize the geodesic. Calculations with $\nabla E_{\text{LI mfld}}$ are cumbersome and easily get trapped in local minima. We avoid these computations by moving the problem into the wavelet domain, where we will show that it can be expressed as $M \times N$ distinct 1D problems that are straightforward to solve.

## 4 Optimization in the Wavelet Domain

We show that moving the norm $E_{\text{LI mfld}}$ into the wavelet domain results in a function that can be minimized over each independent variable separately, thereby vastly simplifying the minimization calculations and resulting in a very fast computation. We will also find that this representation provides insensitivity to moderate amounts of deformation.

### 4.1 Background on Wavelets

For our purposes, wavelets are a set of orthonormal functions that allow local analysis of a function according to scale; for details see [11]. At every scale the wavelet transform has three outputs, defined in the horizontal $H$, vertical $V$ and diagonal $D$ directions,




(a) (b) (c)

**Fig. 3.** (a) 2D Haar wavelet decomposition to three scales, (b) 1D Haar wavelet, (c) 1D biorthogonal spline wavelet.

and a downsampled version of the input that is then processed at the next scale; see Fig. 3(a). Wavelet functions used to filter an image can be constructed in a wide variety of forms, but for our purposes we consider only functions that have the same general form as a derivative filter. The 1D Haar wavelet at one scale (see Fig. 3(b)) is exactly a simple finite difference filter, and so filtering with a Haar wavelet is equivalent to downsampling by two and filtering with a finite difference filter in each dimension, i.e. extracting the gradient at every other pixel. The critical observation here is that each term of this wavelet transform is independent. Wavelet basis functions can be chosen to be orthogonal, and in this case changing the value of the wavelet coefficient at one location at one scale affects no other coefficients at any scale, for its support of two adjacent points at the next coarsest scale is downsampled by two, so these points influence no other coefficient. This allows us to define gradients in terms of independent wavelet coefficients. If we filter with a smoother wavelet of a similar gradient-like shape, such as the biorthogonal spline wavelet (see Fig. 3(c)), this can be considered to be filtering with a smoothed gradient filter, with desirable continuity properties. In this work we will use the family of biorthogonal spline wavelets (with orders nr = 1, nd = 3).

### 4.2 The Lighting Metric in the Wavelet Domain

We rewrite the function $E_{\text{LI mfld}}$ (7) in terms of wavelet coefficients. If these coefficients are defined so that $H(m,n)$ is the horizontal component and $V(m,n)$ is the vertical component of a 2D gradient-like wavelet calculated via a discrete wavelet transform, then $H \approx I_x$ and $V \approx I_y$, where each has been downsampled by a factor of two. Using the $L^2$ norm, $E_{\text{LI}}$ can be rewritten approximately as

$$E_{\text{wav}}(I) = \frac{1}{2} \sum_{m,n} \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2}, \tag{11}$$

where $H$ and $V$ depend on point locations $(m,n)$, but we leave this out of the notation for clarity, as $(m,n)$ are fixed inside the sum. The converted cost function does not make use of the diagonal component of the 2D wavelet decomposition, as all terms are expressible using only $H$ and $V$.

In the wavelet domain, each wavelet basis location is now independent of its neighbors, as the local descriptions of the gradients are handled during the wavelet filtering, a result of the orthogonality of the wavelets as described in the previous section. A primary contribution of this work is the insight that using the wavelet domain to express

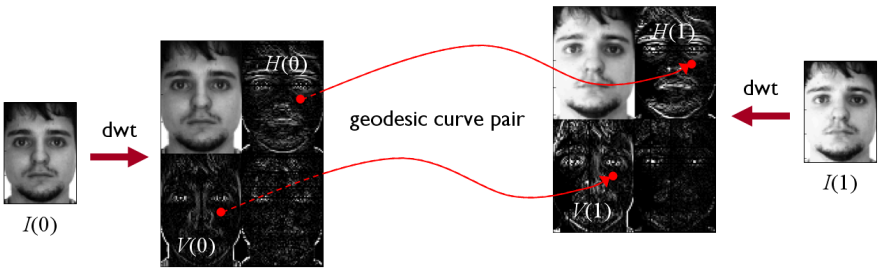



**Fig. 4.** Algorithm schematic: The discrete wavelet transform (dwt) is applied to the input images to generate the horizontal and vertical components $H$ and $V$ of the wavelet decomposition at one scale. At each point pair location in $H(0), V(0)$, the geodesic curve is calculated to the corresponding point location in $H(1), V(1)$. These curves are then integrated, and the resulting values from each point pair are summed for the total image matching cost.

an image matching cost function based on gradients allows the similarity computation to be separated into independent problems at every point location in wavelet space. We recall that the terms comprising the cost function in the wavelet domain are sampled from the original terms at every other pixel. Again taking $H' = \lim_{\delta t \to 0^+} \frac{\delta H}{\delta t}$ and $V' = \lim_{\delta t \to 0^+} \frac{\delta V}{\delta t}$ so that the $\frac{1}{(\delta t)^2}$ term cancels, the minimization problem (8) can be rewritten as

$$I_{\text{geod}}(t) = \frac{1}{2} \sum_{m,n} \underset{H(t),V(t)}{\arg\min} \int_0^1 \frac{H'^2 + V'^2}{H^2 + V^2 + \epsilon^2} dt. \tag{12}$$

where $H$ and $V$ are curves through time, and each individual point on the curves is in $\mathbb{R}^{M \times N}$. The $M \times N \times T$ dimensional problem of (8) has now been separated into $M \times N$ independent continuous 1D problems to be summed, one for each location $(m, n)$ in the wavelet domain. The geodesic path at each point location is defined by two 1D curves, $H(t)$ and $V(t)$, which are coupled, meaning that their geodesic paths are co-dependent and are optimized together; see Fig. 4. We can calculate the geodesic path for each point location separately, and then the full geodesic path of the image as a whole is simply the combination of all these distinct paths. The starting and ending values $H(0),\ H(1),\ V(0),\ V(1)$ are the coefficients from the wavelet decompositions of the given images $I(0)$ and $I(1)$, and so this reduces to a series of boundary value problems.

The minimization problem in (12) is a functional of a form that can be easily converted to a set of differential equations using the Euler-Lagrange equation [3], as described in Sec. 3, which can then be solved numerically. We chose to first convert the relation into polar coordinates, as this proved to be more stable to solve numerically. Defining $r = \sqrt{H^2 + V^2}$ and $\theta = \tan^{-1} \frac{V}{H}$, the inner functional to be minimized becomes

$$\underset{r(t),\theta(t)}{\arg\min} \int_0^1 \frac{r'^2 + r^2\theta'^2}{r^2 + \epsilon^2} dt. \tag{13}$$

Following the vector form of the Euler-Lagrange equation, the differential equations that describe the curves $r(t)$ and $\theta(t)$ that together minimize the term inside the sum

for a single point location $(m, n)$ are

$$\begin{aligned} r'' &= r\theta'^2 + (rr'^2 - r^3\theta'^2)(r^2 + \epsilon^2)^{-1}, \\ \theta'' &= 2r^{-1}r'\theta'(r^2(r^2 + \epsilon^2) - 1). \end{aligned} \tag{14}$$

This pair of second order equations can be solved as a system of four first order equations using any numerical integration scheme, and we chose to use the Boundary Value Problem solver from MATLAB. The output is a pair of numerical 1D curves $r(t)$ and $\theta(t)$, starting at $r(0), \theta(0)$ and ending at $r(1), \theta(1)$, that can be converted back to 1D curves $H(t), V(t)$, and that minimizes the cost from (12). This process is repeated for each wavelet domain point $(m, n)$ separately. We now have $M \times N$ pairs of geodesic curves. A visual schematic of the algorithm can be seen in Fig. 4.

Once all the optimal curves have been found, it remains to integrate along each of them to calculate the cost contribution from each location, and sum these point costs for the overall value of the energy of the image matching. These integrations can be computed numerically, discretizing the curve into $T$ segments and summing the value of the cost function at each of these segments. Once the total energy is calculated, we recall the relation from (2) and return the square root of twice the energy value as the true geodesic length.

### 4.3 Limiting Behavior

When $\epsilon$ is reduced to 0, equation (13) decouples into two separate problems:

$$\arg\min_{r(t)} \int_0^1 \frac{r'^2}{r^2} dt \quad \text{and} \quad \arg\min_{\theta(t)} \int_0^1 \theta'^2 dt. \tag{15}$$

These equations are optimized by exponential curves in $r(t)$ and linear curves in $\theta(t)$, and when the boundary values are included, the optimal curves are

$$r(t) = r_0 e^{\ln \frac{r_1}{r_0} t} = r_0 \left(\frac{r_1}{r_0}\right)^t \quad \text{and} \quad \theta(t) = (\theta_1 - \theta_0)t + \theta_0. \tag{16}$$

These functions can be integrated analytically, resulting in a total energy of

$$E = \left(\ln \frac{r_1}{r_0}\right)^2 + (\theta_1 - \theta_0)^2, \tag{17}$$

a value determined entirely by the boundary points, invariant to the path connecting them. This is observed to be exactly the cost of the Gradient Direction plus a constant term depending on the ratio of the lengths of the $H$ and $V$ terms in the two images. So we expect the cost reported here to be very similar to the Gradient Direction, but more highly penalizing cases where the difference in gradient norms between the two images is large, while the Gradient Direction is invariant to uniform scalar changes in intensity magnitude. It is reasonable and often desirable to have cases where a uniform intensity change is small be penalized less than cases where the magnitude is large. When the magnitude of $r$ is the same at corresponding pixels in both images, the cost is exactly

that of the Gradient Direction. In this limiting case when $\epsilon = 0$ the geodesic path is not meaningful, but for all positive $\epsilon$ a geodesic path does exist. When the gradient norms are small, we prefer the linear penalty incurred by the $\epsilon$ term, as discussed in Sec. 3.1, so that small amounts of noise in smooth regions do not bias the measure.

In practice when $\epsilon$ is positive, these properties are consistent, but the geodesic cost is influenced by its entire path on the manifold. The cost to rotate by an angle $\theta$ when $r_1 = r_2$ is essentially constant, regardless of the magnitude of $r_1$. The cost to go from $(r_1, \theta_1)$ to $(r_2, \theta_2)$ is close to the cost of rotating a constant $r$ by $\theta_2 - \theta_1$ plus the cost of scaling from $r_1$ to $r_2$ without any rotation.

### 4.4 Deformation Insensitivity

The algorithm presented above provides a way to compare images that is insensitive to changes in scene illumination. We now claim that this algorithm can also handle moderate amounts of deformation. Expanding our function to include several scales $s$ of wavelet coefficients, where larger scales correspond to coarser levels of the decomposition, the function to be minimized is now

$$I_{\text{geod}}(t) = \frac{1}{2} \sum_{m,n} \sum_{s} \underset{H(t),V(t)}{\arg\min} \lambda_s \int_0^1 \frac{H'^2 + V'^2}{H^2 + V^2 + \epsilon^2} dt. \tag{18}$$

We choose the weighting coefficient to be $\lambda_s = 2^s$, which we justify from its similarity to the Wavelet Earth Mover's Distance weighting as discussed below, and because it was observed empirically to provide the most accurate results. Using several scales increases accuracy because we can now consider both global image properties from the coarse scales, and edge details from the finer scales. In our experiments we use the first three scales of the wavelet transform. The resulting algorithm now involves a separate geodesic curve construction and integration for each scale and location.

Further, we argue that simply using wavelets adds moderate deformation insensitivity. Deformations within the support of each wavelet basis function are handled together during the wavelet transform, so deformations localized to these region have little overall impact on the wavelet coefficients. A similar observation was made previously when the Earth Mover's Distance was explored in the wavelet domain. The Earth Mover's Distance (EMD) algorithm [14] provides a way to compare two distributions by measuring the distance and quantity of "mass" that must be moved in order to convert one distribution into the other, where "mass" is thought of as whatever is populating the bins of a histogram. This similarity measure captures certain types of deformation, where no particular geometric structure is preserved or favored, but local changes in mass cost significantly less than global structure modifications.

The Wavelet EMD [15] approximates the Earth Mover's Distance in the wavelet domain, converting an algorithm of complexity $O(n^3 \log n)$ into an $O(n)$ algorithm without any significant performance difference, where $n$ is the number of points in an image, and its cost depends on wavelet coefficients at all scales. At each individual scale, it limits the distance individual mass units can move to the span of the wavelet at that scale. The weighting on the magnitude of each scales' wavelet coefficients in the distance calculation is $2^{2s} = 4^s$, similar to the weighting we incorporated into

our multiscaled cost function (18), where our base is 2 instead of 4. When the image gradients are small, our proposed cost function is essentially linear, as discussed in Sec. 3.1, meaning that it behaves similarly to the Wavelet EMD, and we understand how our new metric is able to handle moderate amounts of deformation, as this is the purpose of the Earth Mover's Distance. When the image gradients are larger, the new metric becomes more exponential, which allows the image comparison to be penalized less heavily when large lighting changes are present.

## 5 The Faster Algorithm

We will now discuss how to optimize our calculations to create a very computationally efficient algorithm. For any given pair of starting input values $H(0), V(0)$ and ending input values $H(1), V(1)$, the geodesic curve connecting them is always the same, so the cost of this input is always the same. This means that the geodesic curves can be calculated and integrated offline, and at run time the only computation that has to be performed is to look up the value of the integral for the given $(H(0), V(0), H(1), V(1))$. To further reduce the amount of space and time required, at every point we convert the input $(H(0), V(0), H(1), V(1))$ into polar coordinates, $(r_1, r_2, \theta_1, \theta_2)$, and then rotate so that $\theta_2 = 0$, as these rotated values preserve the relation between the points and will result in the same output cost. This allows us to generate a lookup table of integral values that depends on only three values $(r_1, r_2, \Delta\theta)$ instead of four.

We discretize the space of $r$ values into 40 bins of exponentially increasing size in the range $[0, 1.5]$, as this is the range of wavelet coefficient values observed in practice for images with pixel values in $[0, 1]$, with coarser scales generally consisting of smaller values. We used $\epsilon = 0.01$ in our experiments. The space of $\Delta\theta$ values we discretize into 80 bins of uniform size in the range $[0, 2\pi)$. The resulting costs are symmetric about $\Delta\theta = \pi$, so we really only have to store the first half of these values, and the lookup table to be stored is of dimension $40 \times 40 \times 40$. The online calculation at each location $(m, n)$ in wavelet space consists of converting $(H(0), V(0), H(1), V(1))$ into polar coordinates $(r_1, r_2, \Delta\theta)$, looking up the corresponding integral value in the table, and adding this value to the overall cost being calculated.

This calculation is limited principally by the speed at which a given machine can perform a lookup in a $40 \times 40 \times 40$ array, which is in general a very fast operation. The cost of this calculation is on the order of milliseconds, fast enough to use in practice when many image comparisons must be computed very quickly. On a 3.16 GHz machine running MATLAB in serial, this takes on average $1.3 \times 10^{-3}$ seconds for a pair of images with 5000 pixels each. We emphasize that the lookup table is application-independent; once it has been generated, which takes 1.5 hours, the same table can be used for any pair of images from any domain.

## 6 Experiments

One class of object that is regularly presented with large amounts of lighting variation and moderate amounts of deformation is the human face. Although nothing in our algorithm is specific to faces, the limited amount of deformation present with expression

(a) neutral (b) expressions (c) lightings

**Fig. 5.** The variations of one person from the AR Face Database [12].

change, along with potentially high variations due to lighting change, make them a relevant application of our work. We use a common face dataset studied for this problem, the subset of the AR Face Database [12] that contains variation in expression and lighting. We reduce the size of the standard cropped AR images by a factor of two in each dimension, as face recognition algorithms routinely perform the best on images of this scale, and so the images we compare are $83 \times 59$ pixels in dimension, and are smoothed slightly before processing. We use a neutral face from each of the 100 people in the dataset as gallery images, and the three variations in expression and the three variations in lighting for each person comprise the test set; see Fig. 5. The identity of each test image is determined by the gallery image returning the lowest cost pairing.

The algorithm presented here is a fast method for comparing images in the presence of lighting change and moderate deformations, and so we compare to other lighting and deformation insensitive algorithms that do not require training data. It was shown in [5] that the Gradient Direction method, described in Sec. 3, consistently performs better than the other standard pixel-based lighting-insensitive methods (Self-Quotient, luminance map estimation, Eigenphases, Whitening), so we compare to Gradient Direction. We also compare to the results of the Deformation and Lighting Insensitive metric ($E_{\text{DLI}}$) [7], and we expect our calculations to be much faster. Other works that present a cost function to handle both lighting change and deformations include that of [20], which calculates image point correspondences using edge maps and Gabor jet information, and [16] which uses mutual information to combine binary edge features with grayscale information. We also compare to simple image differencing and to normalized cross-correlation [9], where the template is a full image, as these methods are frequently used to compare images when many comparisons must be completed very fast. As our method is based on an $L^2$ metric, we use the $L^2$ norm on each of these measures for valid comparison. Results on the AR Face Database are presented in Table 1 for both algorithm speed and accuracy.

We see that our method achieves more accurate results than the Gradient Direction method on the lighting variation images, and significantly more accurate results on the expression variation images, as expected. This confirms the insensitivity of the method to lighting change, with the added benefit that we are able to construct geodesic information which allows for meaningful extensions such as mapping and interpolating large image variations. The accuracy of the method is also above that of the $E_{\text{DLI}}$ work where the lighting metric was first presented, which also handled deformations explicitly, and our calculations here are $10^3$ times faster than that work, making our method useful in template matching applications where the original method was prohibitively slow.

The previous best results on this dataset, as far as the authors are aware, were produced by Pixel-Level Decisions in [6], where simple thresholding was applied to pixel differences of a chosen image property. Standard deviation calculated within a window

| *Method* | *Time (sec)* | *Expression* | *Lighting* | *Overall* |
|---|---|---|---|---|
| Image Differencing | $3.1 \times 10^{-5}$ | 83.0% | 9.0% | 46.0% |
| Normalized Cross-Correlation [9] | $7.2 \times 10^{-3}$ | 84.0% | 59.3% | 71.7% |
| Significant Jet Point [20] | – | 80.8% | 91.7% | 86.3% |
| Binary Edge Feature and MI [16] | – | 78.5% | 97.0% | 87.8% |
| Gradient Direction [5] | $3.8 \times 10^{-4}$ | 85.0% | 95.3% | 90.2% |
| $E_{\text{DLI}}$ [7] | $1.0 \times 10^{0}$ | 89.6% | 98.9% | 94.3% |
| **Proposed Method** | $1.3 \times 10^{-3}$ | 93.7% | 96.7% | 95.2% |
| Pixel Level Decisions [6] | $5.6 \times 10^{-4}$ | 98.0% | 94.0% | 96.0% |
| **Proposed Method thresholded** | $1.3 \times 10^{-3}$ | 97.3% | 97.0% | 97.2% |

**Table 1.** Identification results on the AR Face Database. The *Time* column reports the MATLAB calculation time of a single image pair comparison in seconds, except in two cases where time was not reported and we were unable to reproduce the authors' results.

around each pixel was the property that provided the best results. The differences between these standard deviations at every pixel location in each image were computed, and the total number of pixel differences less than a pre-determined threshold were counted for the final similarity value. We present these results here to demonstrate that the surprisingly strong results achieved from this extremely simple algorithm can be applied to other pixel-based methods, and we use a similar thresholding step on our results as well. [6] also suggests compensating for local error by repeating the procedure with the images shifted a few pixels in every direction, but we do not compare these results as they are not relevant to the ideas in this paper. However, this repeated shifting could be applied to improve the results of any of the these methods. As the threshold value for our point costs in wavelet space, we use the cost value that counts the lowest $20\%$ of the point costs across all images, as this was the value used by [6]. The exact threshold value is not sensitive, and we observed that all values thresholding $9\%$ to $47\%$ of the costs resulted in overall accuracies within $1\%$ of each other, and the ideal threshhold on this dataset, if hand-picked, results in an overall accuracy of $98.0\%$. We see in Table 1 that this simple thresholding extension removes $58.6\%$ of the errors in our method.

The proposed algorithm performs well with variations in lighting, and also handles moderate amounts of deformation. Many methods perform very poorly on the scream category of this database, but the multiscaled method presented here achieved $83.0\%$ accuracy in this case, and $93.0\%$ with thresholding, higher than either Gradient Direction ($57.0\%$) or the $E_{\text{DLI}}$ metric ($79.6\%$), which was designed to handle deformations.

Not only does the proposed algorithm produce accurate identification results, but the computation time required is extremely small. We emphasize that no training data or learning stage is required for the method proposed in this paper.

## 7 Conclusion

We have presented a fast algorithm for handling illumination changes and moderate deformations applicable to any class of images. Geodesic distances were calculated between pairs of images, as defined on an image manifold given structure by an illumination-insensitive metric that was based on the change in image gradients. The metric was cal-

culated in the wavelet domain, where each point location contributed independently to the overall image comparison cost, allowing geodesic costs to be computed extremely efficiently using a pre-calculated lookup table. Using wavelets at multiple scales allowed for insensitivity to moderate deformations in a manner similar to the Wavelet Earth Mover's Distance. Strong results were presented on the AR Face Database, where our algorithm is seen to be both extremely fast and accurate. Using geodesics to calculate image comparisons instead of simple pixel differences allows our method to be incorporated into a wide array of applications where having information along a morphing path is relevant. Because this algorithm is so fast, it could also be applied successfully in situations where Normalized Cross-Correlation is often used, where many image comparisons must be computed in a very short amount of time.

**Acknowledgments:** This work was supported by US Office of Naval Research MURI Grant N00014-08-10638, and by National Science Foundation Grant No. 0915977.

## References

1. M. Beg, M. Miller, A. Trouvé, and L. Younes. Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *IJCV*, 61:139–157, 2005.
2. J. Bruna and S. Mallat. Classification with Scattering Operators. *CVPR*, 2011.
3. R. Courant and D. Hilbert. *Methods of Mathematical Physics, volume I*, chapter IV. Interscience Publishers, Inc., 1955.
4. M. do Carmo. *Riemannian Geometry*. Birkhuser, 1992.
5. R. Gopalan and D. Jacobs. Comparing and Combining Lighting Insensitive Approaches for Face Recognition. *CVIU*, 114:135–145, 2010.
6. A. P. James. Pixel-Level Decisions Based Robust Face Image Recognition. In M. Oravec, editor, *Face Recognition*, chapter 5, pages 65–86. INTECH, 2010.
7. A. Jorstad, D. Jacobs, and A. Trouvé. A Deformation and Lighting Insensitive Metric for Face Recognition Based on Dense Correspondences. *CVPR*, 2011.
8. Y. LeCun, F. Huang, and L. Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. *CVPR*, 2004.
9. J. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, 1995.
10. D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60, 2004.
11. S. Mallat. *A Wavelet Tour of Signal Processing*. Elsevier, 2009.
12. A. Martinez and A. C. Kak. PCA versus LDA. *PAMI*, 23:228 – 233, 2001.
13. S. Negahdaripour. Revised Definition of Optical Flow: Integration of Radiometric and Geometric Cues for Dynamic Scene Analysis. *PAMI*, 20:961–979, 1998.
14. Y. Rubner, C. Tomasi, and Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *IJCV*, 2000.
15. S. Shirdhonkar and D. Jacobs. Approximate Earth Movers Distance in Linear Time. *CVPR*, 2008.
16. J. Song, B. Chen, W. Wang, and X. Ren. Face Recognition by Fusing Binary Edge Feature and Second-Order Mutual Information. In *IEEE Conf. on Cybernetics and Intelligent Systems*, pages 1046–1050, 2008.
17. T. Tung and T. Matsuyama. Dynamic Surface Matching by Geodesic Mapping for 3D Animation Transfer. *CVPR*, 2010.
18. P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision. *CVPR*, 2008.
19. L. Younes. *Shapes and Diffeomorphisms*. Springer, 2010.
20. S. Zhao and Y. Gao. Significant Jet Point For Facial Image Representation and Recognition. *International Conference on Image Processing*, pages 1664–1667, 2008.