

# 3D RECONSTRUCTION FROM A VASCULAR TREE MODEL

Luis Álvarez, Karina Baños, Carmelo Cuenca, Julio Esclarín, and Javier  
Sánchez

Computer Science Department  
Las Palmas University, Las Palmas 35017 SPAIN  
{lalvarez, kbaños, ccuenca, jesclarin, jsanchez}@dis.ulpgc.es,  
WWW home page: <http://serdis.dis.ulpgc.es/~lalvarez/ami/index.html>

**Resumen** In this paper, we present a vascular tree model made with synthetic materials and which allows us to obtain images to make a 3D reconstruction. We have used PVC tubes of several diameters and lengths that will let us evaluate the accuracy of our 3D reconstruction. In order to calibrate the camera we have used a corner detector. Also we have used Optical Flow techniques to follow the points through the images going and going back. We describe two general techniques to extract a sequence of corresponding points from multiple views of an object. The resulting sequence of points will be used later to reconstruct a set of 3D points representing the object surfaces on the scene. We have made the 3D reconstruction choosing by chance a couple of images and we have calculated the projection error. After several repetitions, we have found the best 3D location for the point.

## 1. Introduction

Given the difficulties to obtain medical images in good conditions, due to their privacy and diverse technical problems that these can present, such as: artifacts, occlusions, poor definition of the image, etc; we have developed a model of vascular tree using PVC.

This way we can obtain a series of images following the same technology of a rotational angiography, planning an arch of fixed radius about the model and obtaining the images with an angular separation of about 3 degrees. Once obtained, they were manipulated in order that the final result is as similar as possible to the angiographies, but without the problems previously mentioned. Besides, by having the 3D model it is possible to test the quality of the results obtained, knowing exact values such as: distances between bifurcations, the diameters of the glasses, etc; which will allow us to know the kindness of our results.

## 2. Obtaining the images

We want to reconstruct images to apply in the Rotational Angiography field that it has the next features:

## II

The series of images in Rotational Angiography is acquired while the imaging assembly rotates in a continuous arc around the patient. The whole acquisition is rather fast, so that the complete series can be acquired with a single injection of contrast agent.

The quality of the individual images is generally fully adequate for diagnosis, with the following added advantages: wide range of projections, optimum views of vascular structures.

For the 3D reconstruction, it is essential for the images to precisely match each other. This requires an extremely stable and reproducible image geometry. The system is calibrated to compensate for distortion in the image intensifier such as pincushion and the varying distortion caused by movement through the magnetic field of the earth.

The images are acquired in the rotational angiography mode over an angle of 180 degrees. The run may be carried out in one of three different angulations: -30 degrees cranial, 0 degrees axial, 30 degrees caudal. Images are acquired at a frames rate of 12.5 frames by seconds, and a rotation speed of up to 30 degrees per second, the whole acquisition takes 8 seconds resulting in an average of 100 images per run.

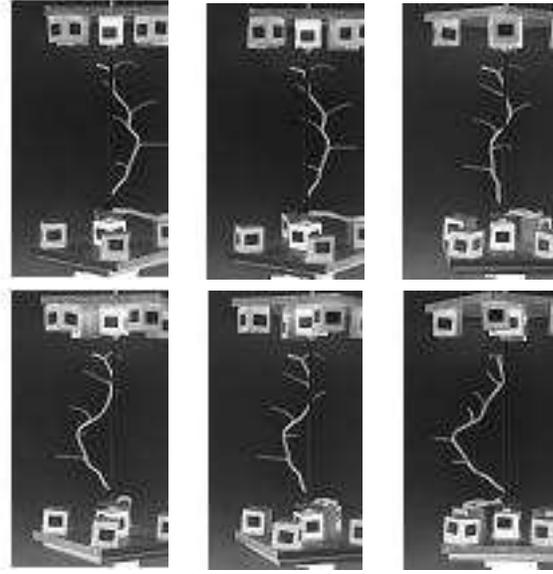
We have made the images with an angle of 3.6 degrees and approximately three meters of distance from the model using a of 70 mm focal distance with a digital camera and because the characteristics of its digitizer then we have a focal distance 105 mm with an arc of 270 degrees that allows around 70 images. We have also obtained images of a calibrator in order to obtain the intrinsic parameters of the camera.

### 3. Multiscale Analysis and Calibration Camera

One of the main concepts of vision theory and image analysis is multiscale analysis. A Multiscale Analysis  $T_t$  associates, with an original image  $u(0) = u_0$  a sequence of smoothed images  $u(t, x, y)$  which depend upon an abstract parameter  $t > 0$ , the scale.

$$f(x, y) \longrightarrow u(t, x, y)u(0, x, y) = f(x, y)$$

The datum of  $u_0(x, y)$  is not absolute in perception theory, but can be considered as the element of an equivalence class. If  $A$  is any affine map to the plane,  $u_0(x, y)$  and  $u_0(A(x, y))$  can be assumed equivalent from a perceptual point of view. Last but not least, the observation of  $u_0(x, y)$  does not generally give any reliable information about the number of photons sent by any visible place to the optical sensor. Therefore, the equivalence class in consideration will be  $g(u_0(A(x, y)))$ , where  $g$  stands for any contrast function depending on the sensor. These considerations lead us to focus on the only multiscale analyses which satisfies these invariance requirements : The Affine Morphological Scale Space (AMSS). This multiscale analyses can be defined by a simple Partial Differential Equation:



**Figure 1.** Acquired images.

$$u_t = t^{\frac{1}{3}}(u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy})^{\frac{1}{3}}$$

where  $u(t,x,y)$  denotes the image analyzed at the scale  $t$  and the point  $(x,y)$ .

In order to calibrate a camera system we need to corner detection and this is very sensitive to noise. The AMSS multiscale analysis present the advantage that we know, analytically, the displacement of the corner location across the scales. Then we can search it in at the scale  $t_n = t_0 + n\Delta t$ , for  $n=1,\dots,N$ , where  $U_t$  represents the discretization step for the scale and  $t_0$  represents the initial scale that we use to begin to look for corners.

We compute for the scale  $t_0$  the location of the extreme of the curvature that we denote by  $(x_0^i, y_0^i)$ , for  $i=1,\dots, M$ , these points represent the initial candidates to be corners. We follow across the location  $(x_n^i, y_n^i)$  of the curvature extreme.

For each sequence  $(x_n^i, y_n^i)$   $n=1,\dots,N$ , we compute in a robust way (using orthogonal regression and eliminating outliers) the best line which fit the sequence of points, this line corresponds to the bisector line of the corner, and we can represent it as a straight line which equation:

$$(x(t), y(t)) = (X_0, Y_0) + \tan\left(\frac{\alpha}{2}\right)^{-\frac{1}{2}} t (b_x, b_y)$$

where  $\alpha$  is the angle of the corner and  $\vec{b} = (b_x, b_y)$ , is the unit vector in the direction of the bisector line of the corner, and  $t$  is the scale. Then we can find the corner doing  $t = 0$  in this equation.

In order to calibrate the camera system, we extract the characteristics of the sequence of views with a morphologic corner detector. This detector gives us sub pixel information. When the views are taken from very close positions, the conventional methods of calibration can be unstable, to solve this problem we divide the sequences of views into several sub sequences (in this way the optical centre displacements are bigger). Now, we calibrate every subsequence of view in an independent way.

In the last step, we make the calibration between the different sub sequences to obtain only one calibration. The method we have used is very stable, even when there are noise and small displacements between the optical centres.

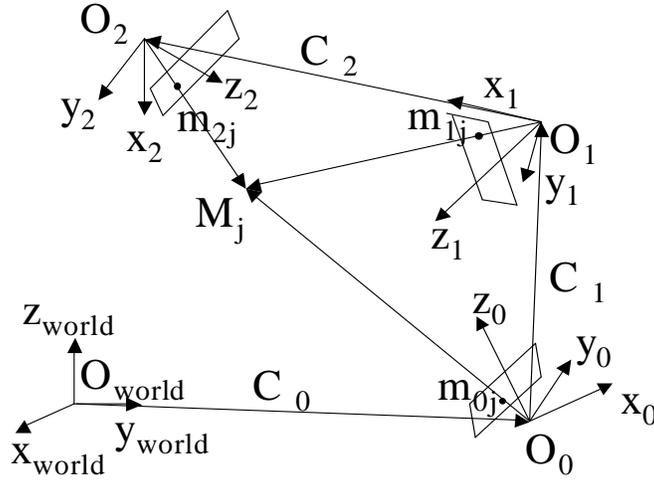
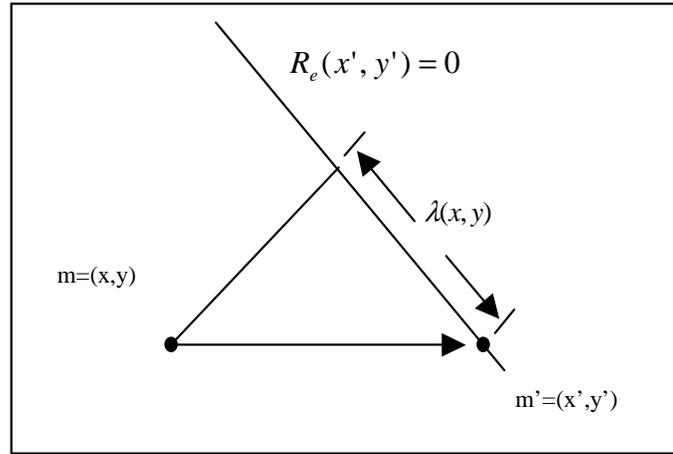


Figure 2. Motion parameters derived from point matches.

#### 4. Optical Flow

Once we have calibrated the camera system we propose a method for the recovering of disparity maps between pair of stereoscopic images. Disparity maps are obtained through a matching process in where we have to find for the pixels in the left image their correspondent on the right image. There are some methods, like correlation-based techniques, that estimate good matching points but do not generate smooth disparity maps for the whole image, so the solutions in this case are not continuous.

To improve the accuracy of the matching process we make use of the so-called epipolar geometry. This geometry represents the relation that exists between stereoscopic images. Thanks to this geometry, the method is able to look for correspondences in straight lines only.



**Figure 3.** Displacement function is parameterized in order to take advantage of the information given by the epipolar geometry.

The method we propose for the computation of disparity maps is based on an energy minimization approach:

$$E(\lambda) = \int (I(\vec{x}) - I'(\vec{x} + \vec{h}(\lambda)))^2 + \int \nabla \lambda D(\nabla I) \nabla \lambda$$

Where  $I$  and  $I'$  are the stereoscopic images,  $\lambda$  is a parameter that gives us the distance between the point that it is the projection of  $m$  over the epipolar straight line and the responsive point of  $m$  in  $I'$ ,  $m'$  (see image 3). The second term in the equation is used to regularize.  $D$  is a diffusion tensor that diffuses in one or another way depending where the point is placed. If gradient of  $I$  is high then the regularization is over the contour line and if it is low, we make regularization.

This energy consists of an attachment term that enables the process to find similar pixels in both images and a regularization term that is necessary to constraint the number of possible solutions and to generate smooth solutions. This method is a dense method in the sense that for every pixel on one image we obtain its correspondent on the other image.

When we minimize this energy, we obtain the Euler-Lagrange equations, which are represented by means of partial differential equations. This is a diffusion-reaction equation that behaves anisotropically at contours with high values for the gradient of the images and isotropically at homogeneous regions where the image gradient is low. The diffusion part is formulated in such a way that the discontinuities of the images are preserved. We use a scale-space and pyramidal strategy to allow the method to locate large displacements. Thanks to this energy minimization approach the resulting disparity maps that we may obtain are smooth by regions.

## 5. Corresponding Points and 3D Reconstruction.

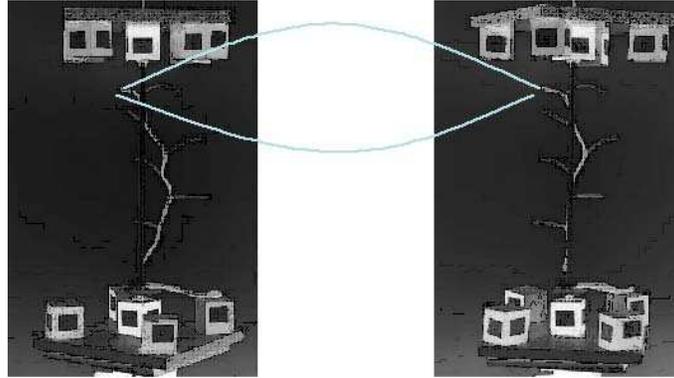
To search the corresponding points in every image, we start with a point in one image, and using the optical flow techniques, we search the corresponding point in the next image. Once this point is obtained, we go back and we search if the corresponding point in the first image is the start point. If it is true, we continue searching other point in the next image. Once we have this new point we come back until the first image verifying that the points calculates are the same points that we have with a small error. We finish this process when we search a point bad placed.

First, we will describe an efficient algorithm for computing sequences of corresponding points with maximum lengths. Second, we will describe a fast algorithm for computing sequences of corresponding points, but in this case without maximum lengths. Experimental results show the validity of this algorithm. Both algorithms try to include a candidate to a sequence of  $n$  corresponding points to obtain a sequence of  $n + 1$  points. We will explain different quality criterions, which are applicable to decide whether to add a new point to the sequence, or not.

In order to calculate the sequence of points in correspondence, we must know the Optical Flow from every view  $j$  to the precedent view  $j - 1$  (backward), that we denote by  $h_-^j(x) = h_-^j(x, y) = (u_-^j(x, y), v_-^j(x, y))^T$  and the Optical Flow from the view  $j$  to the next view  $j + 1$  (forward),  $h_+^j(x) = h_+^j(x, y) = (u_+^j(x, y), v_+^j(x, y))^T$ .

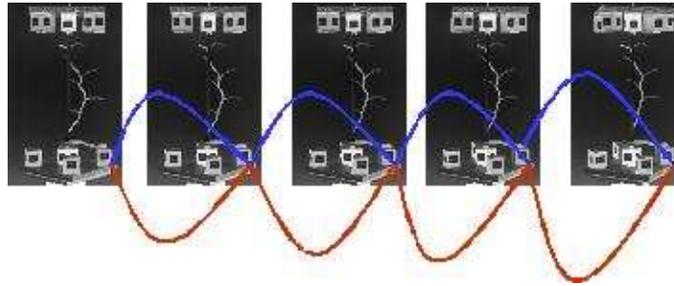
The main idea is using the Optical Flow  $h_+^j(x)$  to localize the correspondent point  $x'$  in the view  $j + 1$ , from the point  $x$  in the view  $j$ , and using the Optical Flow  $h_-^{j+1}(x)$  in the opposite direction from the view  $j+1$  to the view  $j$  to come back from the point  $x'$  in the view  $j + 1$  to a point  $x^{prime}$  in the view  $j$ . Then we have, from the Optical Flow definitions,  $x' = x + h_+^j(x)$  and  $x^{prime} = x' + h_-^{j+1}(x') = x + h_+^j(x) + h_-^{j+1}(x + h_+^j(x))$ . In ideal conditions, both points,  $x$  and  $x''$  must be the same point, but in real conditions, this is usually false, due to several reasons as limitations to calculate the Optical Flow, imprecision in the Camera calibration, numerical errors, etc. In those cases, we have a distance between these points  $d(x, x'') = \|x'' - x\|$ . With this idea we can construct sequences of couples of points from every couple of images. Then the distance  $d(x, x'')$  can be used as a criterion of quality.

We can extend this idea to obtain sequences with more than two points. One point  $x_j$ , in the view  $j$ , has its corresponding point  $x_{j+1} = x_j + h_+^j(x_j)$ , in the view  $j + 1$ , and comes back to the point  $x'_j = x_{j+1} + h_-^{j+1}(x_{j+1})$  with an error  $d_j(x_j, x'_j)$ (forward and backward). To add a new point to the sequence, we use the point  $x_{j+1}$  in the view  $j + 1$  and the Optical Flow  $h_+^{j+1}(x)$  to go from the view  $j + 1$  to the view  $j + 2$ . In this way we obtain, in this view, the point  $x_{j+2}$ , in correspondence with the point  $x_{j+1}$ , in the view  $j + 1$ . The backward Optical Flow, allows us to establish a first measure of quality of the new sequence of three points. In this way, in the view  $j + 1$ , we have the error forward and backward



**Figure 4.** Backward and Forward Optical Flow for a point in a couple of views.

$d_{j+1}(x_{j+1}, x'_{j+1})$  where  $x'_{j+1} = x_{j+2} + h_-^{j+2}(x_{j+2})$ , and in the view  $j$ , the error forward and backward is  $d_j(x_j, x'_j)$  with  $x'_j = x'_{j+1} + h_-^{j+1}(x'_{j+1})$ .



**Figure 5.** Backward and Forward Optical Flow for a sequence of several points.

Generally, to add a new point  $x_{j+n}$  to the sequence of  $n$  points in correspondence  $x_j, x_{j+1}, \dots, x_{j+n-1}$ , we use the last point added to the sequence,  $x_{j+n-1}$ , and the Optical Flow  $h_+^{j+n-1}(x)$  to calculate the point  $x_{j+n}$  in the view  $j+n$ . This point could become a new point in the sequence of corresponding points,  $x_{j+n} = x_{j+n-1} + h_+^{j+n-1}(x_{j+n-1})$ . To do that, we calculate the errors backward  $d_i(x_i, x'_i)$  ( $i = j, j+1, \dots, j+n-1$ ) where we use the Optical Flow  $h_-^{i+1}(x)$  to calculate the point backward  $x'_i$ , in the view  $i$ , because  $x'_i = x'_{i+1} + h_-^{i+1}(x'_{i+1})'$ . This point will be added to the sequence if these errors are lower than a threshold that we have defined.

We have seen that the forward and backward error is an initial quality measure of a sequence of corresponding points. The forward and backward error also allows us to establish a first test to decide whether to add a new point to the sequence or not. When we have a large number of views the initial point of

a sequence will not appear on the view of the candidate point and, therefore, the forward and backward errors will be big. For a better 3D reconstruction, it is convenient that the sequence of points would be as large as possible in order to turn the algorithm less sensible to errors. We may define more complex criterions to obtain sequences with many points. Instead of using a constant threshold for the forward and backward errors, we may adapt this threshold to the sequence length. This would benefit larger sequences of points with bigger error rather than short sequences with a smaller error. Another possible criterion that would be more robust is that of using the candidate point, the sequence of corresponding points and the calibration matrices to reconstruct the 3D point and measure the reprojected errors on the projection planes. In case that the reprojected errors would be smaller than a given threshold we would include the candidate point into the sequence. It is also possible and adaptive scheme to benefit the larger sequences.

```
*****
*****
```

### 5.1. The Optimal Algorithm

The Optimal Algorithm to calculate sequences of points in correspondence is the following:

1. For every view
  - a) For every pixel
    - 1) Calculate the sequence of points in correspondence and its quality
    - 2) Add the sequence of points in correspondence calculated in the previous step to an ordered repository. The first order criterion is the number of points and the second one is the quality of the sequence of points in correspondence
2. For every sequence in the sequence repository
  - a) Check that the sequence of points in correspondence is not included in another sequence of points in correspondence with a higher number of points, or with the same number of points but with a higher quality
    - 1) If the sequence of points in correspondence is not included in any final repository of sequences of points in correspondence, then add the sequence to the final repository of sequences
    - 2) If the sequence of points is included, then reject the sequence

In the step iterates through all views building for each pixel on the view a sequence of corresponding points and assigning a quality value.  $n$  views will produce  $n * widthview * heightview$  sequences of corresponding points. In the step the sequences of corresponding points are selected. Normally some of the sequences will be included on larger sequences or they would be as close to consider that they are practically the same sequence. Although this algorithm constructs large sequences of corresponding points, it has the disadvantage that it is computation costly. Step ... takes into account that a new sequence will not

be introduced into the repository if the sequence already exists. At the beginning when the repository size is small it is fast to test if a sequence is already included, but when the size is increased the number of the comparisons will also increase making the process much slower.

## 5.2. The Fast Algorithm

To decrease the computational cost of the efficient algorithm presented above we modify the algorithm by adding a flag to every pixel. This flag indicates if a pixel is included in a sequence of corresponding points. The algorithm would be as follows:

1. For every view
  - a) For pixel on the view not included in a sequence
    - 1) Compute the sequence of corresponding points. Mark every point in the sequence as included in a sequence of corresponding points
    - 2) The computed sequence in the previous step is added to a repository

Contrary to the efficient algorithm, the fast algorithm does not guarantee that the sequences of corresponding points would contain the larger number of points.

When we have a set of images, then we take a couple of cameras by chance and we reconstruct the 3D point. We project this point to the plane of the whole cameras and we calculate the projection error, the distance between both points, the real point and the projected point in every camera.

Later we take another couple of cameras, always by chance, and we repeat the same steps several times, up to 8 times if it is possible. We keep with the point that minimizes the projection error.

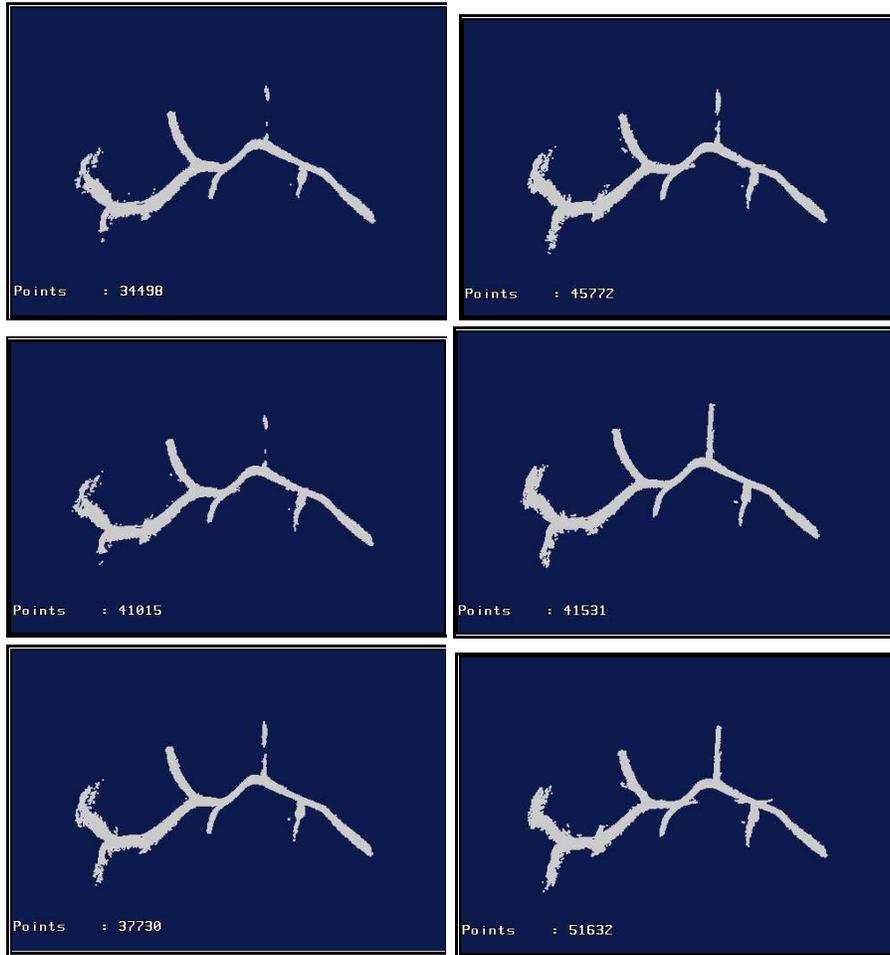
## 6. Results

In order to obtain the 3D reconstruction we have used two sets of images, first set has 16 images, and second set has 25 images. And we have used three different thresholds of projection error: 1.0, 1.2 and 1.5. In the next images, we can see the different 3D reconstructions:

When we use more views to make the 3D Reconstruction, we have more points, but the time to do it is bigger. In the other hand when the projection error increases, the number of points increases too, but the points are located with less precision.

## Referencias

1. L. Álvarez, K. Baños, C. Cuenca, J. Esclarín and J. Sánchez. 3D reconstruction from a vascular tree model, EUROCAST, Las Palmas, 2003.



**Figura 6.** 3D Reconstruction in the first line we have the results with 16 views in the second line we have the results with 25 images and from left to right we have the image with a projection error 1.0, 1.2 and 1.5.

2. L. Álvarez, C. Cuenca. Calibración de multiples cámaras utilizando objetos de calibración esféricos. Caepia 2001.
3. L. Álvarez, R. Deriche, J. Sánchez, and J. Weickert. Dense disparity map estimation respecting image derivatives: a pde and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13:3–21, January 2002. Also published as Inria Research Report no 3874.
4. L. Álvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1):41–56, 2000.
5. O. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
6. Hans-Hellmut Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from images sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, 1986.
7. Olivier Faugeras and Renaud Keriven. Complete dense stereovision using level set methods. *Proceedings of Fifth European Conference on Computer Vision*, 1998.
8. J. Weickert. *Anisotropic Diffusion in Image Processing*, Teubner, Stuttgart, 1998.
9. P. Peene, P. Cleeren, B. D’Herde, L. Storme, J. Vanrusselt and G. Souverijns. Non/subtracted Rotational Angiography on a Multipurpose Digital C/arm Radiography System. *Medica Mundi*, 1999.
10. J. Moret, R. Kemkers, J. Op de Beek, R. Koppe, E. Klotz and M. Grass. *3D Rotational Angiography: Clinic value in Endovascular treatment*. *Medicamundi*. 1998
11. L. Robert, R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. B. Buxton, R. Cipolla (Eds.), *Computer vision - ECCV’96, Volume I, Lecture Notes in Computer Science*, Vol. 1064, Springer, Berlin, 439-451, 1996.