Notes on BioMedical Image Processing
Mercredi, 20 Juillet 2011 16:24

This section is written in English.

You will find in this section an attempt to gather general notes on biomedical image processing. Biomedical image processing is a relatively recent scientific domain at the interface of multiple scientific specialties including mathematics (geometry, information theory, signal analysis, optimization,...), computer sciences (implementation of algorithmic methods, parallel programming, user interface,...), physics (medical image device reconstruction, deconvolution, fluid mechanic, tracer properties,...), medicine (basic knowledge of anatomy, tissue properties, specific diseases biomarkers), biology (microscopy cell analysis, histology, animal models). We propose here an overview of this large domain spanning many important notions from the pixel to the patient.

Image processing is a very large research domain which includes satellite images analysis, handwriting character recognition, production chain quality assessement, stereoscopic vision for robots, security camera detection, image tagging, compression, etc...These notes specifically focus on the processing and analysis of biomedical images, which is a sub category of image processing. However, when one specific section deals with general image processing concepts, some examples for illustration purposes can be taken in other scientific domains, or even in every day life (like digital photography) when it helps understanding.

Definition of a digital image, spacing, dimensions

A typical 3D image is a function I which associates to a point X of coordinates (x,y,z) a scalar value usually called the intensity I(x,y,z). The set where the intensities live can be{tex} mathbb{N}, mathbb{Z}, mathbb{R}{/tex}...

More generally, a n-D image is a function which associates to a n-uplet (x_1,x_2,...,x_i,...x_n) in Ω a p-uplet (y_1,y_2,...,y_i,...,y_n) belonging to a destination set I. For example, a movie can be considered as a 4D color image where the 4-uplet is (x,y,z,t) and the p-uplet is the RGB values of each point. In the case of a 2D image where the destination set is a scalar ensemble, an image is similar to a mathematical matrix.

A theoretical image has a dense coordinate ensemble, which means that the intensity function is defined for any point in the definition domain. In real life, the intensity function of a digital image is given on a finite subset of points only, which means that we have a discrete representation of the function. Usually, this subset of points is forming a regularly spaced grid, each node of which is called a **pixel** (in 2D) standing for "picture element" or more generally a **voxel** standing for "volume element".

For a given image, an origin point O and direction axis{tex}(vec{x_1}, vec{x_2},...vec{x_i},...vec{x_n}){/tex} are to be defined: for a typical 3D image, they are usually called{tex} vec{x}, vec{y}, vec{z} {/tex}. The distance between two consecutive voxels along the direction{tex} vec{x}, vec{y}, vec{z} {/tex} is called the **spacing** along that particular direction and is usually noted S_X, S_Y or S_Z. The spacing is usually expressed in *mm* and it indicates the link between the digitized image and the physical object it represents in the real world. A microscopy image will typically have a spacing of the order of a micron (10^{-3} mm) in each direction, whereas an image of Manhattan skyline will have a spacing in each direction of the order of the meter (10^{3} mm). The spacing of an image is sometimes referred to as the image **sampling**, or as the image **resolution** (although this concept mostly designates the maximum separation power of an imaging device). The spacing is usually equivalent to the **voxel size** and is usually noted{tex}S_X times S_Ytimes S_{Z} mm^3 {/tex}. When the image spacing is the same for all directions, it is called isotropic. As we saw previously, the intensity within one voxel region is constant. Note that for 2D images, when an image is to be displayed on a screen, printed or scanned, the spacing is also sometimes expressed in *dots per inch (dpi)* which simply indicates the number of pixels in one inch (1 inch = 25.4mm).

Similarly, the number of pixels along the direction{tex} vec{x}, vec{y}, vec{z} {/tex} is referred to as the image **dimension** along that direction and can be noted{tex}N_X times N_Ytimes N_{Z}{/tex}. When one multiplies the image dimension by the image spacing, one gets the physical dimensions of the global region covered by the image, also called the field of view of the image. The field of view of a microscopy image can be a few{tex}mu m^2{/tex} whereas a photography of the skyline can cover several Km^2.

Note: the coordinates $X=(x_1,x_2,...,x_i,...x_n)$ of an image I relatively to the origin O can be given as **indices** (expressed in *number of voxels*), or as **physical coordinates** (expressed in *mm*). We have the relation:

{tex}x_i^{Physical Coordinate}=x_i^{Index}*Spacing(vec{x_i}){/tex}.

When not specified, X will designate from now on the Index coordinate since the intensity function is only partially known on the indices included in the image bounding box. However, we must keep in mind that the index coordinates can not be compared between different images, or even the same image which has been resampled, contrary to the physical coordinates which are absolute coordinates.

{tex}%IMAGE OF MANHATTAN SKYLINE + MICROSCOPY IMAGE{/tex}

Resampling

A biomedical digital image is a discrete representation of a continuous biological object on a spatial grid. If one considers a given image with a given spacing (or sampling) on a given grid, one might want to change the sampling and/or the grid orientation so that to represent the image differently: this is called **resampling** of an image.

When the new grid has a smaller spacing, the operation is called **downsampling**, or subsampling, and it involves loss of information. Downsampling can be useful to turn very large images into smaller ones which are easier to manipulate and store, for display purposes for instance. Because of Nyquist-Shannon sampling theorem, one needs to get rid of high frequencies of the image intensity function before downsampling to avoid aliasing on the resampled image: a low frequency filter (see the filter section) is to be used before downsampling.

On the contrary, when one wants to increase the image sampling rate (reduce the spacing), the operation is called **oversampling**, or upsampling, and it involves interpolation of missing information. Oversampling can be used to avoid pixelization effect when zooming in an image of low resolution.

Resampling is used whenever the voxels grid changes, even if the spacing does not change. For instance, when the grid is shifted or rotated, the image needs to be resampled: this situation happens when the image was acquired with a tilted field of view (e.g. the subject was not aligned with the main axis of the medical scanner). In photography, this can happen when you take a picture of the skyline with the camera not aligned with the horizontal direction.

How to resample an image

We will consider the 2D situation for simplicity, the 3D and nD situation can be infered pretty easily. Let I (frame of reference R, origin O, axis {tex} vec{x}, vec{y}, vec{z} {/tex}) be the image to resample, and I' (frame of reference R', origin O', axis {tex} vec{x'}, vec{y'}, vec{z'} {/tex}) the resampled image. In the general case, resampling involves estimating the intensity value outside of the original grid. In other words, to calculate the values I'(X'), we need to estimate I(X) where X is not a node of the original grid of which intensity value is known. In order to get the resampled image I', we need to populate its grid of intensities of I', which implies knowing the intensity I'(X') for all the voxel positions X' on the grid of I'.

Let P be a physical point in the original image I of coordinates X_0, and let X'_0 be the coordinate of the same physical point P in the frame of reference of the new image I'.

Why we use T^{-1}

The intuitive way - which is also the classical mathematical way in case of transformations in continuous spaces - of resampling an image I into a new image

I' would be to consider every point X in I and to estimate the coordinate X'=T(X)' of the same physical point in I' with T being the forward transformation linking the coordinate X in I and the coordinates X' in I'. However, there is absolutely no guarantee that X' lies on the grid of I', which means that one needs to come up with a strategy to eventually populate the intensity levels on the nodes of the grid of I' (nearest neighbors, computing a continuous map to interpolate the values on the nodes...). Moreover,in case T in not surjective, this forward strategy will lead to holes in the resampled image I' (points of I' which are not images of points of I by T). Similarly, in case T is not injective, a point X' in I' could be the image of several points X in I, with different intensities: the intensities in the resampled image I' would thus depend on the order one processes the voxels of I.These reasons explain why it is necessary to use the backward transformation T^{-1}.

Case where spacing of I' is larger than I

WORK IN PROGRESS...

As we saw, this is situation implies downsampling. If T=Identity, this is pure resampling. If the resampling ratio is an integer, the nodes of I' are included in the the nodes of I. In any case, to avoid aliasing, one needs to apply a low frequency filter prior to resampling.

Practically, let $s_x$be the spacing in the {tex} vec{x}{/tex} direction of I, let s'_x be the spacing in the {tex} vec{x'}{/tex} direction of I'. In the downsampling case we consider, we have s'

x

> s

x

which means that I' has a lower resolution than I.

Let $F=1/s_x$ be the sample rate of I, Let $F'=1/s'_x$ be the sample rate of I', the cut off frequency f $_c$ is F' in this case. The ideal filter is the sinc function 2F'sinc(2F'x). It needs to be used over I prior to downsampling. Alternatively, as an approximation of the sinc function, one can use a Gaussian filter with standard deviation {tex}sigma = F/F' = s'

x

/s

x

{/tex}.

%There are two possible situations: either the frame of reference of the new image I' shares the same frame of reference as the original image I (which translates as $X_0=X'_0$ in physical coordinates), or they are different (which translates as $X'_0=T(X_{0)}$ where T is the transformation between R and R').

{tex}%IMAGE OF THE 2 POSSIBLE SITUATIONS ON AN MRI{/tex}

R=R':

As we said, this situation corresponds to the case where a physical point P of the biomedical object represented in I does not move with respect to R=R'. This is the case for downsampling or upsampling, only the spacing changes but the object represented in the image *does not move*. This is the simplest situation since resampling is the equivalent to interpolating the intensity function I(X). Let X

0

be a position outside of the original grid of I, we do not know directly I(X

0

) but we know the intesity values of grid points which are close to X

0

. As the instensity function is supposed to be a continuous function, we assume we can estimate the value of I(X

0

) as a linear combination of the intensity values of its neighbors. There are different classical ways of estimating the interpolated value.

Nearest neighbor interpolation: If we note X_A the point of the grid which is closest to X_0, we simply have $I(X_0) = I(X_A)$. On a regular grid, X_A can be calculated as $X_A = (\mathrm{floor}(x_0+0.5), \mathrm{floor}(y_0+0.5), \mathrm{floor}(z_0+0.5))$. This is certainly the simplest way to interpolate the intensity function.

Linear interpolation: This is the most commonly used interpolation method since it reprensents a good trade off between calculation time and precision.

Note: the particular downsampling case where the new coarse grid shares nodes with the original grid (e.g. division by 2 of the grid dimention in each direction) is processed similarly providing a low frequency filter is used before resampling.