# Fast BlockMatching Registration with Entropy-based Similarity

*Release 0.00*

Eduardo Suárez-Santana[1], Rafael Nebot[1], Carl-Fredrik Westin[3] and Juan Ruiz-Alzola[1,2]

June 25, 2007

[1]Canary Islands Institute of Technology
[2]Department of Signals & Communications, University of Las Palmas de Gran Canaria
[3]Laboratory of Mathematics in Imaging, Brigham and Women's Hospital and Harvard Medical School

**Abstract**

This paper describes the implementation of a multidimensional block-matching nonrigid registration algorithm. The main features of the algorithm are its simplicity, its free form nature, the modularity of the similarity measure, which makes it possible using local entropy-based similarity measures and the avoidance of the optimization module. The algorithm implementation described in this paper is based on the method by Suarez et al. [5, 3]. This paper, which has already been submitted to the Insight Journal, is accompanied with the source code, input data, parameters and output data used for validating the algorithm described in it.

## Contents

Nonrigid registration using entropy-based similarity measures is most often accomplished in three ways: (1) using global optimization, (2) using a differential similarity measure and (3) using a block-matching scheme. Examples and implementations of (1) and (2) are found in the Insight Toolkit [1]. However, block-matching schemes are rarely found in literature to perform this task.

Global optimization facilitates the implementation of a wide range of registration models. However, it makes finer meshes computationally more expensive, it requires the right choice of parameters and optimization algorithm, and it becomes difficult to parallelize.

Differential similarity measures is another good approach. It can be fast and accurate. Nevertheless, the numerical approach and its implementation is usually complex for similarity measures based on entropy, the number of iterations may be large because of the propagation of the displacement information, it needs the computation of derivatives which may need smoothing steps for medical images, and boundary conditions may lead to non-free-form implementations.

Block-Matching is probably less elegant from a theoretical point of view, but it shows several nice features:

- It is simple to formulate
- It is simple to implement
- It is difficult to get a numerical instability
- It allows parallelization
- It is robust against noise
- Few parameters are in general required
- It is suitable for hardware implementations

Perhaps these are some of the reasons why the MPEG standard for video uses a block-matching scheme as well. However, some tricks are needed in order to make it fast and to compute local estimations of entropy-based similarity measures.

The algorithm implemented includes several contributions already published by Suarez et al. [5, 3]. A review of this algorithm is given in section 1, and section 2 describes the details of the implementation. Future work follows after this section in order to introduce the improvements of next releases. Section 4 presents the experiment and its result attached to the code.

# 1 The Algorithm

The registration algorithm consists of a pyramidal block-matching scheme [5]. A block is defined as a neighborhood of voxels around a selected one. For the explanation of the algorithm, we will follow figure 1, which is the representation of one level of the pyramid.

First, *Image B* is sampled in the positions of the samples of *Image A* to get *Image B Transformed*. Then, in the *Data Matching* step, for every voxel in *Image B Transformed*, a block is taken which then is matched against a set of test blocks in *Image A*. The displacement that achieves the best similarity between blocks, is stored as the displacement for the original voxel. With the multi-resolution scheme used, it is only necessary to test displacements of one voxel. It can be shown that given a point-wise similarity measure, it is equivalent to switch between slow block matching per voxel and a faster convolution for every displacement.

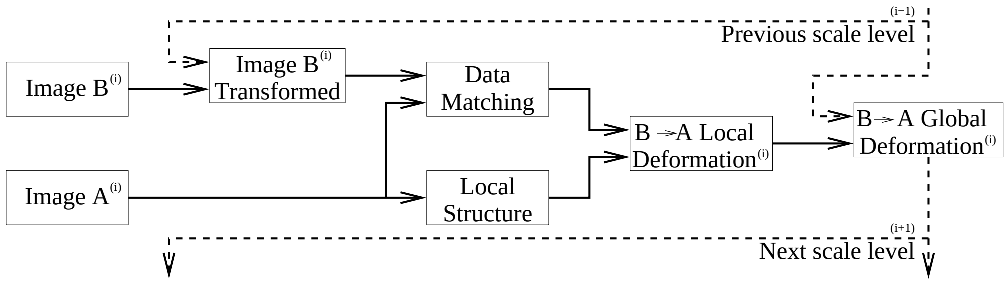



Figure 1: Algorithm pipeline for pyramidal level (i).

Next, the discrete vector field is regularized weighted by *Local Structure* in *Image A*, in order to speed up the propagation of the displacement information (*B→A Local Deformation*). Finally, composition of local deformation field for the working level with the global deformation field for the previous level gives the global deformation field for this level (*B→A Global Deformation*).

A point-wise similarity measure based on entropy is calculated by estimating the joint probability density function (pdf) of the whole images $\mathsf{p}(i_A, i_B)$. Then, for every pair of matching voxels $(A(x^1,x^2,x^3), B(y^1,y^2,y^3)))$, we can measure its point-wise joint density as $\mathsf{p}(i_A(x^1,x^2,x^3), i_B(h(x^1,x^2,x^3)))$. In this way, the mutual information on a block of $N$ matched samples $(i_A[k], i_B[k])$ would be:

$$MI_{block}(I_A, I_B) \backsimeq \sum_{k=1}^{N} \log \frac{\mathsf{p}(i_1[k], i_2[k])}{\mathsf{p}(i_1[k])\mathsf{p}(i_2[k])} \quad . \tag{1}$$

This is similar to divide the number of samples in two groups: the former used to estimate the pdf and the latter to evaluate it. In this case, the former would be the whole matched image and the latter the samples in the block. A full explanation of this technique can be found in [4].

## 2 The Implementation

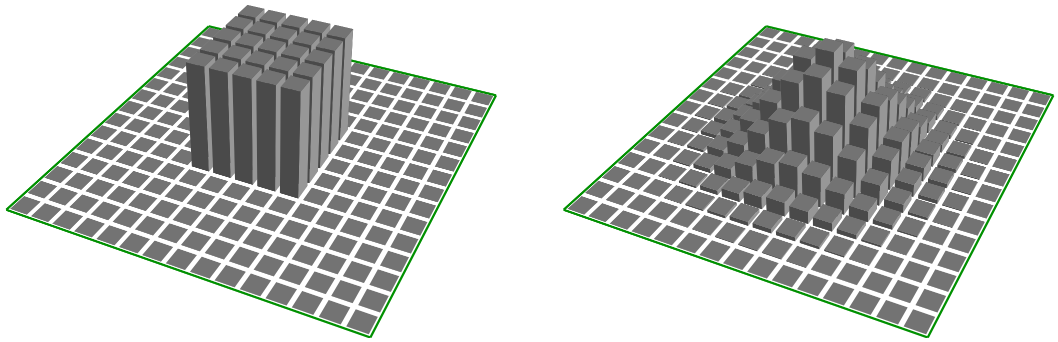
The algorithm has been implemented using the Insight Toolkit [1]. Hence, you need to have the following software installed:

- Insight Toolkit 2.8
- CMake 2.4

To help the readers of the source code, some notation will be introduced. We are registering images $A$ attached to the reference system $X$, and $B$ attached to the reference system $Y$. $X$ is the reference system where the matching is done.

### 2.1 Source code

Every iteration (shown in figure 1) is performed by the method `UpdateDeformation`. As input we receive the deformation field `initialDeformationInX`, and the images `ImageAInX` and `ImageBInY`. There, the method `LinearResampleIntoX` moves $B$ to the reference system $X$ (*Image B Transformed* in the figure).



(a) Square kernel ($w$=2) (b) Gaussian kernel ($\sigma_{sim}$=2.2)

Figure 2: Kernels for convolution with point-wise similarity images.

*Data Matching* takes place inside `Update Deformation` as well. There, for every unitary test displacement, is carried out point-wise similarity. We have implemented Mutual Information in a method called `SimilarityMI`. Block Matching, or its equivalent implementation as a convolution with a square window, has been carried out with Gaussian smoothing, because it is faster and has less aliasing (see figure 2). *Local Structure* is implemented in the method `Structure`. Weighting with local structure is done with the method `SmoothWithCertainty`. The typical deviation ($\sigma$) of these two Gaussian smoothing steps, similarity and regularization, are the only parameters needed for the algorithm. Default values are used in the constructor to facilitate the use of the class.

Composition of local with initial deformation to get the global deformation is done by sampling the initial deformation field. To resample this field it is necessary to transform the displacement vectors into mesh positions:

```
vect = initialDeformationInXIterator.Get();
initialDeformationInX->TransformIndexToPhysicalPoint(
  initialDeformationInXIterator.GetIndex(),
  position );

for ( int coordinate = 0; coordinate < ImageDimension; ++coordinate )
  vect[ coordinate ] += position[ coordinate ];

initialDeformationInXIterator.Set( vect );
```

Then, method `SmoothWithCertainty` is used before transforming the mesh back to displacement vectors.

## 2.2 Use of Not-A-Number

This algorithm makes intensive use of the signal/certainty philosophy and of the *Normalized Convolution* algorithm [2]. Representation of the unknown voxel values has been done using the value NaN. Some modifications to `itkWarpImageFilter` have been necessary to reuse this class. Hence, using NaNs to pad images will make consistent the representation of NaN values for samples where no information is available. This trick allows saving memory and the free-form behaviour of the deformation field. To use the normalized convolution algorithm, it is necessary to decouple signal and certainty from these images. Hence, the `StripNanImage` has been implemented.

# 3 Future Work

This is a first release of the algorithm. Some optimizations can still be done, both in the algorithm and in the implementation.

## 3.1 The implementation

We have done our best to implement the algorithm using the Itk Coding Style. However, Itk kernel developers will need to have a look at it to integrate it with the rest of Itk. Next there are some items to take into account:

- **Use of an initial transformation**. The algorithm is almost prepared for that, but this feature is not available at the moment.
- **Efficient use of symmetric matrices**. At the moment, there is not such implementation in Itk.
- **Efficient use of image adaptors**. The similarity measure, as an example, could be coded using them.
- **Independence of the normalized convolution**. It has been coded inside the class. It should be coded as a separate one.
- **Independence of the similarity measure**. It has been coded inside the class as well and they should be coded as separate classes.
- **Adaptation for interaction**. The algorithm needs to be adapted to be suitable for user interaction. In that way, a user would set some control points that would modify signal/certainty in the regularization step. This would lead the registration process, following the indications of a physician.

## 3.2 The algorithm

Despite the algorithm may be modified to be more efficient, there are some direct optimizations to be made in following releases:

- **Use of another pyramidal scheme**. The algorithm makes use of the `MultiResolutionPyramidImageFilter` algorithm already implemented in Itk. In this algorithm, the resolution of each level is reduced an integer factor from the original data. Having a float factor would allow more levels and it would increase the quality of the registration.
- **Use of a reduced neighborhood**. Neighborhood information is needed in differential registration algorithms to compute gradients. This algorithm makes use of a neighborhood of radius one. In 3-D that means to test 27 possible displacements to get the right one. This could be reduced to 7 test displacements (two in each direction plus the center), and get the final displacement as a composition, in contrast to gradient estimation, that would need at least test 4 positions (one in each direction plus the center).

## 4 Results

Because of the size of medical data, only a 2-D example has been uploaded to the Insight Journal. The original image (`data_original_eduardo256.jpg`) is a picture of the first author.

The experiment consists of deforming the original image with a synthetic deformation field (`data_input1_odraude256.jpg`) and then registering them back with the registration algorithm. To test the Mutual Information similarity measure, the intensity of the original image has been modified by a sin function and some noise has been added (`data_input0_fevbsep256.jpg`). So that, `data_input0_fevbsep256.jpg` and `data_input1_odraude256.jpg` are the images to be registered. Images are shown in figure 3.

The registration process took, on a 3.4 GHz intel Pentium running linux, about eight seconds. With the output deformation field, the moving image (`data_input1_odraude256.jpg`) has been moved back with the deformation field, resulting in the registered image `data_outputAuthor_odraude256def.mha`. Simple visible inspection is the guarantee of the success of the algorithm.

## 5 Conclusions

This paper has presented the implementation of a block matching algorithm for registration of medical images. Since there is a lot of improvements and future work, we are on an early stage to compare it with other registration algorithms. Our experience is that block matching can be fast, and thus may have an advantage in certain applications. Furthermore, the use of a certainty function would make it possible to take care of the user interactions to lead the registration process by setting points of anatomic relevance or homologous points.

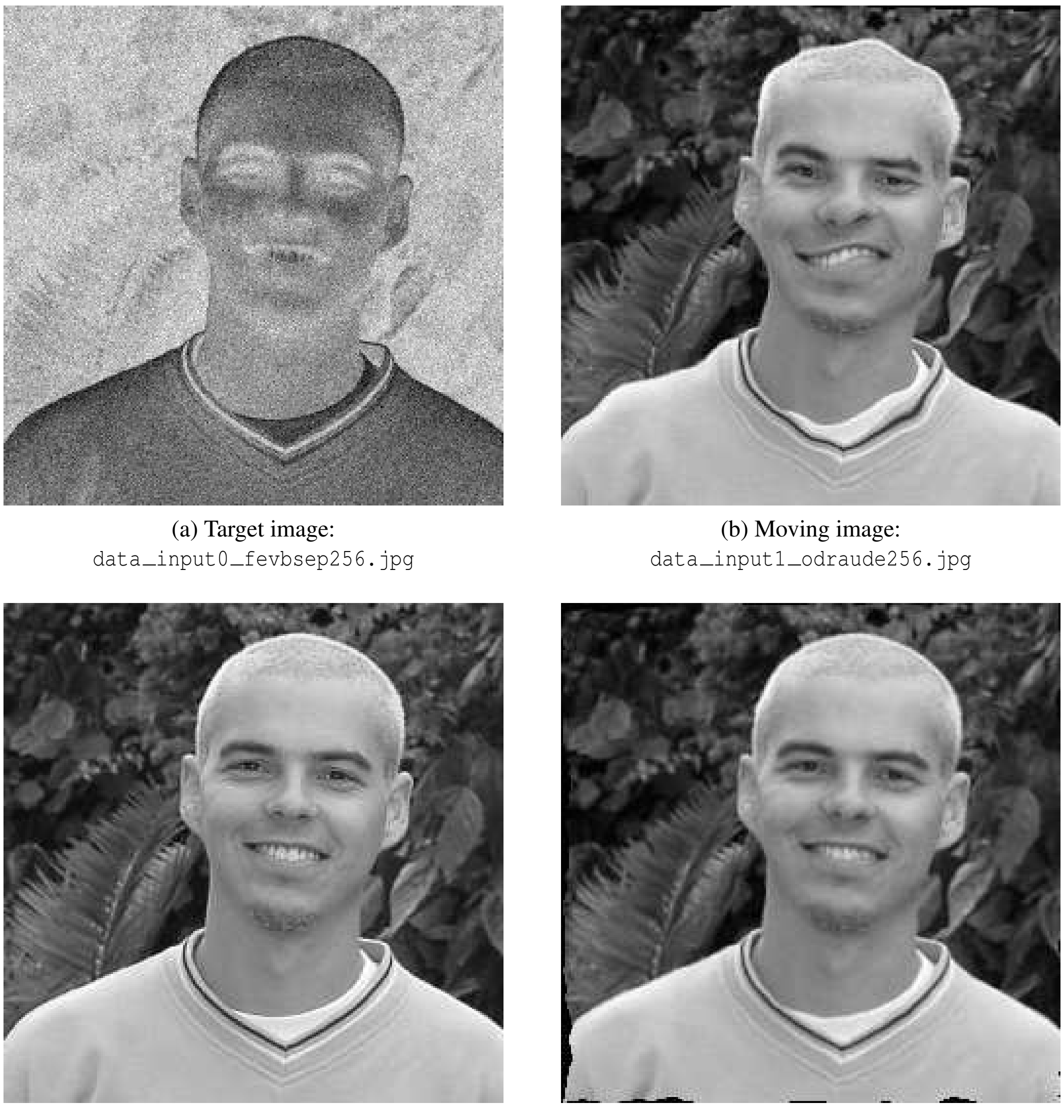
## 6 Acknowledgements

This work has been partially supported by the grant from the Spanish Ministry of Science and Technology PTQ2004-1444. This work is part as well of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005149, and Neuro Analysis Center (NAC) NIH P41-RR13218. Information on the National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih.gov/bioinformatics. Special thanks to Dan Blezek and Luis Ibañez for their hints to this implementation.

## References

[1] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-10-6, http://www.itk.org/ItkSoftwareGuide.pdf, first edition, 2003. (document), 2

[2] H. Knutsson and C-F. Westin. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In *Proceedings of Computer Vision and Pattern Recognition*, pages 515–523, New York City, USA, June 1993. IEEE. 2.2



(a) Target image:
data_input0_fevbsep256.jpg

(b) Moving image:
data_input1_odraude256.jpg

(c) Original image:
data_original_eduardo256.jpg

(d) Registered image:
data_outputAuthor_odraude256def.mha

Figure 3: Images in the experiment.

[3] Eduardo Suárez. *Un marco general para la normalización geométrica de datos tensoriales y su aplicación al registrado de imágenes médicas*. PhD thesis, University of Las Palmas de Gran Canaria, jul 2003. (document)

[4] Eduardo Suárez, Jose A. Santana, Eduardo Rovaris, Carl-Fredrik Westin, and Juan Ruiz-Alzola. A fast mutual information estimation for nonrigid registration. In *Proceedings of the 9th Internation Workshop on Computer Aided Systems Theory (EUROCAST '03)*, Cast and Tools for Complexity in Biological, Physical and Engineering Systems, pages 89–93, Las Palmas de Gran Canaria, Spain, February 2003. 1

[5] Eduardo Suárez, Carl-Fredrik Westin, Eduardo Rovaris, and Juan Ruiz-Alzola. Nonrigid registration using regularized matching weighted by local structure. In *Proceedings of the Fifth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '02)*, number 2 in Lecture Notes in Computer Science, pages 581–589, Tokyo, Japan, September 2002. (document), 1