1. **Definition:**

Let be the synthesised model, be a point in, be the value at the point in. Let be the training image, be a point in, be the value at the point in.

denotes a relative neighbourhood set. e.g. when the neighbourhood size is .

Hence the neighbourhood of in can be written as the set . is defined as the neighbourhood of point in.

We then define the norm distance between by:

For each , we can find a neighbourhood in that is closest to , so that

Here denotes the minimum energy of (i.e. the closest distance between ), denotes the point corresponding to the neighbourhood closest to . The exponent causes the optimization to be more robust against outliers [Kwatra et al. 2005].

C:\Users\Tianshen Huang\Documents\PRM\PAR\PAR ANN\x64\Release\Berea200_1_D4s025_xz.png C:\Users\Tianshen Huang\Documents\PRM\PAR\PAR ANN\x64\Release\Berea200_1_D4s025_xz.png

Training image Model

Hence the total energy that we seek to minimize is defined as

The total energy is minimized in an iterative way, alternating between two steps:

In *search step*, we search the nearest neighbourhood for every point .

In *optimize step*, we update the value for each point , based on the nearest neighbourhoods for all the neighbouring points .

The overall process of optimization algorithm is illustrated as below.



1. **Iterative refinement**

In Every iteration:

1. **Search step**

For each point in , we find its nearest neighbourhood in:

Hence every will be matched with a point in, the value of that point , and the corresponding minimum energy.

The searching process is a standard nearest neighbour search in high-dimensional space. We accelerate this step mainly in two ways. First, we reduce the search dimension by PCA projection to both the neighbourhood in . We keep only the number of coefficients that preserve 95% of the variance.

Secondly, we use approximate nearest neighbour techniques (ANN) to speed up the search. ANN searches for approximate nearest neighbour that lies no farther than times the distance to the true nearest neighbour. for a good compromise between speed and accuracy [Kopf et al. 2007].

1. **Optimize step**

Generally in this step, an weighted average is yielded based on recommendations for all the neighbouring points of , such that the total energy is minimized:

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Here . To solve this we begin with a basic form: when the minimization is Least Squares problem.

* 1. Basic form: Least squares

When , the total energy is:

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

* 1. Robust optimization: Iteratively Reweighted Least Squares

Denote by , and assume the weight is constant during the *optimize step*:

Since and are constant, are constant with respect to , then the minimization of only depends on . Hence the derivative of with respect to is:

Setting the derivative to 0 yields the solution:

Assume the weight is constant during each iteration, and setting the derivative of the above function with respect to to 0 yields the following:

After one iteration, each voxel is assigned a new weighted average value, so the whole model is updated. Then a new iteration is based on updated model.

* 1. Position histogram matching in optimize step

In order to preserve the global statistics (by default: to uniformly select patterns from training image), the weight is modified.

Where is the frequency that (corresponding to ) is selected as nearest neighbourhood in model .

is the frequency that (corresponding to ) is **supposed to be** selected as nearest neighbourhood in the model . By default, each is supposed to be uniformed used.

Assume. Hence:

* 1. Discrete solver

The modified weighted average is:

In order to keep track of the position used, **the actual updated**  should come from the candidate set .

So we choose the nearest value to from , and get the updated :

* 1. Position histogram matching in search step