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 . Analogously, 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

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.

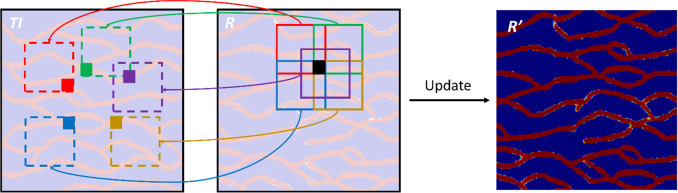
Secondly, we use approximate nearest neighbour techniques (ANN) to speed up the search. ANN searches for approximate nearest neighbour that lies no farther than times the distance to the true nearest neighbour. for a good compromise between speed and accuracy [Kopf et al. 2007].

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) |

For each , the sum of square difference of its neighbourhood is calculated – each participates in the total sum for a number of times (equals to the size of ), one for each position in neighbourhood it belongs to. Hence (2) can be rearranged as:

|  |  |  |
| --- | --- | --- |
|  |  |  |

The minimum of this sum is arrived when the quadratic function reaches its minimum for each . Since and are constant, and is constant w.r.t. , this is done by setting the derivative w.r.t. to zero:

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

So each point is assigned an average of the corresponding values in .

* 1. Robust optimization: iteratively re-weighted least squares

The least squares estimation (3) of w.r.t. will cause the outliers - that are not very close to to have an undue influence on . This can be improved by using a robust energy function with the exponent, is used here for optimum performance [Kwatra et al. 2005]. The total energy (1) is now:

|  |  |
| --- | --- |
|  |  |

Denote by :

|  |  |  |
| --- | --- | --- |
|  |  |  |

Assume the weight is constant during the *optimize step*. This type of energy function is typically solved using iteratively re-weighted least squares (IRLS):

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

The outcome is a weighted average of (3).

Additionally, we apply a 2D Gaussian distribution for each point within a neighbourhood, so that the point closer to the centre of the neighbourhood has a greater influence than those far away.

* 1. Position histogram matching in optimize step

For many training images, the above process could converge to a wrong local minimum, because the energy function only measures the local neighbourhood similarity, without accounting for any global statistics. Reweighting schemes that adjust the weight dynamically during each iteration are developed [Kopf. et al 2007, Chen et al. 2010], to drive the solution to certain desired global statistics.

Here we implemented a reweighting scheme based on position histogram [Chen et al. 2010]. It aims to ensure that most of pixels in training image can be found in the model, and they are encouraged to be equiprobably utilized (for default uniform training image).

Let be the frequency that is copied and used in . The size of position histogram is the size of training image , and denotes the desired frequency of histogram (by default ).

Chen [2010] modified the weights in (4) as:

|  |  |
| --- | --- |
|  |  |

We modify the weight differently, using a Gaussian distribution function:

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Here , the mean of Gaussian distribution and standard deviation . This helps to drive a more uniform utilization of training image.

* 1. Discrete solver

The modified weighted average can be written as:

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 :

* 1. Position histogram matching in search step