A Schrödinger Equation for the Fast Computation of Approximate Euclidean Distance Functions

Karthik S. Gurumoorthy and Anand Rangarajan

Dept. of Computer and Information Science and Engineering
University of Florida, Gainesville, FL, USA

Abstract. Computational techniques adapted from classical mechanics and used in image analysis run the gamut from Lagrangian action principles to Hamilton-Jacobi field equations: witness the popularity of the fast marching and fast sweeping methods which are essentially fast Hamilton-Jacobi solvers. In sharp contrast, there are very few applications of quantum mechanics inspired computational methods. Given the fact that most of classical mechanics can be obtained as a limiting case of quantum mechanics (as Planck's constant h tends to zero), this paucity of quantum mechanics inspired methods in image analysis is surprising. In this work, we derive relationships between nonlinear Hamilton-Jacobi and linear Schrödinger equations for the Euclidean distance function problem (in $1D$, $2D$ and $3D$). We then solve the Schrödinger wave equation instead of the corresponding Hamilton-Jacobi equation. We show that the Schrödinger equation has a closed form solution and that this solution can be efficiently computed in $O(N \log N)$, N being the number of grid points. The Euclidean distance can then be recovered from the wave function. Since the wave function is computed for a small but non-zero h, the obtained Euclidean distance function is an approximation. We derive analytic bounds for the error of the approximation and experimentally compare the results of our approach with the exact Euclidean distance function on real and synthetic data.

1 Introduction

Image analysis [1,2] has a tradition of importing and adapting a host of classical physics based approaches including Lagrangian based variational principles and their associated Euler-Lagrange equations [3], Hamiltonian dynamics [4] and more recently Hamilton-Jacobi based methods [5]. Approaches in image analysis do not strictly adhere to the classical mechanics sequence [6] of i) first specifying a Lagrangian action principle, ii) deriving the corresponding Euler-Lagrange equation, iii) employing a Legendre transformation to convert the Lagrangian dynamics to a first-order Hamiltonian dynamics, and finally, iv) using a canonical transformation to derive the Hamilton-Jacobi equation whose solution also yields a solution to the original variational problem. Instead, most research in image analysis uses a combination of one or more of these four approaches depending on the problem at hand. For example, in surface reconstruction [3], a popular approach consists of writing a variational form and then finding a solution using preconditioned conjugate gradient or quasi-Newton type iterative methods. While we notice a plethora of classical mechanics inspired techniques in image analysis, the

same cannot be said about quantum mechanics. Despite the well known fact that most of classical mechanics is a limiting case of quantum mechanics as Planck's constant $\hbar \to 0$ [7], there is very little application of quantum mechanical principles in image analysis problems. Rather than speculate on the reasons for this dearth of applications, we wish to point out that in this paper, we are primarily interested in exploiting a concrete relationship between the classical, non-linear Hamilton-Jacobi equation and the quantum, linear Schrödinger equation. We feel that focusing more narrowly on this relationship (which will become obvious as we proceed) is more productive than dwelling on the mysterious and specifically quantum mechanical issues of i) interpretation of the wave function, ii) role of probabilities and, iii) the problem of measurement. While these issues are certainly important, they do not play any role in this paper. In summary, we are mainly interested in exploiting the relationship between the Schrödinger and Hamilton-Jacobi equations in order to derive computationally efficient algorithms which are applicable in image analysis problems where Hamilton-Jacobi theory is used.

In the theoretical physics literature, a Schrödinger wave equation at the energy state E has the form $\psi(X,t) = \phi(X) \exp(\frac{iEt}{\hbar})$, ($\hbar \equiv \frac{h}{2\pi}$) [8], where $\phi(X)$—the stationary state wave function—is the eigenstate of the Hamiltonian operator H corresponding to the eigenvalue E. When the Hamilton-Jacobi scalar field S^* appears as the exponent of the stationary state wave function, specifically $\phi(X) = \exp(\frac{-S^*(X)}{\hbar})$, and if $\phi(X)$ satisfies the linear Schrödinger equation, namely $H\phi = E\phi$, we show that as $\hbar \to 0$, S^* satisfies the Hamilton-Jacobi equation for a carefully chosen problem. The novel aspect is that a nonlinear Hamilton-Jacobi equation is obtained in the limit as $\hbar \to 0$ of a linear Schrödinger equation. Consequently, instead of solving the Hamilton-Jacobi equation, one can solve the Schrödinger wave equation (taking advantage of its linearity), and then compute an approximate S^* for small values of $\hbar$. This computational procedure would be approximately equivalent to solving the original Hamilton-Jacobi equation.

With the basic setup in place, we now turn our attention to an actual application. Our goal is to apply the Schrödinger formalism to a well known problem that has been successfully attacked by Hamilton-Jacobi theory. To this end, we choose the problem of computing Euclidean distance functions on a grid for a given set of points where the task is to assign at each grid point a value corresponding to the Euclidean distance to its nearest neighbor from the given point-set. The literature is replete with elegant and pioneering works which have successfully solved this problem. To name a few, the well known fast marching [9] and fast sweeping [10] methods are essentially $O(N \log N)$ Hamilton-Jacobi based algorithms where N is the number of grid points. [The fast sweeping method has an added advantage in that the algorithm appears (empirically)[1] to be $O(N)$.] These techniques focus on directly solving the non-linear eikonal equation [4], whereas our approach shows that one can instead solve a linear equation and obtain the solution to the non-linear eikonal equation in the limit as $\hbar \to 0$. The important connection between the Schrödinger wave equation and the Hamilton-Jacobi equation [7] is illuminated during the process. In the more traditional (computer science) algorithms literature, computational geometry inspired techniques like Voronoi diagrams and KD-Trees [11] have also solved this problem. However, computing Voronoi diagrams or

---

[1] After a careful reading of [10], it is still unclear to us if the fast sweeping method is formally rather than empirically $O(N)$.

building data structures for KD-Trees in $3D$ and higher dimensions is expensive and the $O(N \log N)$ complexity is not retained at higher dimensions [11]. Our technique (which as we shall see is based on application of the fast Fourier transform (FFT) [12]) is very simple and elegant and remains $O(N \log N)$ irrespective of the spatial dimension.

The basic question asked and answered in this paper is: Can we design a Schrödinger equation that computes an exponentiated Euclidean distance function on a grid? The distance function is obtained from the exponent of the Schrödinger wave function. A naïve approach to solve this Euclidean distance problem would be to visit *every* grid point and compute the Euclidean distance to *all* members of the point-set and pick the smallest distance. The complexity of this naïve approach is obviously related to the product of the number of grid points and the cardinality of the point-set. The fast sweeping and fast marching methods avoid this naïve complexity and as we shall see, so does the Schrödinger wave function approach.

2 Euclidean distance functions

We now describe the Euclidean distance function problem. Given a point-set $Y = \{Y_k \in \mathbb{R}^D, k \in \{1, \ldots, K\}\}$ where D is the dimensionality of the point-set and a set of equally spaced Cartesian grid points X, the Euclidean distance function problem requires us to assign

$$S^*(X) = \min_{k} \|X - Y_k\| \tag{1}$$

with the Euclidean norm used in (1). If efficient computation is a non-issue, this is a simple problem. We visit each grid point in the set X and compute the distance to every point $Y_k, \forall k \in \{1, \ldots, K\}$ and assign $S^*(X)$ the minimum distance. If the number of grid points is N, the naïve complexity is $O(NKD)$.

2.1 Hamilton-Jacobi formulation

The Hamilton-Jacobi equation approach to the Euclidean distance function problem stems from considering the following variational problem which in $2D$ is

$$I[q] = \int_{t_o}^{t_1} L(q_1, q_2, \dot{q}_1, \dot{q}_2, t) dt \tag{2}$$

where the Lagrangian L is defined as

$$L(q_1, q_2, \dot{q}_1, \dot{q}_2, t) \equiv \frac{1}{2}(\dot{q_1}^2 + \dot{q_2}^2). \tag{3}$$

Defining $p_i \equiv \frac{\partial L}{\partial \dot{q}_i}$ and applying a Legendre transformation [6] [by inverting $p_i = \frac{\partial L}{\partial \dot{q}_i}$ to get the function $\dot{q}_i(q_1, q_2, p_1, p_2, t) \, \forall i$], we define the Hamiltonian of the system as

$$H(q_1, q_2, p_1, p_2, t) \equiv p_1 \dot{q}_1(q_1, q_2, p_1, p_2, t) + p_2 \dot{q}_2(q_1, q_2, p_1, p_2, t)$$
$$- L(q_1, q_2, \dot{q}_1(q_1, q_2, p_1, p_2, t), \dot{q}_2(q_1, q_2, p_1, p_2, t), t) \tag{4}$$

which for the Euclidean distance function problem turns out to be

$$H(q_1, q_2, p_1, p_2, t) = \frac{1}{2}(p_1^2 + p_2^2). \tag{5}$$

The Hamilton-Jacobi equation is obtained via a canonical transformation [6] of the Hamiltonian. In the $2D$ case, it is

$$\frac{\partial S}{\partial t} + H(q_1, q_2, \frac{\partial S}{\partial q_1}, \frac{\partial S}{\partial q_2}, t) = 0 \tag{6}$$

where we have replaced [7] the generalized momentum variables p_i with

$$\frac{\partial S}{\partial q_i} = p_i = \frac{\partial L}{\partial \dot{q}_i}. \tag{7}$$

Since the Hamiltonian in (5) is a constant *independent* of time, equation (6) can be simplified to the static Hamilton-Jacobi equation. By separation of variables, we get

$$S(q_1, q_2, t) = S^*(q_1, q_2) - Et \tag{8}$$

where E is the total energy of the system and $S^*(q_1, q_2)$ is called the Hamilton's characteristic function [13]. Using the definition of p_i from (7) in (5) and observing that $\frac{\partial S}{\partial q_i} = \frac{\partial S^*}{\partial q_i}$, we get

$$\frac{1}{2}\left[\left(\frac{\partial S^*}{\partial q_1}\right)^2 + \left(\frac{\partial S^*}{\partial q_2}\right)^2\right] = E. \tag{9}$$

Choosing the energy E to be $\frac{1}{2}$, we obtain

$$\|\nabla S^*\| = 1 \tag{10}$$

which is the well known eikonal equation [4] where the forcing term is 1. S^* is the required Hamilton-Jacobi scalar field which is efficiently obtained by the fast sweeping and fast marching methods.

2.2 A Schrödinger wave equation for computing Euclidean distance functions

We now derive and solve a Schrödinger wave equation for this Euclidean distance function problem instead of solving the non-linear eikonal equation.

The Schrödinger wave equation is written as [8]

$$i\hbar\frac{\partial \psi}{\partial t} = H\psi \tag{11}$$

where $\psi(X, t)$ is the wave function and H is the Hamiltonian operator obtained by first quantization[2]—where the momentum variables p_i are replaced with the operator $\frac{\hbar}{i}\frac{\partial}{\partial x_i}$. Using the definition of our Hamiltonian from (5), ψ satisfies (in $2D$)

$$\frac{\hbar}{i}\frac{\partial \psi}{\partial t} = \frac{\hbar^2}{2}\left(\frac{\partial^2 \psi}{\partial x_1^2} + \frac{\partial^2 \psi}{\partial x_2^2}\right). \tag{12}$$

---

[2] First quantization is still mysterious. For an informal but illuminating treatment, please see http://math.ucr.edu/home/baez/categories.html.

Using separation of variables $\psi(X, t) = \phi(X)f(t)$, we get

$$\frac{\hbar}{i}\frac{\dot{f}}{f} = \frac{\hbar^2}{2}\frac{\nabla^2\phi}{\phi} = E \tag{13}$$

where E is the energy state of the system. By choosing the energy of the system to be $\frac{1}{2}$ as before, we get

$$f(t) = \exp\left(\frac{it}{2\hbar}\right) \tag{14}$$

and hence

$$\psi(X, t) = \phi(X)\exp\left(\frac{it}{2\hbar}\right) \tag{15}$$

where $\phi(X)$ satisfies the equation

$$\hbar^2\nabla^2\phi = \phi. \tag{16}$$

2.3 Deriving the eikonal equation from the Schrödinger equation

We now show that when $\phi = \exp(\frac{-S^*}{\hbar})$ and satisfies (16), S^* asymptotically satisfies the eikonal equation (10) as $\hbar \to 0$. We show this for the $2D$ case but the generalization to higher dimensions is straightforward.

When $\phi(x_1, x_2) = \exp(\frac{-S^*(x_1,x_2)}{\hbar})$, the first partials of ϕ are

$$\frac{\partial\phi}{\partial x_1} = -\frac{1}{\hbar}\exp\left(\frac{-S^*}{\hbar}\right)\frac{\partial S^*}{\partial x_1}, \frac{\partial\phi}{\partial x_2} = -\frac{1}{\hbar}\exp\left(\frac{-S^*}{\hbar}\right)\frac{\partial S^*}{\partial x_2}. \tag{17}$$

The second partials needed for the Laplacian are

$$\frac{\partial^2\phi}{\partial x_1^2} = \frac{1}{\hbar^2}\exp\left(\frac{-S^*}{\hbar}\right)\left(\frac{\partial S^*}{\partial x_1}\right)^2 - \frac{1}{\hbar}\exp\left(\frac{-S^*}{\hbar}\right)\frac{\partial^2 S^*}{\partial x_1^2},$$

$$\frac{\partial^2\phi}{\partial x_2^2} = \frac{1}{\hbar^2}\exp\left(\frac{-S^*}{\hbar}\right)\left(\frac{\partial S^*}{\partial x_2}\right)^2 - \frac{1}{\hbar}\exp\left(\frac{-S^*}{\hbar}\right)\frac{\partial^2 S^*}{\partial x_2^2}. \tag{18}$$

From this, equation (16) can be rewritten as

$$\left(\frac{\partial S^*}{\partial x_1}\right)^2 + \left(\frac{\partial S^*}{\partial x_2}\right)^2 - \hbar\left(\frac{\partial^2 S^*}{\partial x_1^2} + \frac{\partial^2 S^*}{\partial x_2^2}\right) = 1 \tag{19}$$

which in simplified form is

$$\|\nabla S^*\|^2 - \hbar\nabla^2 S^* = 1. \tag{20}$$

The additional $\hbar\nabla^2 S^*$ term [relative to (10)] is referred to as the viscosity term [9]. (Note that this term emerges naturally from the Schrödinger equation derivation—an intriguing result.) Since $|\nabla^2 S^*|$ is bounded, as $\hbar \to 0$, (20) tends to

$$\|\nabla S^*\| = 1 \tag{21}$$

which is the original eikonal equation (10) for the Euclidean distance function problem. This relationship motivates us to solve the linear Schrödinger equation (16) instead of the non-linear eikonal equation and then compute the distance function via

$$S^*(X) = -\hbar\log\phi(X). \tag{22}$$

3 Closed form solutions for the approximate Euclidean distance function and proofs of convergence

We now derive the closed form solution for $\phi(X)$ (in $1D$, $2D$ and $3D$) satisfying equation (16) and hence for $S^*(X)$ by (22) and observe that we get the *actual* Euclidean distance function in the limit as $\hbar \to 0$.

In order to satisfy the condition that $S^*(Y_k) = 0, \forall Y_k, k \in \{1, \ldots, K\}$, we consider the forced version of equation (16) which is

$$-\hbar^2 \nabla^2 \phi + \phi = \sum_{k=1}^{K} \delta(X - Y_k). \tag{23}$$

Using a Green's function approach [14] (where the form of the solution depends on the number of spatial dimensions), we can write expressions for the solution ϕ. Below, let $r = \min_k \|X - Y_k\|$—the actual Euclidean distance function at the grid point X.

1D:

In $1D$, the solution [14] for ϕ is

$$\phi(X) = \frac{1}{2\hbar} \sum_{k=1}^{K} \exp\left(\frac{-|X - Y_k|}{\hbar}\right). \tag{24}$$

Using the relationship in (22), we get

$$S^*(X) = -\hbar \log \sum_{k=1}^{K} \exp\left(\frac{-|X - Y_k|}{\hbar}\right) + \hbar \log(2\hbar). \tag{25}$$

Observe that

$$S^*(X) \leq -\hbar \log \exp\left(\frac{-r}{\hbar}\right) + \hbar \log(2\hbar)$$
$$= r + \hbar \log(2\hbar). \tag{26}$$

Also,

$$S^*(x) \geq -\hbar \log\left[K \exp\left(\frac{-r}{\hbar}\right)\right] + \hbar \log(2\hbar)$$
$$= -\hbar \log K + r + \hbar \log(2\hbar). \tag{27}$$

As $\hbar \to 0$, $\hbar \log K \to 0$ *and* $\hbar \log \hbar \to 0$. Furthermore, we see from (26) and (27) that

$$\lim_{\hbar \to 0} S^*(X) = r. \tag{28}$$

2D:

In $2D$, the solution [14] for ϕ is

$$\phi(X) = \frac{1}{2\pi\hbar^2} \sum_{k=1}^{K} K_0\left(\frac{\|X - Y_k\|}{\hbar}\right) \tag{29}$$

where K_0 is the modified Bessel function of the second kind. Using (22), we get

$$S^*(X) = -\hbar \log \sum_{k=1}^{K} K_0\left(\frac{\|X - Y_k\|}{\hbar}\right) + \hbar \log(2\pi\hbar^2). \tag{30}$$

Then,

$$S^*(X) \leq -\hbar \log K_0\left(\frac{r}{\hbar}\right) + \hbar \log(2\pi\hbar^2). \tag{31}$$

Using the relation $K_0(\frac{r}{\hbar}) \geq \frac{\exp(-\frac{r}{\hbar})}{\sqrt{\frac{r}{\hbar}}}$ when $\frac{r}{\hbar} \geq 0.5$, we get

$$S^*(X) \leq -\hbar \log \left[\sqrt{\frac{\hbar}{r}} \exp\left(\frac{-r}{\hbar}\right)\right] + \hbar \log(2\pi\hbar^2)$$

$$= -\hbar \log \sqrt{\frac{\hbar}{r}} + r + \hbar \log(2\pi\hbar^2). \tag{32}$$

Moreover

$$S^*(X) \geq -\hbar \log \left[K K_0\left(\frac{-r}{\hbar}\right)\right] + \hbar \log(2\pi\hbar^2). \tag{33}$$

Using the relation $K_0(\frac{r}{\hbar}) \leq \exp(\frac{-r}{\hbar})$ when $\frac{r}{\hbar} \geq 1.5$, we get

$$S^*(X) \geq -\hbar \log \left[K \exp\left(\frac{-r}{\hbar}\right)\right] + \hbar \log(2\pi\hbar^2)$$

$$= -\hbar \log K + r + \hbar \log(2\pi\hbar^2). \tag{34}$$

As $\hbar \to 0$, $\hbar \log K \to 0$, $\hbar \log r \to 0$ *and* $\hbar \log \hbar \to 0$. Furthermore, we see from (32) and (34) that

$$\lim_{\hbar \to 0} S^*(X) = r. \tag{35}$$

3D:
In $3D$, the solution [14] for ϕ is based on the modified spherical Bessel function of the second kind:

$$\phi(X) = \frac{1}{4\pi\hbar^2} \sum_{k=1}^{K} \frac{\exp\left(\frac{-\|X - Y_k\|}{\hbar}\right)}{\|X - Y_k\|}. \tag{36}$$

Using (22),

$$S^*(X) = -\hbar \log \sum_{k=1}^{K} \frac{\exp\left(\frac{-\|X - Y_k\|}{\hbar}\right)}{\|X - Y_k\|} + \hbar \log\left(4\pi\hbar^2\right). \tag{37}$$

Then,

$$S^*(X) \leq -\hbar \log \frac{\exp\left(\frac{-r}{\hbar}\right)}{r} + \hbar \log(4\pi\hbar^2)$$

$$= r + \hbar \log r + \hbar \log\left(4\pi\hbar^2\right). \tag{38}$$

Also,

$$S^*(X) \geq -\hbar \log \left[K \frac{\exp\left(\frac{-r}{\hbar}\right)}{r} \right] + \hbar \log(4\pi\hbar^2)$$

$$= -\hbar \log K + r + \hbar \log r + \hbar \log(4\pi\hbar^2). \qquad (39)$$

As $\hbar \to 0$, $\hbar \log K \to 0$, $\hbar \log r \to 0$ *and* $\hbar \log \hbar \to 0$. Furthermore, we see from (38) and (39) that

$$\lim_{\hbar \to 0} S^*(X) = r. \qquad (40)$$

Hence, we have shown that (in $1D$, $2D$ and $3D$), the closed form solution for ϕ guarantees that S^* approaches the true Euclidean distance function in the limit $\hbar \to 0$.

4 Bounding the error of the approximate Euclidean distance function

The solution for ϕ in $1D$, $2D$ and $3D$ motivates us to compute the function

$$\tilde{\phi}(X) = \sum_{k=1}^{K} \exp\left(\frac{-\|X - Y_k\|}{\hbar}\right) \qquad (41)$$

(instead of computing ϕ) and then to compute the approximate distance function

$$\tilde{S}^*(X) = -\hbar \log \tilde{\phi}(X) \qquad (42)$$

(instead of computing S^*). The reasons are two-fold. Firstly, $\tilde{\phi}$ can be computed efficiently in $O(N \log N)$ using the fast Fourier transform (FFT) [12] as explained in the subsequent section. Secondly, $\lim_{\hbar \to 0} \tilde{S}^*(X) = r$, as shown below, where r is the true Euclidean distance function value at the grid point X ($r = \min_k \|X - Y_k\|$), and this is the $\lim_{\hbar \to 0} S^*(X)$ as seen from the previous section. Hence, for small values of $\hbar$, $\tilde{S}^*(X)$ is a very good approximation to $S^*(X)$.

As $\hbar \to 0$, $\sum_{k=1}^{K} \exp\left(\frac{-\|X - Y_k\|}{\hbar}\right)$ can be approximated as $\exp\left(\frac{-r}{\hbar}\right)$. Hence $\tilde{S}^*(X) \approx -\hbar \log \exp\left(\frac{-r}{\hbar}\right) = r$. The bound derived below between $\tilde{S}^*(X)$ and r also unveils the proximity between the computed and the actual Euclidean distance function.

Note from (41) that

$$\tilde{S}^*(X) \leq -\hbar \log \exp\left(\frac{-r}{\hbar}\right) = r. \qquad (43)$$

Also, observe that

$$\tilde{S}^*(X) \geq -\hbar \log \left[K \exp\left(\frac{-r}{\hbar}\right) \right]$$

$$= -\hbar \log K + r \qquad (44)$$

and hence,

$$r - \tilde{S}^*(X) \leq \hbar \log K. \tag{45}$$

From (43) and (45),

$$|r - \tilde{S}^*(X)| \leq \hbar \log K. \tag{46}$$

Equation (46) shows that as $\hbar \to 0$, $\tilde{S}^*(X) \to r$. It is worth commenting that the bound $\hbar \log K$ is actually very tight as (i) it scales only as the logarithm of the cardinality of the point-set (K) and (ii) it can be made arbitrarily small by choosing a small but non-zero value of $\hbar$.

5 Efficient computation of the approximate Euclidean distance function

The motivation for computing $\tilde{\phi}$ instead of ϕ is the fact that the direct computation of ϕ at the N grid locations is $O(NK)$ which is $O(N^2)$ when the cardinality of the point-set is $O(N)$, whereas computing $\tilde{\phi}$ at the N grid locations can be done in $O(N \log N)$ using an FFT implementation [12]. The realization that the FFT can be employed to compute $\tilde{\phi}$ stems from the insight that the summation term, namely, $\sum_{k=1}^{K} \exp\left(\frac{-\|X-Y_k\|}{\hbar}\right)$ is actually the *discrete convolution* between the functions $f(X)$ and $g(X)$ with $f(X) = \exp\left(\frac{-\|X\|}{\hbar}\right)$ computed at the grid locations, and the function $g(X)$ which takes the value 1 at the point-set locations and 0 at other grid locations. By the convolution theorem [15], a discrete convolution can be obtained as the inverse Fourier transform of the product of two individual transforms, which for two $O(N)$ sequences can be computed in $O(N \log N)$ and hence $\tilde{\phi}$ can be determined efficiently at the N grid locations. One just needs to compute the discrete Fourier transform (DFT) of the sampled version of the functions $f(X)$ and $g(X)$, compute their point-wise product and then compute the inverse discrete Fourier transform. Taking the logarithm of the inverse discrete Fourier transform and multiplying it by $(-\hbar)$, gives the approximate Euclidean distance function. The algorithm is adumbrated in Table 1.

Table 1. Approximate Euclidean distance function algorithm

---

1. Compute the function $f(X) = \exp\left(\frac{-\|X\|}{\hbar}\right)$ at the grid locations.
2. Define the function $g(X)$ which takes the value 1 at the point-set locations and 0 at other grid locations.
3. Compute the FFTs of f and g, namely $F(u)$ and $G(u)$ respectively.
4. Compute the function $H(u) = F(u) * G(u)$.
5. Compute the inverse FFT of H which gives $\tilde{\phi}(X)$ at the grid locations.
6. Take the logarithm of $\tilde{\phi}(X)$ and multiply it by $(-\hbar)$ to get the approximate Euclidean distance function at the grid locations.

5.1 Computation of the approximate Euclidean distance function in higher dimensions

Our technique has a straightforward generalization to higher dimensions. Regardless of the spatial dimension, the approximate Euclidean distance function, $\tilde{S}^*$ can be computed by exactly following the steps delineated in the table above. It is worth mentioning that computing the discrete Fourier transform using the FFT is always $O(N \log N)$ *irrespective* of the number of spatial dimensions. Hence, for all dimensions, $\tilde{S}^*$ can be computed at the given N grid points, in $O(N \log N)$. This speaks for the scalability of our technique, which is generally not the case with other methods, for example KD-Trees [11].

6 Experiments

In this section, we show the efficacy of our technique by computing the approximate Euclidean distance function $\tilde{S}^*$ and comparing it to the actual Euclidean distance function S, first on randomly generated $2D$ point-sets and then on a set of bounded $3D$ grid points.

We began with a $2D$ grid consisting of points between $(-30, -30)$ and $(30, 30)$. Hence, the total number of grid points is $N = 61 \times 61 = 3721$. We randomly chose around 1000 grid locations as data points (point-set). Then $50,000$ experiments were run for values of $\hbar$ ranging from 0.1 to 0.5 in steps of 0.01. The errorbar plot in Figure 1 shows the mean and standard deviation of the *percentage* error at each value of $\hbar$. The error is less than 0.5% at $\hbar = 0.1$ demonstrating the algorithm's ability to compute accurate Euclidean distances.

Next, we took the Stanford bunny dataset[3] and used the coordinates of the data points on the model as our point-set locations. Since the input data locations need not be at integer locations, we scaled the space uniformly in all dimensions and rounded off the data so that the data lies at integer locations. The input data was also shifted so that it was approximately symmetrically located with respect to the x, y and z axis. We should comment that shifting the data doesn't affect the Euclidean distance function value and uniform scaling of all dimensions is also not an issue, as the distances can be rescaled once they are computed.

After these basic data manipulations, the cardinality of the point set was $K = 3019$ with the data confined to the cubic region $-16 \leq x \leq 16$, $-15 \leq y \leq 15$ and $-12 \leq z \leq 12$. Our grid consisted of the set of all integer locations within this cubic region. The number of grid locations was $N = 25575$. We computed the Euclidean distance function value at each of these grid locations using our technique for different values of $\hbar$ and compared it to the true Euclidean distance function value.

At small values of $\hbar$, $\exp\left(\frac{-\|X\|}{\hbar}\right)$ drops off very quickly and hence for grid locations which are far away from the point-set, the convolution done using FFT needs to be precise (without round-off error) for the computed distance to be meaningful. Such high precision support may not be available and hence our technique may produce erroneous

---

[3] This dataset is available at http://www.cc.gatech.edu/projects/large_models/bunny.html

results at these grid locations for very small values of $\hbar$. But at those grid locations which are close to the point-set, the accuracy of the computed distance improves as $\hbar$ is decreased. Hence, to circumvent this problem of choosing $\hbar$, we ran our technique for different values of $\hbar$ and chose the distance function values obtained at large values of $\hbar$ at those grid locations whose average computed distance is larger and vice versa.

When we ran our technique for the set of $\hbar \in \{0.1, 0.2, 0.3, 0.4\}$ and used $\{3, 6, 10\}$ respectively as the threshold of the average computed distance for choosing the appropriate distance function, it gave the following set of results. The maximum absolute difference between the actual and the computed Euclidean distance value over all the grid locations is 0.9066 and the average absolute difference is 0.1322. The accuracy is fairly high since the furthest grid point from the point-set is at a distance of 15.5242 and the average overall distance computed at the grid locations from the point-set is 3.5449. The average error is $\frac{0.1322}{3.5449} * 100 = 3.72\%$. This error can be lowered by using higher precision numerical methods for convolution [16].

We plotted the isosurface obtained by connecting the grid points which are at a distance of 0.5 from the point set, determined both by the true Euclidean distance function and our technique. Figure 2 shows the two surfaces. Notice the similarity between the two plots. It provides anecdotal visual evidence for the usefulness of our approach.

7 Discussion

In this paper, we have introduced a new approach to solving the non-linear eikonal equation (with a constant forcing term equal to 1). We have proved that the solution of the eikonal equation can be obtained as a limiting case of the solution to the corresponding linear Schrödinger wave equation. The key here is the embedding of the nonlinear Hamilton-Jacobi equation in a linear Schrödinger equation. Our Schrödinger wave equation formalism for solving the Euclidean distance function problem (which has been successfully attacked by pioneering Hamilton-Jacobi solvers such as the fast sweeping [10] and fast marching [9] methods) leverages this deep relationship between the two regimes of modern physics. In the future, we would like to solve the more general, static Hamilton-Jacobi equation using techniques inspired from quantum mechanics as a counterpart to classical mechanics based techniques. In all likelihood, this will involve direct discretization of the Schrödinger wave equation which was not required for the Euclidean distance function problem. We expect that the linearity of the Schrödinger equation will result in fast algorithms even in this more general setting.

References

1. Horn, B.K.P.: Robot vision. The MIT Press (1986)
2. Kimmel, R.: Numerical geometry of images: Theory, algorithms, and applications. Springer (2003)
3. Grimson, W.E.L.: An implementation of a computational theory of visual surface interpolation. Computer Vision, Graphics, and Image Processing **22**(1) (1983) 39–69
4. Siddiqi, K., Tannenbaum, A., Zucker, S.W.: A Hamiltonian approach to the eikonal equation. In: Energy Minimization Methods in Computer Vision and Pattern Recognition (EMM-CVPR). Volume LNCS 1654., Springer-Verlag (1999) 1–13
5. Kao, C.Y., Osher, S.J., Tsai, Y.H.: Fast sweeping methods for static Hamilton-Jacobi equations. SIAM Journal on Numerical Analysis **42**(6) (2004) 2612–2632
6. Goldstein, H., Poole, C.P., Safko, J.L.: Classical mechanics. Addison-Wesley (2002)
7. Butterfield, J.: On Hamilton-Jacobi theory as a classical root of quantum theory. In A. Elitzur, S. Dolev, N. Kolenda, eds.: Quo-Vadis Quantum Mechanics. Springer (2005) 239–274
8. Griffiths, D.J.: Introduction to quantum mechanics. Addison-Wesley (2004)
9. Osher, S.J., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics **79**(1) (1988) 12–49
10. Zhao, H.K.: A fast sweeping method for eikonal equations. Mathematics of Computation **74** (2005) 603–627
11. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational geometry: Algorithms and applications. Springer (2008)
12. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation **19**(90) (1965) 297–301
13. Arnold, V.I.: Mathematical methods of classical mechanics. Springer (1989)
14. Abramowitz, M., Stegun, I.A.: Handbook of mathematical functions with formulas, graphs, and mathematical tables. Government Printing Office, USA (1964)
15. Bracewell, R.N.: The Fourier transform and its applications. McGraw-Hill Science and Engineering (1999)
16. Hida, Y., Li, H.S., Bailey, D.H.: Quad-double arithmetic: Algorithms, implementation, and application. Technical Report LBNL-46996, Lawrence Berkeley National Laboratory, Berkeley, CA 94720 (2000)
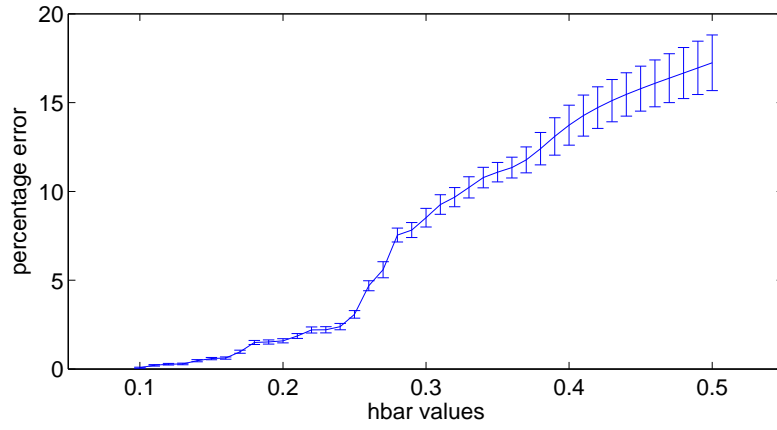
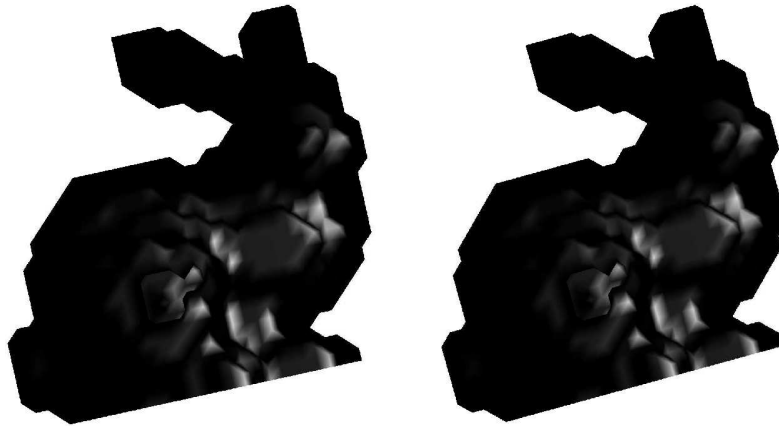**Fig. 1.** Percentage error versus $\hbar$ in 50,000 $2D$ experiments.

Fig. 2. Isosurfaces: (i) Left: Actual Euclidean distance function and (ii) Right: Our approach.