**The Algorithm**

The method proposed is called Multiplicative Intrinsic Component Optimization (MICO). I performs both image segmentation and bias field removal. It does this by modeling the image and its characteristics in an advantageous way and then it uses energy minimization to optimize each problem of the pipeline.

This paper proposes an optimization algorithm based on energy minimization. The constraints of the problem are dictated by the above mentioned characteristics of the

**Image Model**

MR images are decomposed into 2 multiplicative components: the true image characterizes the physical properties of each tissue and the bias field accounts for intensity inhomogeneity through the image.

The key characteristic of the true image is that all voxels/pixels of a certain tissue have a constant values. Whereas the key characteristic of the bias field model is that it varies smoothly. Both these characteristics will be taken advantage of in the processes that are to come next in the pipeline.

**Bias Field Approximation**

The bias field is approximated as being a linear combination of a given set of smooth basis functions (20 polynomials of the first 3 degrees). The algorithm allows us to change the number of these basis function by the use of variable *Bas* and to compute them, we have to use the *getBasisOrder3* function.

The optimizer finds the ideal vector of weights for each of these basis function to better approximate the underlying inhomogeneity in the image.

The removal of the bias field is done by dividing the affected image by the approximation found by the algorithm.

**Image Segmentation**

This sub-routine of the algorithm takes advantage of the properties of the true image. The optimization algorithm approximates a constant for each voxel of a certain tissue. In the code, we can select the number of tissues by manipulating the variable *N*; *k* is the iterator that represents the current tissue.

Of course, a voxel may contain more than one tissue due to the partial volume effect. To combat this, we, the image is split into regions in which we assign tissue values based on membership functions. In the code, these functions are stored in the variable M. These membership functions can be either hard (1 or 0 i.e. “the voxel is tissue *k*”) or fuzzy. In the fuzzy case, we allow for a soft segmentation with confidence scores between 1 and 0. We can control the fuzzy math in the program by use of the variable *q* – the fuzzifier. The larger *q* is, the more pronounced the fuzzy effects on the segmentation.

Ultimately, the true image is modeled as a sim of the products of the constants assigned to each tissue and the values of the membership functions that describes its voxels, be it fuzzy or hard.

**The 3D Case**

The conceptual idea behind the 3D implementation remains the same. The variations are only of a technical nature. Special routines are used to work with the 3D (nifti) files and *meshgrid* is used to generate operational arrays for the calculations that are performed.

**Experimental Results**