Documentation: ConvertCTtoMRumap.m

Table of Contents

This function enables the conversion of low dose CT images to PET/MR compatible CT attenuation maps You can also use this program to strip off the pixel data from the dicom header if you use the last section :)

Author: Lalith Kumar Shiyam Sundar, M.Sc. Date: Feb 15th, 2018 Quantitative Imaging and Medical Physics, Medical University of Vienna.

Inputs: umapGen.pathOfLowDoseCT -> path to the low dose CT dicom series umapGen.pathOfDixonSeries -> path to the Dixon series (F, IN, OPP, W, U-map) umapGen.pathToStoreUmap -> where to store the u-maps.

Usage: convertCTtoMRumap(umapGen);

Algorithm behind

The program does the following tasks:

- Reads the low dose CT obtained from a Siemens PET/CT (TPTV)
- Reads the Dixon series from the Siemens Biograph mMR.
- Automatic segmentation of the patient bed from the low-dose CT image.
- Creates a composite dixon image by adding up In-phase, opp-phase, fat and water images.
- Registers the low-dose CT image to the dixon composite image.
- Performs bilinear scaling based on carney (Carney et.al, 2006, Transforming CT images for attenuation correction in PET/CT)
- Pushes the transformed CT-umap to the dixon u-map dicom series, by stripping away the native dixon binary data.

```
function []=convertCTtoMRumap(umapGen)

%-----------------------------------------------------------------------
%
%                              PROGRAM START
%-----------------------------------------------------------------------
%
```

Read the low dose CT map.

```
lowDoseCTfile='Low-Dose-CT.nii';
cd(umapGen.pathOfLowDoseCT); % go to the path containing the low dose CT
convertDicomtoNii(cd,cd);
niftiFile=dir('*.nii'); % remember SPM already converts the pixel values to hounsfield units using the rescale intercept and slope value.
movefile(niftiFile.name,lowDoseCTfile);

% Segmenting the bed from the CT image

OrgCT=spm_read_vols(spm_vol(lowDoseCTfile)); % reads in the Nifti CT images.
figure,imshow3D(OrgCT ); % Displaying the mid axial slice of the CT volume
title('Original CT image with bed');

% 2.) Performing a otsu thresholding.

levelThresh=multithresh(OrgCT,1); % segmenting the foreground and background
SkullThresh=imquantize(OrgCT,levelThresh); %labelling the objects.
foreGround=(SkullThresh==2); % segmenting the bright object - foreground.
[BinaryCT,~]=VolumePartitioner(foreGround); % getting only the head.

for lp=1:size(BinaryCT,3)
    OnlyTheHead(:,:,lp)=imfill(BinaryCT(:,:,lp),'holes'); % filling up the holes in the head, this is done to segment out only the head
end
OnlyTheHead=OnlyTheHead>0; % converting into logical;
figure,imshow3D(OnlyTheHead);title('Binary mask with the holes in the head filled');
tempImg=OnlyTheHead.*OrgCT;

% scaling it to 1000 for applying carney's formula.
scaledImg=tempImg+1000;
OnlyTheHead=scaledImg.*OnlyTheHead; % using the head mask to get only the head from the scaled Image.

% Stealing the NIFTI header and writing the CT volume without the bed as NIFTI file
% We need a NIFTI header for the CT volume without the bed, which we have
% generated. So we are gonna use the NIFTI header from the original CT volume.

NiftiHdrCT=spm_vol(lowDoseCTfile); % read the nifti header of the original CT file using the 'spm_vol' function.
NiftiHdrCT.fname='CTwithoutBed.nii'; % name for the CT volume without the bed - i have hardcoded it, you can change it.
```

```
spm_write_vol(NiftiHdrCT,OnlyTheHead); % writing the CT volume without
 the bed as a NIFTI file.

% Code-patch for setting the origin to the center of the image volume,
 instead of
% manual clicks.

st.vol=spm_vol('CTwithoutBed.nii');
vs=st.vol.mat\eye(4);
vs(1:3,4)=(st.vol.dim+1)/2;
spm_get_space(st.vol.fname,inv(vs));
pathOfNiftiCT=[umapGen.pathOfLowDoseCT,filesep,'CTwithoutBed.nii'];
```

Dixon part

```
cd(umapGen.pathToStoreUmap); mkdir('CT-umap');
CTuMapPath=[umapGen.pathToStoreUmap,filesep,'CT-umap'];
DixonCompositeFile='Dixon-Composite-Image.nii';

% Read the dixon uMap and copy it to the reference folder of the CT
 series
cd(umapGen.pathOfDixonSeries);
dixonUmapFolder=dir('*UMAP*');
dixonUmapPath=[umapGen.pathOfDixonSeries,filesep,dixonUmapFolder.name];
cd(dixonUmapPath);
dixonUmap=readImages(cd);
copyfile(dixonUmapPath,CTuMapPath); % folder where the Dixon dicom
 header information is stored
cd(umapGen.pathOfDixonSeries);
fNames=dir;
fNames=fNames(arrayfun(@(x) x.name(1), fNames) ~= '.'); % read
 volunteer folders
iterVar=0; %intialization
for lp=1:length(fNames)
    if isempty(strfind(fNames(lp).name,'UMAP'))
        iterVar=iterVar+1;
        DixonSeries{iterVar}=fNames(lp).name;
    else
    end
end

% convert the dixon series : Fat, in, opp, water to nifty files.

cd(umapGen.pathOfDixonSeries);
mkdir('Nifti-Dixon');
pathOfNiftiDixon=[umapGen.pathOfDixonSeries,filesep,'Nifti-Dixon'];

for lp=1:length(DixonSeries)
    tempDixonPath=[umapGen.pathOfDixonSeries,filesep,DixonSeries{lp}];
    cd(tempDixonPath);
    convertDicomtoNii(cd,cd);
    niftyFile=dir('*.nii');
    movefile(niftyFile.name,[DixonSeries{lp},'.nii']);
    movefile([DixonSeries{lp},'.nii'],pathOfNiftiDixon);
end

% Preparing a Dixon composite image using the fat, water, inphase,
 outphase maps

cd(pathOfNiftiDixon); % go to the folder containing the individual
 dixon volumes (nifti format)
NiftiDcmFiles=dir;NiftiDcmFiles=NiftiDcmFiles(arrayfun(@(x) x.name(1),
 NiftiDcmFiles) ~= '.'); % reading the files inside the folder.
DixonComposite=zeros(size(spm_read_vols(spm_vol(NiftiDcmFiles(1).name)))); %
 Initialize an empty volume grid.
for lp=1:length(NiftiDcmFiles)
    DixonComposite=DixonComposite
+spm_read_vols(spm_vol(NiftiDcmFiles(lp).name));
end
NiftiDixonHdr=spm_vol(NiftiDcmFiles(1).name);
NiftiDixonHdr.fname='Dixon-Composite-Image.nii';
spm_write_vol(NiftiDixonHdr,DixonComposite);
```

Perform co-registration between the dixon and CT without the bed -> Transforming from PET/CT to PET/MR

```
CoregInputs.SourceImgPath=pathOfNiftiCT;
CoregInputs.RefImgPath=[pathOfNiftiDixon,filesep,DixonCompositeFile];
CoregInputs.MaskImgPath={''}; % No mask is being rotated here.
CoregInputs.Prefix='Coreg_'; % the coregistered image will have a
 prefix called 'Coreg_'.
CoregInputs.Interp=1; % '1' indicates tri-linear interpolation
Coregistration_job(CoregInputs);
cd(umapGen.pathOfLowDoseCT);
CTimgToScale=spm_read_vols(spm_vol('Coreg_CTwithoutBed.nii'));
```

Perform bilinear scaling and push the CT-Umap to the dixon header (reference CT u-Map)

```
cd(umapGen.pathOfLowDoseCT)
cd ..
tempPath=cd;
cd(umapGen.pathOfLowDoseCT)
movefile ('*.nii', tempPath)
CTuMap=carneyBilinearScaling(umapGen.pathOfLowDoseCT,CTimgToScale);
pushDataToDicom(CTuMapPath,CTuMap);
disp('Processing done!')

end
```

Published with MATLAB® R2016a