bUnwarpJ: Consistent and Elastic Registration in ImageJ. Methods and Applications.

Ignacio Arganda-Carreras[a], Carlos O. S. Sorzano[b], Jan Kybic[c] and Carlos Ortiz-de-Solorzano[d]

[a]Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain;
[b]Bioengineering Laboratory of Univ. San Pablo CEU, Madrid, Spain;
[c]Center for Machine Perception, Czech Technical University, Prague, Czech Republic;
[d]Cancer Imaging Laboratory, Centre for Applied Medical Research (CIMA), Pamplona, Spain

Abstract

Image registration is a common problem in computer vision, and specifically in biomedical applications that require inter or intra modality image alignment. We have developed an ImageJ plugin called bUnwarpJ, which elastically registers pairs of images. This simple and easy-to-use plugin can be used by researchers and clinicians to create anatomical atlases, segment images using atlases, align pairs of images distorted by both physical and acquisition related distortions, etc.

Registering two images consists on finding the image transformation that maps corresponding pairs of pixels between the original and "distorted" images. We use here the term distorted in a wide sense, to account not only for "sensu strictu" image distorsions, but also for anatomical variations between or within individuals. We use an algorithm that simultaneously calculates the direct and inverse transformations and minimizes the similarity error between the target and source images after imposing a consistency constraint. This approach provides bidirectional registration –from image A to B or from B to A– in a single computation. We use B-splines to represent both images and deformations and make use of a powerful optimizer to converge fast to the best image alignment.

Our plugin allows guiding the registration process using the image similarity, the consistency of the deformations, vector-spline regularization and/or a set of optional landmarks, which can be calculated and fed from other ImageJ plugins such as the automatic extractors Scale-Invariant Feature Transform (SIFT) and Multi-Scale Oriented Patches (MOPs). The user can give a weight to each of these terms in the registration process. This paper provides a general description of the algorithm and its implementation in order to help developers and users to exploit all its potential.

1. INTRODUCTION

The registration of a source image S onto a target image T consists of finding the image transformation that best maps one image into the other. Given that the direct transformation is often not invertible, we need to calculate the inverse transformation of T into S separately. Inspired in the work of Sorzano *et al.*,[1] we developed the ImageJ[2] plugin bUnwarpJ,[3] which calculates both transformations simultaneously while imposing a geometric consistency constraint on them. The consistency constraint helps the optimizer to avoid getting trapped in local minima and calculates the direct and inverse transformations in one computation.

bUnwarpJ was originally designed to register histological sections,[3] although it has been used to register other types of images, for instance electrophoretic 2-D gels.[4] The first release of bUnwarpJ dates from July 2006. Since then, several upgrades and new tools have been added. We describe the main features of the plugin in the following sections.

2. MATERIALS AND METHODS

The deformation function calculated by the registration method should be bijective. In other words, it should unequivocally link every pixel in the target image T with a pixel in the source image S. It should also have biological meaning appropriate for the particular image modality and source of misalignment. Some authors propose using diffeomorphic deformation functions, which are invertible, differentiable and bijective.[5–7] This means that if the transformation were applied to a real physical object -for instance a tissue section- then no folding or tearing of the object would be allowed. Therefore, enforcing diffeomorphism is costly and might be overly restricted to register some types of biological images. To solve this problem, Christensen *et al.*[8] computed two independent deformations that combined should be as close as possible to the identity transformation. This closeness to identity is explicitly introduced into the objective function. This two-way constrained registration is known as consistent registration.

The standard registration method presented by Sorzano *et al.*[1] proposes the calculation of the elastic deformation field trough the minimization of an energy functional composed by three terms: the energy of the similarity error between both images (represented by the quadratic pixel error), the error of the mapping of soft landmarks, and a regularization term based on the divergence and the curl of the deformation to ensure its smoothness. They use a Levenberg-Marquardt minimization enhanced by a Broyden-Fletcher-Goldfarb-Shanno (BFGS) estimate of the local Hessian of the goal function. We extended the method[3] adding to the energy functional a factor of the consistency of the deformation field. This way, we calculate both the direct and inverse transformations at the same time. Therefore, the new energy functional includes the dissimilarity between the source and target images -now in both directions- E_{img}, an optional landmark constraint E_{μ}, the regularization term $(E_{div} + E_{rot})$, and an energy term E_{cons} that accounts for the geometrical consistency between the elastic deformation in both directions. Namely, the energy function is now given by

$$E = w_i E_{img} + w_{\mu} E_{\mu} + (w_d E_{div} + w_r E_{rot}) + w_c E_{cons}. \tag{1}$$

Where w_x are the specific weights given to the different energy terms. These weights can be set by the bUnwarpJ user in the *Advanced Options* window of the plugin.

2.1 Image and Deformation Representation

We chose to use B-splines to interpolate the images and model the deformation functions. B-splines are computationally efficient, differentiable, have good approximation properties and can be used to represent both linear and non-linear transformations, providing close control of the level of detail of the transformation. Moreover, we use a multiresolution (iterative coarse-to-fine) implementation, which improves the convergence speed and robustness of the algorithm.[9] The resolution detail can be directly controlled by the user with the *Initial/Final Deformation* parameters in the *Advanced Options* window. The user can choose the level of detail, i.e. the number of B-splines used to represent the deformation fields at each resolution level. The first version of the plugin provided four levels that varied from "Very Coarse" (1 - lowest level) to "Very Fine" (4 - highest level). For a resolution level n, we create a deformation grid of $2^{n-1} \times 2^{n-1}$ intervals. That means that in that first version we created deformation grids of 1×1 intervals at the lowest resolution level and of 8×8 intervals at the highest level. To create i intervals, we need $i + 1$ points, i.e. B-spline coefficients. Finally, we add 2 extra coefficients -one at the beginning and one at the end of every row and column of the grid- in order to avoid problems at the boundaries. Therefore, for a resolution level n we represent the deformation field using $(2^{n-1} + 3) \times (2^{n-1} + 3)$ B-spline coefficients.

2.2 Search for the optimum

Our algorithm starts searching for the minimum of the energy functional at the lowest level of both the image and deformation pyramids. Once the optimum registration at that level is found, it moves up to the next level in one of the pyramids. The system first increases the deformation level and then alternates image and deformation steps until the maximum resolution of both pyramids is reached. Once the algorithm reaches the top of one of the pyramids, it moves up only in the other pyramid until reaching its top. The way of combining the image and deformation resolution pyramids when using 4 deformation levels is visually described in Figure 1. In our

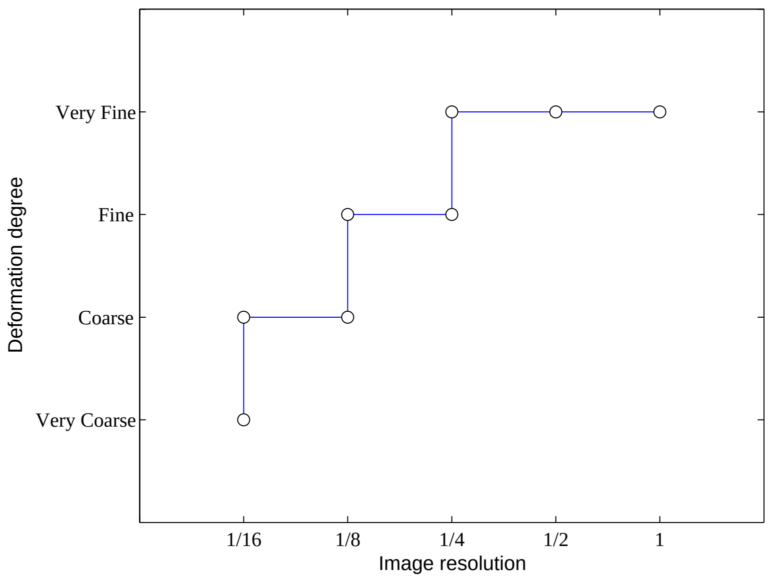

Figure 1. Resolution steps for 4 levels of deformation pyramid, i.e. the deformations vary from "Very Coarse" to "Very Fine" and the image resolution from 1/16 to 1.

implementation, the image pyramid is one level larger than the deformation pyramid, i.e. 5 image resolution levels vs 4 levels of the deformation field pyramid. These numbers and the way the pyramids are combined can be easily changed. The latest plugin release allows for 5 deformation levels as well.

2.3 New plugin features

Since its first release, bUnwarpJ has been several times updated and new features have been added to:

- Load an initial transformation. By default, our algorithm starts searching for the optimum image mapping from the identity deformation. However, many users who had already registered their images with affine transformations wanted to use bUnwarpJ to refine their results or to solve the non-rigid distortions between their images not corrected by the affine transformation. The last release of bUnwarpJ lets the user load an affine transformation to replace the identity as a starting point of the elastic registration.
- Compare transformations. With our plugin, the user can now convert the resulting registration transformations into a raw format, i.e. to a pixel-to-pixel matching matrix- This way the results can be compared (see Section 3.1) with other registration methods, for instance using *warping index*.[10]
- Compose transformations. When registering a sequence of n images we are sometimes interested in composing the $n-1$ pair-wise transformations. This way the images are deformed in one step, avoiding the degradation effect of interpolation involved in applying each transformation. That can now be done in the *Input/Output Menu*.
- Automatic landmarks integration. The *PointRoi*'s of the input images are now directly transformed into landmarks. This way, we can now use landmark pairs provided, for instance, by automatic landmark extractors.

3. RESULTS

3.1 Validation

We have developed another ImageJ plugin to test the performance of bUnwarpJ and calculate the accuracy of the method. This plugin, called SplineDeformationGenerator, can apply 5 different deformations on an image:

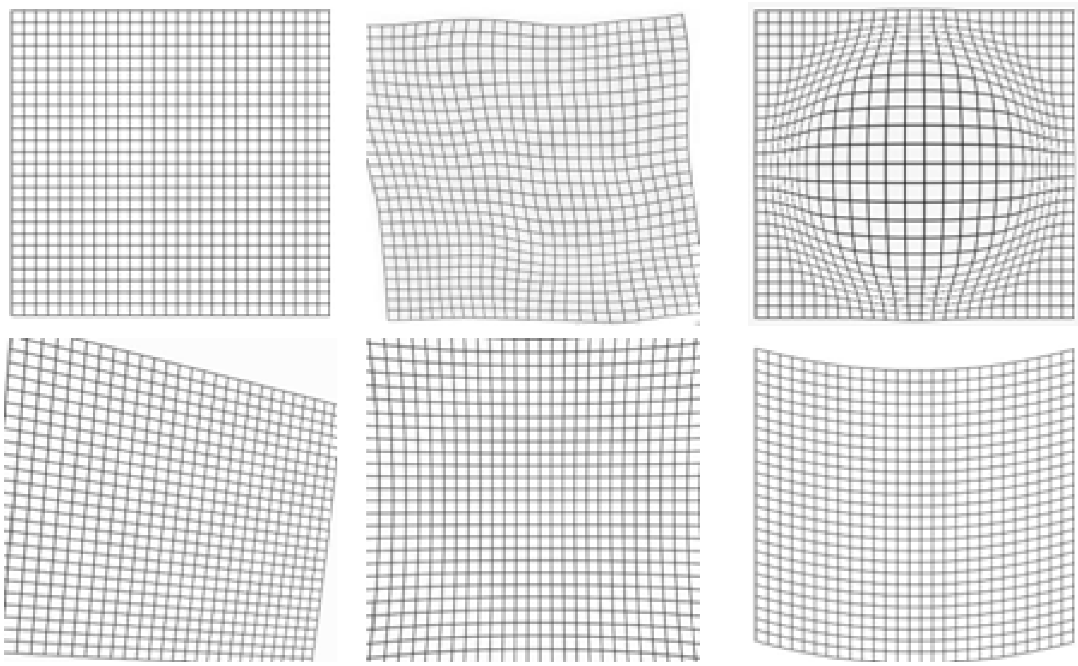

Figure 2. Deformations available with SplineDeformationGenerator. From left to right and from top to down: the original grid image and its corresponding elastic, fisheye, perspective, barrel and smile effect examples of deformations.

elastic, fisheye, perspective, barrel/pincushion and "simile" deformations (see Figure 2). This way, we can produce synthetic data sets to test the algorithm. We measured the accuracy of the registrations using a mean squared distance version of the standard *warping index*[10] defined as

$$\varpi = \sqrt{\frac{1}{\parallel R \parallel} \sum_{\mathbf{x} \in R} \|\mathbf{x} - g(g^*(\mathbf{x}))\|^2}. \tag{2}$$

Where g^* is the synthetic known deformation, g is the deformation in the opposite direction and R is the set of pixels common to both images. The warping index measures the average geometric error -in pixels- between the original transformation and the deformation calculated by our algorithm. Thanks to this plugin and the new deformation comparison features of bUnwarpJ (see 2.3), it is easy to produce random deformations over any image and check the method performance. SplineDeformationGenerator can be freely downloaded from `http://biocomp.cnb.csic.es/~iarganda/SplineDeformationGenerator/`.

3.2 New applications

The most powerful of the new features of bUnwarpJ is that it permits to combine its performance with the automatic landmark extractors SIFT[11] and MOPs,[12] implemented in ImageJ by Stephan Saalfeld.[13] The output of those methods can be directly fed as landmarks in bUnwarpJ. Then, the landmarks can either be directly included in the energy functional by setting a specific landmark weight or used to calculate an affine transformation which serves as a starting point for the elastic registration. This is done by setting w_μ to 0. Using landmarks we can correct for strong rigid distortions in the images before invoking bUnwarpJ. Figure 3 shows an example of applying the MOPS plugin and bUnwarpJ to two monkey brain sections from different individuals. One of the sections was heavily rotated with respect to the other. We first applied the MOPs plugin with the maximal alignment error set to 10.0 pixels. The image upscaling and the rest of the parameters were set to their default values. Then, we just launched bUnwarpJ with default parameters and a deformation pyramid of 4 levels. The orientation was rapidly corrected thanks to the landmarks. Then the optimization continued until convergence.

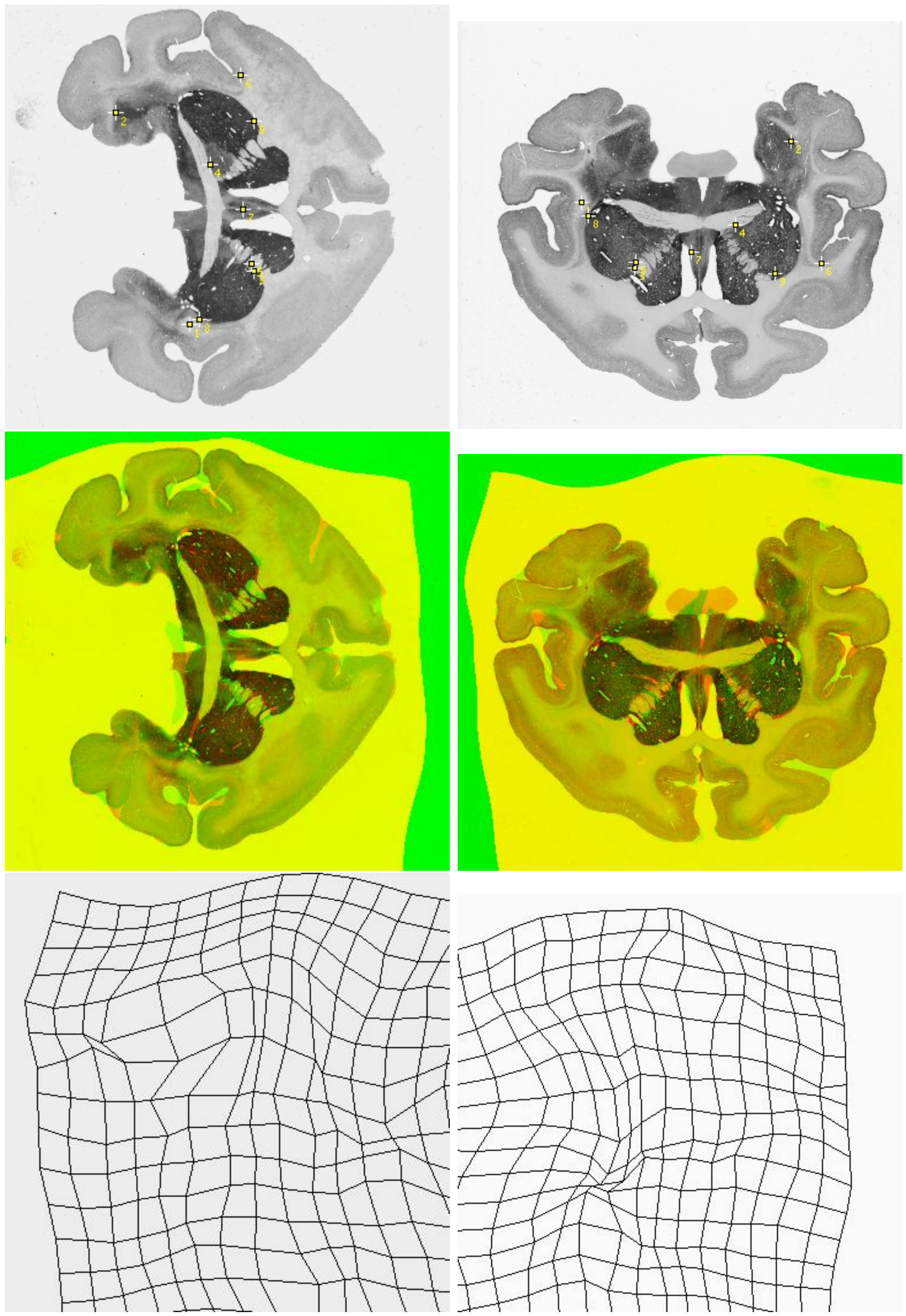

Figure 3. Example of MOPS and bUnwarpJ plugins performance. From top to bottom and from left to right: original source and target images with automatically detected landmarks, RGB registration results (yellow color meaning perfect superposition and red and green colors pointing out the misalignment regions) and final deformation grids.

4. CONCLUSIONS AND FUTURE WORK

We have described our tool for consistent and elastic image registration, bUnwarpJ. The most up-to-date release of this ImageJ plugin and its source code can be freely downloaded from `http://biocomp.cnb.csic.es/~iarganda/bUnwarpJ/`. The plugin has been updated to be able to interact with automatic landmark extractors, allow affine transformation initialization and compare its performance with any other registration method.

We plan to adapt the code to simplify its interaction with the ImageJ macro language and create a more detailed step-by-step user manual.

5. ACKNOWLEDGMENTS

Jan Kybic was supported by the Czech Ministry of Education under Project 1M0567.

REFERENCES

[1] Sorzano, C. O. S., Thévenaz, P., and Unser, M., "Elastic registration of biological images using vector-spline regularization," *IEEE Transactions on Biomedical Engineering* **52**, 652–663 (April 2005).

[2] Rasband, W., *ImageJ*. U. S. National Institutes of Health, Bethesda, Maryland, USA (1997-2007).

[3] Arganda-Carreras, I., Sorzano, C. O. S., Marabini, R., Carazo, J.-M., Ortiz-de Solorzano, C., and Kybic, J., "Consistent and elastic registration of histological sections using vector-spline regularization," in [*Computer Vision Approaches to Medical Image Analysis*], *Lecture Notes in Computer Science* **4241**, 85–95, Springer Berlin/Heidelberg (May 2006).

[4] Sorzano, C. O. S., Arganda-Carreras, I., Thévenaz, P., Beloso, A., Morales, G., Valdes, I., Perez-Garcia, C., Castillo, C., Garrido, E., and Unser, M., "Elastic image registration of 2d gels for differential and repeatability studies," *Proteomics* **8**, 62–65 (2008).

[5] Avants, B. B., Schoenemann, P. T., and Gee, J. C., "Lagrangian frame diffeomorphic image registration: Morphometric comparison of human and chimpanzee cortex," *Medical Image Analysis* **10**, 397–412 (June 2006).

[6] Cootes, T. F., Marsland, S., Twining, C. J., Smith, K., and Taylor, C. J., "Groupwise diffeomorphic non-rigid registration for automatic model building," in [*ECCV (4)*], Pajdla, T. and Matas, J., eds., *Lecture Notes in Computer Science* **3024**, 316–327, Springer (2004).

[7] Rueckert, D., Aljabar, P., Heckemann, R. A., Hajnal, J. V., and Hammers, A., "Diffeomorphic registration using b-splines," in [*MICCAI (2)*], Larsen, R., Nielsen, M., and Sporring, J., eds., *Lecture Notes in Computer Science* **4191**, 702–709, Springer (2006).

[8] Christensen, G. E. and He, J., "Consistent nonlinear elastic image registration," in [*Mathematical Methods in Biomedical Image Analysis*], 37–43, IEEE (2001).

[9] Unser, M., "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine* **16**, 22–38 (November 1999). IEEE Signal Processing Society's 2000 magazine award.

[10] Thévenaz, P., Ruttimann, U. E., and Unser, M., "A pyramid approach to subpixel registration based on intensity.," *IEEE Transactions on Image Processing* **7**(1), 27–41 (1998).

[11] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[12] Brown, M., Szeliski, R., and Winder, S. A. J., "Multi-image matching using multi-scale oriented patches," in [*CVPR (1)*], 510–517, IEEE Computer Society (2005).

[13] `http://fly.mpi-cbg.de/~saalfeld/`.