# Hybrid Pipelining Approach to Image Alignment for Large-Scale Brain Image Data

Rustam Sengupta, Joerg Meyer, Zhihe Zhang

University of California, Irvine, 644E Engineering Tower, Irvine, CA 92697-2625
{rsengupt | jmeyer | zhihez}@uci.edu

## Abstract

Creating thin sections of frozen tissue in the order of a few microns and then manually mounting the resulting slices on glass plates is a common technique in brain imaging. Large collections of such manually mounted cryosections are available for digital archiving. Scanning these slices at high resolution helps preserving them for future generations. The obtained data can also be used to restore the original shape of the specimen, i.e., to create a three-dimensional model. This task usually requires time-consuming individual alignment of the slices. We present a framework that uses a pipelining approach to aid in the alignment process of a large data set, and to automate most of the steps.

### Keywords

Image Registration, Large-Scale Data Visualization, Histograms, Thresholding, Feature Extraction

## 1. Introduction

Creating a three-dimensional model from a set of manually mounted cryosections is a tedious and time-consuming task if done by hand. We have developed a framework that helps with the alignment of such raw image data material. It uses a pipelining approach that consists of multiple steps: (i) detection of image feature points that are used as registration markers, (ii) linear, affine transformation steps [2], and (iii) non-linear transformation steps.

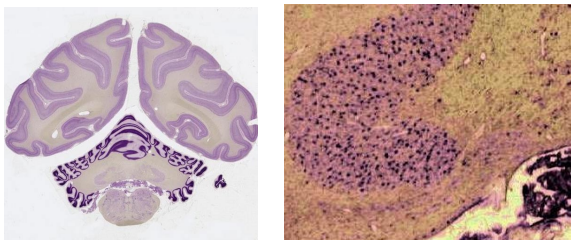

Figure 1: (a) A single cryosection of a Rhesus Macaque Monkey brain, scanned at 2666dpi; (b) detail view.

We apply this technique to a set of 1,400 slices of a Rhesus Macaque Monkey brain that were scanned at a resolution of 2,666dpi (9µm per pixel), resulting in images of approximately 5,000 x 4,000 pixels (76 GB of RGB data). The enormous resolution of the scanned cryosections (figure 1a) enables viewing of individual stained cell nuclei and other detail on the cellular level (figure 1b).

In contrast to other modalities, such as Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), where the slices are always perfectly aligned, in our case, where the cryosections were mounted manually onto the glass plates, the alignment is far from being perfect, making a three-dimensional reconstruction virtually impossible. Instead of reslicing and rescanning a new brain using a more automated slicing technique, we want to use the existing experimental material and use a software-based technique to align the slices so that a three-dimensional reconstruction is possible and a 3-D model is obtained that can be used for a 3-D brain atlas. The purpose of this framework is the creation of a well-aligned 3-D data set of a Rhesus Macaque Monkey brain.

The implications of this framework reach beyond the registration of a monkey brain. The pipeline approach presented in this paper enables registration of large sets of cryosections ranging from mouse brains to human brains and from inner organs to embryonic images. Some of the collections that we possess today are more than eighty years old and need to be preserved before they start to decay, and some cannot be recreated due to ethical constraints and legal restrictions. For these reasons, it is critical to create digital archives that preserve these valuable collections for future generations.

## 2. Brain Image Data

This article describes a framework for a pipeline for image alignment. It is based on a hybrid approach, incorporating a number of image processing and biomedical imaging analysis techniques. The final goal of this work is to facilitate rendering of a high resolution 3-D volume of a Rhesus Macaque Monkey brain.

The data set comprises of 1,400 high-resolution cross-sectional images of the brain of a Rhesus Macaque Monkey. The Center for Neuroscience at the University of California, Davis, performed the cutting and the scanning of the slices. Both tasks were commonly done manually (sometimes several years ago before the introduction of automated scanning and image processing techniques), which gave rise to severe misalignment problems. The slices were scanned at a very high resolution (2,666 dpi). The objective of doing so was to archive their valuable information electronically, and to enable the construction of a 3D volumetric brain atlas. The results will later be transferred to human brains (figure 2).

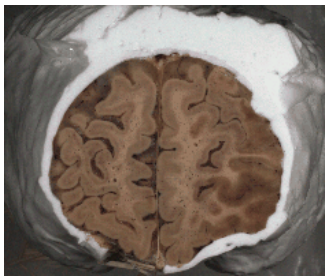

Figure 2: Cryosection of a human brain

In the monkey brain image series, the thickness of each slice is 30µm and the cell nuclei were stained in order to make them clearly visible (figure 1b).

Our algorithm uses different properties of the data set to identify points that are used as registration markers. Some of these properties are specific to the given data set, others are of a general nature. The pipeline approach enables us to add and remove different modules from the pipeline, so that it can be adapted to various data sets. This is a common phenomenon in biomedical imaging. Often times it is not possible to define a general solution for all problems in a given field, because we are using biological material which is subject to variation, and the structure of an algorithm should take this into account. Therefore, we choose a modular pipeline approach in order to minimize efforts necessary for the development of new modules and to maximize reusability of already implemented modules.

For the given application, the chosen set of modules comprises of contour detection, a symmetry metric, thresholding (identification of areas exposing a higher density of dark spots), and identification of pre-existing registration markers. The latter is a module that was developed specifically for detecting pinholes caused by two metal pins that were pushed through the brain as registration markers before slicing [1]. All the others are general purpose modules that can be easily adapted to other applications by varying thresholds or cut-off values.

## 3. Previous Work

A number of image registration and alignment techniques have been used for different modalities. Most of the methods rely on the image content. The survey conducted by Maintz and Viergever [8] highlights the classification and dimensionality of different image registration methods. According to this survey, image registration can be based on a limited set of identified salient points (*landmarks*), on the alignment of segmented binary structures (*segmentation based*), or directly on measures computed from the intensity levels of the image (*voxel property based*).

Landmarks can be anatomical, i.e., salient and accurately locatable points of the morphology of the visible anatomy, usually identified interactively by the user (Evans *et al.* [3], Zubal *et al.* [4]). Landmark-based registration is universally applicable as it can be used on any image, no matter what the specimen is. Landmarks are mostly used in combination with other registration techniques that rely on parameters like curved surfaces and volumes. The drawback of this technique is that usually user interaction is required for the identification of these landmarks.

In segmentation-based techniques (Chen *et al.* [5], Henderson *et al.* [6]) the image is broken into fragments and one image is elastically deformed to match the second one. The rigid model approach is the most popular one currently in use. The *Chamfer Matching* technique introduced by Borgefors [7] is fast and popular for alignment of structures by means of a distance transform. The drawback of segmentation-based registration is that the registration accuracy is limited to the segmentation step.

Voxel-based registration methods operate directly on the intensity values of the image without prior data reduction or segmentation. In some cases, registration is performed by aligning the center of gravity and principal orientation (Banerjee and Toga [10]). Principal axis methods are used if high accuracy is not essential but high processing speed and automatic operation is required.

## 4. Image Registration Pipeline

This article focuses on a hybrid pipelining approach to image alignment. 'Hybrid' in this context means that the properties of different algorithms have been combined in a modular pipeline, where each step of the pipeline either enhances the input images, detects landmark features in the images, or co-registers two images. The information that is propagated through the pipeline contains:

(i) a set of two consecutive input images,
(ii) a set of processed images from previous pipeline stages, and
(iii) image registration data from previous pipeline stages.

Each stage of the pipeline stores the results from exactly one previous run, so that the results from the current image pair can be correlated to the results from the previous image pair. Only one new image needs to be loaded into the pipeline at each time. The other input image is still stored from the previous run.

The advantage of this architecture is that the algorithm becomes scalable. This means that a global view on the data is not required. At each time, only two images are processed by the pipeline, using stored results from previous runs. This was one of the major design goals for the algorithm, because the size of the input data set (76 GB) exceeds by far the size of the main memory.

### 4.1 Horizontal and Vertical Data Flow

The pipeline represents a state machine. Each module of the pipeline stores its current state, and future runs depend on previous runs. Since some steps of the pipeline, such as *Histogram Equalization*, are global optimization steps, a two-pass protocol is necessary to collect all necessary data from the set of input images. This means, for instance, that in a first pass, all input images are read in order to calculate a global histogram, and in a second pass a transformation is applied to all images using the histogram data.

Pass 1:

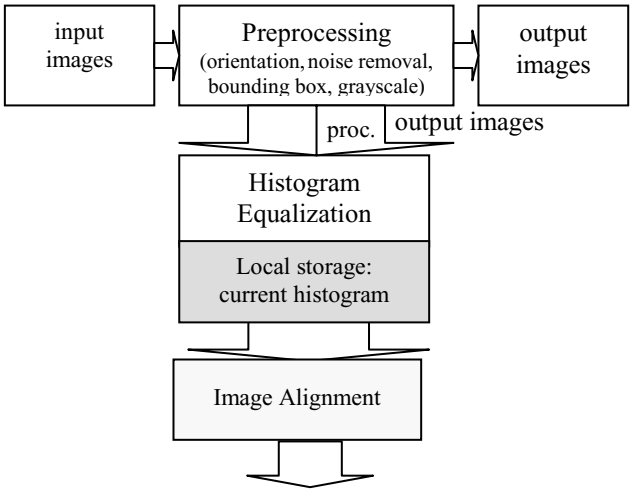


Figure 3: Data flow during first pass

Figures 3 and 4 illustrate the data flow during the first and second pass. The first pass is a data collection step. Histogram data is collected from all images and stored locally in the *Histogram Equalization* module. At the same time, the *Preprocessing* stage scales the images (section 4.8), eliminates noise, discards color information, and stores the resulting images in a new directory. In the second pass, a *Histogram Equalization* step is applied to the preprocessed images. At the same time, a bounding box is determined, and the other modules of the pipeline compute additional alignment data. All alignment data is collected in the last stage of the pipeline and applied to the original image data that is propagated together with the processed image data through the pipeline.

The following sections (4.2–4.5) summarize standard techniques that have been well established in the literature. Sections 4.6–4.8 describe special techniques that have been developed to aid in the alignment of a large-scale slice-based data set.

## 4.2 Preprocessing

This stage involves a sequence of image enhancement and noise removal filters to improve the quality of the image (less noise, more contrast) before it enters the rest of the image alignment pipeline.

### 4.2.1 Noise Removal

The noise present in the image is random in nature and is predominantly due to variations in the scanning process. The noise amplitude is usually low, but high-frequency components of the noise signal have a tendency to affect the quality of contour detection algorithms.

Therefore, the first stage in the pipeline is a noise removal step. It utilizes a common technique for noise removal known as "coring" [11]. In coring, an image signal is split into two or more bands; the high-pass bands are subjected to a threshold non-linearity that suppresses low amplitude values while allowing the high-amplitude values to pass through.

Pass 2:

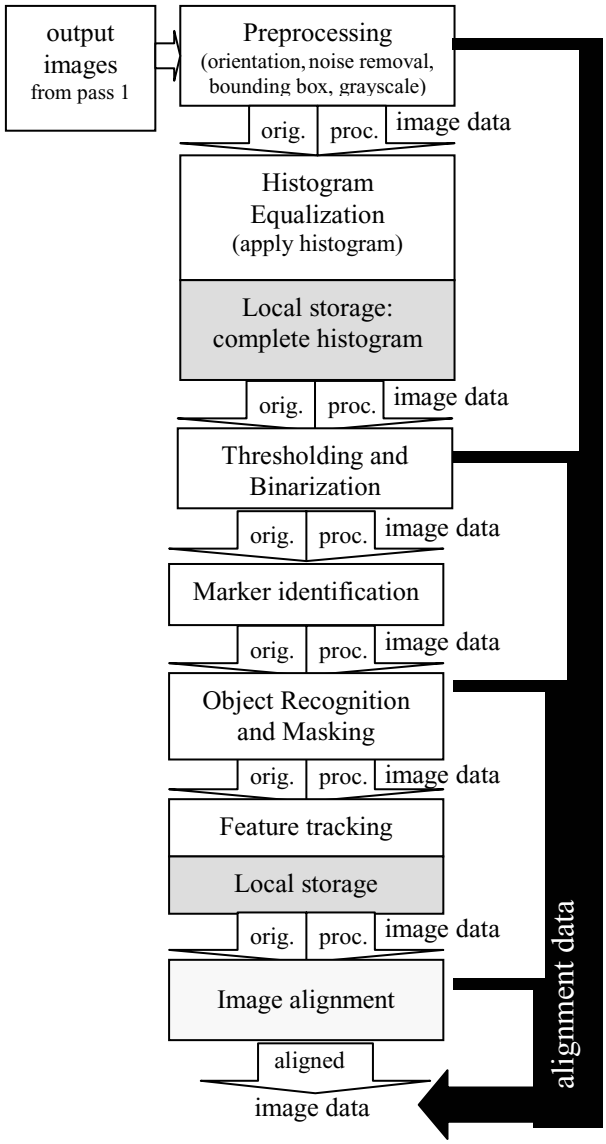


Figure 4: Data flow during second pass

### 4.2.2 Bounding Box Construction

Since the images include a lot of empty pixels, the redundant rows and columns surrounding the region of interest can be eliminated.

Suri *et al.* [13] have described the use of bounding boxes for performance improvement. They define the bounding box of a geometric object as a simple volume that encloses the object forming a conservative approximation of the object. The most common form is an axis aligned bounding box, where the extent of each dimension of space is bounded by the minimum and maximum coordinates of the object in that dimension.

A common approach for constructing a bounding box is to search the pixels in the image from all sides until a significant change in color value is observed. This means that those points are identified where a gradient exceeds a certain threshold. A simple method can be employed to compare the neighboring pixel colors [1]. If the difference

is significant (after noise removal), then we can assume that an edge of the object has been found. A bounding box helps narrowing down the search space, and the alignment is later refined in other stages of the pipeline.
Also, it is not sufficient to use all bounding boxes and center them based on the first, last, or center slice, because this would not necessarily reflect the actual shape of the object. The images could be aligned at the top, at the bottom, or somewhere in-between. Therefore, a lateral section of a similar brain was used to control the alignment (figure 5).

□ SHAPE * MERGEFORMAT □□□□

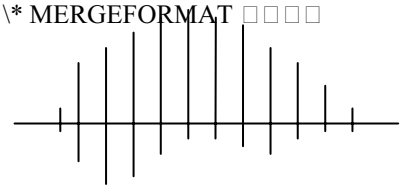

Figure 5: Side view of a set of bounding boxes aligned using a lateral slice to match the shape of the original specimen.

#### 4.2.3 Conversion to Grayscale

Since for the given data set the color does not carry any meaningful information that would help in the image alignment, we eliminate the color and convert the images to grayscale. For some data sets, another possible option would be to use a single channel from an image converted into the HLS or HSV color model.

The color information is only used in the final stage after the coordinates for alignment of the images have been determined. The alignment is then performed on the original color images.

### 4.3 Standardizing the Orientation

The original image slices are mostly oriented in a preferred direction. However, in general, all slices must be slightly rotated to fit a main center axis ('north-south' orientation). For computing the orientation it is necessary to find the correct angle of rotation with respect to this 'north-south' axis, which is defined as perpendicular to the *x* axis of the image.

Various algorithms deal with the problem of computing the orientation of an image [9]. Horn *et al.* [17] define the orientation of a region in an image by the direction of the axis of least inertia (minimum energy concept). The minimum energy concept evaluates the line for which the integral of the squares of the distances to feature points in the image takes on a minimum value,

$$E = \iint_I r^2 \; f(x, y)\, dx\, dy,$$

where $r$ is the perpendicular distance from the point $(x, y)$ to the line sought after, and $f(x, y)$ is the image function.

The orientation angle is evaluated and the image is rotated by that angle clockwise or anticlockwise to orient it towards the standard direction.

After this stage, all the images are oriented in the desired 'north-south' orientation as required.

### 4.4 Histogram Techniques

Histograms are graphs which are plotted between two parameters to indicate how many samples fall into each of a series of measurement intervals. A histogram *h(x)* represents the number of pixels in an image with intensity level *x*. A common histogram is computed for all image slices, followed by a histogram equalization performed on all slices of the set. This step helps to improve the overall contrast of the images and the quality of subsequent pipeline steps (contour detection, etc.)

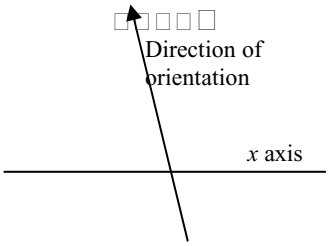


Figure 6: Arrow indicates orientation of a misaligned image slice with respect to the *x* axis of the image.

### 4.5 Thresholding

Using the histogram information a threshold can be computed [19]. Thresholding is a very useful tool for converting a grayscale image to a binary one. A side effect of such a 'binarization' is that the different lobes of the monkey brain can be more easily separated, and that the computation of the geometric center of the fragments is simplified.

In the equalized histogram (figure 8), we observe two peaks corresponding roughly to the object (left peak) and the background (right peak) with a low valley in between. For the 'binarization', we choose a threshold value where the histogram has a minimum (figure 9).

Figure 7: Normalized Histogram Figure 8: Equalization

### 4.6 Object Recognition

In the case of the Rhesus Macaque Monkey Brain data set the different lobes of the brain must be identified for individual alignment.
Once the binary image has been obtained, the foreground objects and the background objects are separately labeled.

Individual contiguous objects are assigned different labels so that they can be identified separately. A bit mask is used to describe the shape of each object.

Figure 9: Image after thresholding ('binarization')

For this form of object separation, we assume that the objects are 8-connected (loose connectivity) and the background is 4-connected (close connectivity). This way we ensure that even loosely connected objects are counted as single objects, and that the background appears as a solid area. Due to this asymmetry, remaining isolated noise pixel areas are not included in this segmentation. Morphology filters, such as dilation followed by shrinking filters, improve the connectivity of previously disjoint regions. These filters are part of the *Object Recognition and Masking* stage of the pipeline, and their window size is adapted to the size of the input image (approx. 1%, e.g., 3x3 for 256x256 pixels).

The following algorithm is used to label 4-connected components.

Input: Binary image $b(x, y)$

```
1.  current_label = 1
2.  for i = 0 to max_x
3.  label(i, 0) = 0
4.  for j = 0 to max_y
5.  label(0, j) = 0
6.  for j = 1 to max_y
7.     for i = 1 to max_x
8.        if (b(i, j) == 1) then
9.           if (b(i-1, j) == 0) and (b(i, j-1) == 0) then
10.             label(i, j) = current_label++
11.          if (b(i-1, j) == 1) and (b(i, j-1) == 0) then
12.             label(i, j) = label(i-1, j)
13.          if (b(i-1, j) == 0) and (b(i, j-1) == 1) then
14.             label(i, j) = label(i, j-1)
15.          if (b(i-1, j) == 1) and (b(i, j-1) == 1) then
16.             label(i, j) = label(i-1, j)
17.              // same as label(i, j-1)
```

Output: Label image $label(x, y)$

A similar algorithm is applied to the binary image in order to assign unique labels to 8-connected objects.

After labeling each object with separate masks, other valuable information required for the registration, such as the area of the object and the center of an area, is computed. The area of an object $k$ in the image (in our case, for example, one of the lobes of the brain) is the summation of all pixels b(x, y) that share label $k$ in the mask.

$$Area_k = \Sigma_x \Sigma_y \, (b(x, y) \;\&\; (label(x, y) == k)),$$

where, $b(x,y)$ denotes the binary values of the image pixels. The ampersand is a binary 'and' operator. The position of the center of an object is described by $x_c$ , $y_c$ :

$$x_c = \frac{\Sigma_x \Sigma_y \; x \, (b(x, y) \;\&\; (label(x, y) == k))}{Area_k}$$

$$y_c = \frac{\Sigma_x \Sigma_y \; y \, (b(x, y) \;\&\; (label(x, y) == k))}{Area_k}$$

When the center of each object is found (in addition to other feature points described below), the image registration can be performed by stacking the object centers on top of each other.

### 4.7 Identification of Image Markers

Shulga [1] has developed a method for automatic detection of pin holes of a particular size and shape. The algorithm trivially rejects all other holes that are not pin holes that meet the given criteria. In his algorithm, the processing starts from a slice near the center of the volume where both pin holes present in the current data set are visible. The user then selects the two pin holes for the first image. For the corresponding images the algorithm looks for the pin holes in a neighborhood defined by a given radius.

### 4.8 Image Pyramid

Keeping in mind the high level of detail in the images (figure 1b), we use a hierarchical image pyramid for simplification of the alignment problem. The resolution of the image is initially lowered, and since the color in stained cryosections usually does not bear any significant information for image alignment purposes, the color information is initially discarded in the interest of speed and efficiency. After the lower resolution slices are 'macroscopically' aligned by the pipeline, the 'microscopic' (cellular level) alignment needs to be performed with all the detail information and the color information present in the input and output images.

For instance, a reduction by a factor of $2^4$ in each dimension reduces the size of the input image (5,000 x 4,000 pixels) to a manageable size of 313 x 250 pixels, which is usually sufficient to compute a bounding box or to detect major landmark features. When up-scaling the coordinates of a contour or bounding box to match the resolution of the original input image, the uncertainty in the given example is $\pm$ 8 pixels. For most applications this can be tolerated. However, for optimal alignment a 'microscopic' feature detection algorithm can be employed that aligns objects within this 8-pixel search radius. This step could be added to the current pipeline. In our implementation, the image pyramid consists of two stages (full and reduced resolution).

## 5. Conclusions

The alignment of biomedical images is of major importance when combining monomodal or multimodal two-dimensional image slices into a 3-D volume. We presented a pipeline-based approach for misaligned monomodal image stacks that incorporates multiple steps: (i) detection of image feature points that are used as registration markers (pin holes, center points), (ii) linear, affine transformation steps (rotation, translation, shear), and (iii) non-linear transformation steps (individual alignment of objects based on segmentation). The difference between (ii) and (iii) is

that in (ii) all pixels in the image are transformed in the same way, while in (iii) the image is segmented into individual objects, so that different parts of the image are subject to different transformations.

We introduced a 2-pass pipelining approach that performs global computations (e.g., histogram) in a first pass, and then executes a sequence of algorithms on the image that both process the image and generate data for the final stage of the pipeline, i.e. for the *Image Alignment* stage. The data is propagated through the pipeline as shown in figures 3 and 4.

Besides some traditional techniques (4.2–4.5), we introduced several new techniques (4.6–4.8), such as an *Object Recognition* and labeling method, and a multiresolution technique that makes the algorithm scalable, i.e., independent of the size of the input data set. The pipeline was designed so that it can easily process data sets in the order of several gigabytes (the Rhesus Macaque Monkey brain data set has 76 GB).

A conscious effort has been to make the pipeline as robust and autonomous as possible, eliminating human assistance and interaction as much as possible. Besides, our special concern has been to make the algorithm as versatile as possible so that it can be used with many other data sets either in its present form or with minor modifications (reconfiguration of pipeline modules, different thresholds and parameters for each module).

After implementation of the first phase of the pipeline, a significant improvement was observed in the data set with the images exhibiting perfect alignment at the 'macroscopic' level.

## 6. Acknowledgements

This work was sponsored by the National Institute of Mental Health (NIH, award no. 5 P20 MH60975) under a subcontract with the Center for Neuroscience at University of California, Davis. The human brain images were provided by the courtesy of: Arthur W. Toga, Laboratory of NeuroImaging (LONI), University of California, Los Angeles. We would also like to thank Elke Moritz for her valuable contributions, and all the members of the Creative Interactive Visualization Laboratory (CIVL) at the University of California, Irvine.

## 7. References

1. Shulga, Dmitry; Meyer, Joerg: Aligning Large-scale Medical and Biological Data Sets: Exploring a Monkey Brain; Visualization, Imaging and Image Processing (VIIP 2001), The International Association of Science and Technology for Development (IASTED), Marbella, Spain, 2001, 434-439.
2. Pitiot, Alain., Malandain, Gregoire, Bardinet Eric, Thompson, M. P., Piecewise Affine Registration of Biological Images, WBIR, 2003, 91-101.
3. Evans, A. C., Marrett, S., Collins, L., and Peters, T. M. Anatomical-functional correlative analysis of human brain using three dimentional imaging systems. In Scheider, R. H., Dwyer III, S. J., and Jost, R.G.(eds), Medical imaging: image processing, , Bellingham, WA, SPIE press,Vol. 1092, 1989, 264-274.
4. Zubal, G. Tagare, H., Zhang, L., and Ducan, J. 3D registration of intermodality medical images. In proceedings of the annual international conference of the IEEE engineering in medicine and biology society, Vol. 13, 1991, 293-294.
5. Chen, Q. Image Registration and its application in Medical imaging. PhD. Thesis, Vrije universiteit Brussels, Brussels, Belgium, 1993.
6. Henderson, J. M., Smith, K. R., and Bucholz, R. D. An accurate and ergonomic method of registration for image-guided neurosurgery. Computerized medical imaging and graphics, 18(4), 1994, 273-277.
7. Borgefors, G. Hierarchaial chamfer matching: a parametric edge matching algorithm. IEEE Transactions on pattern analysis and machine intelligence, 10, 1998, 849-865.
8. Maintz, J.B., Viergever, M. A. A Survey of Medical Image Registration, Medical Image Analysis 2 (1), 1998, 1-36.
9. J.B.A. Maintz, P.A. van den Elsen, M.A. Viergever, "Registration of 3D medical images using simple morphological tools", in: *Information Processing in Medical Imaging*, Editor(s): J. Duncan, Springer Verlag, Vol. 1230, Lecture Notes in Computer Science, 1997, 204-217.
10. Banerjee, P. K. and Toga, A. W. Image alignment by integrated rotational and translational matrix. Physics in medicine and biology, 39, 1994, 1969-1988.
11. Simoncelli, Eero P, Adelson, Edward H., Noise Removal via Bayesian Wavelet Coring. Proceedings of 3rd IEEE International Conference on Image Processing. Vol. I, Lausanne, Switzerland, 1996, 379-382.
12. Malladi, R., Sethian, J.A., A Unified Approach to Noice Removal, Image Enhancement, and Shape Recovery. *IEEE Transactions on Image Processing,* Vol. 5(11), 1996, 1554-1568.
13. Subhash Suri, Philip M. Hubbard, John F. Hughes: Analyzing bounding boxes for object intersection. ACM Trans. Graph., 18(3), 1999, 257-277.
14. Foley, James D., Andries van Dam, Steven K. Feiner, and John F. Hughes. 1996. Computer Graphics: principles and practice. Glenview, IL, Addison-Wesley Publishing Company Inc.
15. J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1994, 593-600.
16. Tiziano Tommasini, Andrea Fusiello, Emanuele Trucco, Vito Roberto: Making Good Features Track Better. CVPR, 1998, 178-183.
17. Horn, B. K. P, McGraw Hill (MIT Press), 1986.
18. Arai, T. and K. Machii and S. Kuzunuki. Retrieving Electronic Documents with Real-World Objects on InteractiveDesk. UIST, 1995, 37-38.