HW05 Report

Medical Image Processing

Dr. Emad Fatemizadeh

Ali Nourian - 98102238

Electrical Engineering

Date: July 3, 2023

1 Theoretical Questions

1.1 Question 1

Intensity inhomogeneity (IIH) in images is a phenomenon that occurs when the intensity between adjacent pixels is different. It can be caused by various factors, such as occlusion, uneven heat dissipation, lighting conditions, or imaging devices. IIH can affect the quality and accuracy of image segmentation, which is a process of dividing an image into meaningful regions or objects.

One of the methods to deal with IIH is the active contour model (ACM), which is a technique that uses a curve or surface to evolve towards the object boundaries by minimizing an energy functional. The energy functional usually consists of three terms: a region fitting term, a smoothness term, and an external force term.

There are several possible causes of intensity inhomogeneity in MRI, such as:

- Imperfections of the image acquisition process, such as nonuniform radiofrequency coils, gradient field nonlinearity, eddy currents, etc.
- Irregular anatomical areas, such as shoulder, hips, ankles, etc. that cause magnetic susceptibility artifacts.
- Presence of metallic objects or implants that distort the magnetic field.
- Patient motion or breathing that introduce motion artifacts.

However, the traditional ACMs may not work well for images with IIH, because they assume that the intensity within each region is homogeneous or can be corrected by a bias field. Therefore, some modified ACMs have been proposed to handle IIH more effectively. One of them is the adaptive fuzzy c-means clustering (AFCM) method, which is based on the fuzzy c-means (FCM) algorithm.

The FCM algorithm is a clustering method that assigns each pixel to one or more clusters based on its similarity to the cluster centers. The similarity is measured by a membership function, which indicates the degree of belongingness of each pixel to each cluster. The FCM algorithm iteratively updates the cluster centers and the membership functions until they converge. The AFCM method extends the FCM algorithm by introducing an adaptive penalty term and a spatial constraint term into the objective function. The adaptive penalty term penalizes the pixels that have large intensity differences from their cluster centers, while the spatial constraint term incorporates the spatial information of neighboring pixels into the membership function. These two terms help to reduce the effect of IIH and noise on the segmentation results.

The AFCM method can be combined with an ACM to form a hybrid model for image segmentation. The AFCM method provides an initial segmentation and an intensity inhomogeneity correction for the ACM, while the ACM refines the segmentation and enforces the boundary smoothness. The hybrid model can achieve better segmentation performance than either method alone.

1.2 Question 2

1.3 1

The fundamental concept behind the level set method is to represent a contour as the zero level set of a higher-dimensional function known as a level set function (LSF). This approach allows us to describe the motion of the contour as the evolution of the level set function. Instead of explicitly parameterizing curves or surfaces, the level set method enables numerical computations on a fixed Cartesian grid, known as the Eulerian approach. It provides a flexible framework for analyzing shapes and surfaces without the need to handle complex topological changes, such as shape splitting, merging, or the development of holes.

In the level set method, the level set function acts as a representation of the shape or contour of interest. The zero level set of the function corresponds to the boundary of the shape, while the positive or negative values of the function indicate the interior or exterior regions, respectively. By evolving the level set function over time, we can track the changes in the shape or contour.

One of the key advantages of the level set method is its ability to handle shapes with changing topology. For example, when a shape splits into two or merges back together, it can be challenging to describe these transformations using traditional parameterization methods. However, with the level set approach, these changes in topology can be easily captured and tracked by simply evolving the level set function. This makes the level set method particularly useful for modeling dynamic objects, such as the inflation of an airbag or the behavior of a drop of oil floating in water.

The level set method allows us to perform computations and simulations on a fixed Cartesian grid, which simplifies the numerical analysis of shapes and surfaces. It eliminates the need for explicit parameterizations of the objects of interest and provides a convenient way to handle complex shape deformations. By working with the level set function, we can easily observe and analyze changes in the shape's topology, making it a powerful tool for studying time-varying objects.

The level set method offers several advantages that make it a desirable approach for various applications:

1. Complex Topology Representation: One of the significant advantages of level set methods is their ability to represent contours with complex topology. Unlike parametric active contour models, level set methods can handle topological changes, such as splitting and merging, in a natural and efficient way. This means that the method can accurately capture and track shapes that undergo complex transformations without requiring additional indirect procedures in the implementation.
2. Grid-Based Computations: Another advantage of level set methods is the ability to perform numerical computations on a fixed Cartesian grid. This eliminates the need to explicitly parameterize the points on a contour, which is required in parametric active contour models. By working with a fixed grid, computations become more straightforward and efficient, making it easier to analyze and manipulate shapes and surfaces.

These advantages make the level set method highly versatile and applicable in various fields, including image processing, computer graphics, computational geometry, optimization, computational fluid dynamics, and computational biology. The method's ability to handle complex topology and perform computations on a grid simplifies the modeling and analysis of shapes and surfaces, making it a powerful tool in numerous applications.

The evolution of a curve can be mathematically expressed as:

$$\frac{\partial C(s,t)}{\partial t} = F\mathcal{N}$$

In this equation, F represents the speed function that controls the motion of the contour, and $\mathcal{N}$ represents the inward normal vector to the curve C.

This equation describes how the curve C changes over time. The left-hand side of the equation represents the rate of change of the curve with respect to time, $\frac{\partial C(s,t)}{\partial t}$. The right-hand side of the equation is a product of the speed function F and the inward normal vector $\mathcal{N}$.

The speed function F determines the speed at which the curve evolves. It can be designed based on various factors, such as image gradients, edge information, or user-defined constraints. The choice of the speed function depends on the specific application and desired behavior of the curve evolution.

The inward normal vector $\mathcal{N}$ is a vector that is perpendicular to the curve C at each point. It represents the direction in which the curve is moving. The inward normal vector is essential for guiding the evolution of the curve towards or away from certain features or regions in the image or space.

By solving this differential equation, the curve C can be evolved over time based on the given speed function F and the inward normal vector $\mathcal{N}$. This allows for the dynamic deformation and evolution of the curve to adapt to the desired characteristics or features.

The process of converting the curve evolution equation, which is expressed in terms of a parameterized contour, to a level set formulation involves embedding the dynamic contour as the zero level set of a time-dependent Level Set Function (LSF). When the embedding LSF takes negative values inside the zero level contour and positive values outside, the inward normal vector can be represented as $\mathcal{N} = \frac{-\nabla\phi}{|\nabla\phi|}$, where ϕ is the gradient operator. Consequently, the curve evolution equation is transformed into the following Partial Differential Equation (PDE):

$$\frac{\partial\phi}{\partial t} = F|\nabla\phi|$$

In the level set formulation, the contour is represented implicitly as the zero level set of the LSF ϕ. The LSF is a scalar field defined over the entire domain, where the zero level set corresponds to the evolving contour. The sign of the LSF inside and outside the contour provides a way to distinguish between the two regions.

To calculate the inward normal vector $\mathcal{N}$, the gradient of the LSF ϕ is computed using the gradient operator ∇. The negative gradient $-\nabla\phi$ points inward and is divided by its magnitude $|\nabla\phi|$ to normalize it, resulting in the inward normal vector $\mathcal{N}$.

The converted PDE $\frac{\partial\phi}{\partial t} = F|\nabla\phi|$ describes the evolution of the LSF ϕ over time. The right-hand side of the equation involves the speed function F multiplied by the magnitude of the gradient $|\nabla\phi|$. This formulation enables the LSF to evolve in a way that is influenced by the speed function and the local geometry of the contour.

By solving this PDE numerically, the LSF ϕ evolves, and the zero level set tracks the contour's motion over time. The level set formulation provides a versatile framework for handling complex topological changes, such as splitting and merging of contours, and allows for efficient computations on a fixed Cartesian grid.

1.4 2

$$\varepsilon(\theta) = \mu R_p(\phi) + \varepsilon_{ext}(\phi)$$

where $R_p(\phi)$ is the level set regularization term defined in th following, $\mu > 0$ is a constant, and $\varepsilon_{ext}(\phi)$ is the external energy that depends upon the data of interest (e.g., an image for image segmentation applications). The level set regularization term ($R_p(\phi)$) is defined by:

$$R_p(\phi) = \int_\Omega p|\nabla_\phi|dx$$

The energy $\varepsilon_{ext}(\phi)$ is designed such that it achieves a minimum when the zero level set of the LSF ϕ is located at desired position (e.g., an object boundary for image segmentation applications).

A naive choice of the potential function is $p(s) = s^2$ for the regularization term R_p, which forces $|\nabla_\phi|$ to be zero. Such a level set regularization term has a strong smoothing effect, but it tends to flatten the LSF and finally make the zero level contour disappear. In fact, the purpose of imposing the level set regularization term is not only to smooth the LSF ϕ, but also to maintain the signed distance property $|\nabla_\phi| = 1$, at least in a vicinity of the zero level set, in order to ensure accurate computation for curve evolution. This goal can be achieved by using a potential function $p(s)$ with a minimum point $s=1$, such that the level set regularization term $R_p(\phi)$ is minimized when $|\nabla_\phi|=1$ .Therefore, the potential function should have a minimum point at $s=1$ (it may have other minimum points). We will use such a potential p in the proposed variational level set formulation. The corresponding level set regularization term $R_p(\phi)$ is referred to as a distance regularization term for its role of maintaining the signed distance property of the LSF. A simple and straightforward definition of the potential for distance regularization is

$$p = p_1(s) = \frac{1}{2}(s-1)^2$$

which has $s=1$ as the unique minimum point. With this potential $p=p_1(s)$, the level set regularization term can be explicitly expressed as:

$$P(\phi) = \frac{1}{2}\int_\Omega (|\nabla_\phi|-1)^2 dx$$

The energy functional $P(\phi)$ was proposed as a penalty term in an attempt to maintain the signed distance property in the entire domain. However, the derived level set evolution for energy minimization has an undesirable side effect on the LSF ϕ in some circumstances, which will be described in Section II-D. To avoid this side effect, we introduce a new potential function p in the distance regularization termR_p. This new potential function is aimed to maintain the signed distance property $|\nabla_\phi|=1$ only in a vicinity of the zero level set, while keeping the LSF as a constant, with $|\nabla_\phi|=0$, at locations far away from the zero level set. To maintain such a profile of the LSF, the potential function $p(s)$ must have minimum points $s=0$ at and $s=1$. Such a potential is a double-well potential as it has two minimum points (wells). Using this double-well potential $p=p_2$ not only avoids the side effect that occurs in the case of $p=p_1$, but also offers some appealing theoretical and numerical properties of the level set evolution.

1.5 3

The advantage of the method of initialization in this paper is that it allows the use of more general and efficient initialization of the level set function, without requiring a signed distance function as the initial level set function. The paper states that the initial level set function can be any function that is close to a signed distance function near the zero level set, and it can be defined on a relatively coarse grid and then interpolated to a finer grid. This means that the initialization process is more flexible and less computationally expensive than conventional level set methods that require reinitialization. The paper also provides some examples of different initialization methods, such as using a binary mask, a smoothed mask, or a Gaussian kernel.

The initialization in this paper is the process of defining an initial level set function that represents the initial contour or region of interest for image segmentation. The paper proposes a new variational level set formulation that does not require the initial level set function to be a signed distance function, as long as it is close to a signed distance function near the zero level set. This means that the initial level set function can have any shape or topology, and it can be defined on a coarse grid and then interpolated to a finer grid. The paper also suggests some methods for generating the initial level set function, such as using a binary mask, a smoothed mask, or a Gaussian kernel. The initialization process is important because it affects the speed and accuracy of the level set evolution. A good initialization can reduce the number of iterations and avoid local minimal.

1.6 Question 3

Histogram of Oriented Gradients (HOG) is a feature extraction technique commonly used in computer vision and image processing tasks, especially in object detection and recognition. It was first introduced by Navneet Dalal and Bill Triggs in 2005.

The goal of HOG is to represent the local gradient information in an image. It captures the distribution of gradients or edge directions within a localized region of the image. The underlying assumption is that the shape of an object can be well described by the distribution of intensity gradients or edge directions in its vicinity.

Here's a step-by-step explanation of how HOG extracts features:

- Preprocessing: The input image is preprocessed to enhance the features. This can involve operations like converting the image to grayscale, normalizing the image, and applying local contrast normalization.
- Gradient Computation: The image is divided into small regions called cells. For each pixel in a cell, the gradient magnitude and orientation are computed. This is typically done using techniques like the Sobel operator, which calculates the image gradients in the x and y directions.
- Orientation Binning: The gradient orientations are quantized into a fixed number of bins or angular ranges (e.g., 9 bins covering 0 to 180 degrees). Each gradient magnitude is assigned to its nearest bin.
- Histogram Calculation: Within each cell, a histogram of the gradient orientations is constructed by accumulating the gradient magnitudes into their corresponding bins. This histogram summarizes the local distribution of edge directions.
- Block Normalization: To account for variations in illumination and contrast, neighboring cells are grouped together into blocks. The histograms of the cells within a block are concatenated to form a block-level histogram. This block-level histogram is then normalized using techniques like L2-norm or power normalization to make the feature representation more robust to lighting variations.
- Feature Vector: Finally, the normalized block-level histograms are concatenated to form the HOG feature vector, which represents the entire image. This feature vector can be used as input to various machine learning algorithms, such as support vector machines (SVMs), for object detection or recognition tasks.

The HOG feature extraction process focuses on capturing the local shape or edge information rather than the detailed pixel-level appearance. This makes it robust to changes in illumination, background clutter, and small variations in object appearance.

By using the HOG feature representation, it becomes possible to detect and recognize objects based on their distinctive local edge patterns, making it particularly useful in applications like pedestrian detection, face detection, and object tracking.