PVE-corrections

Jan Axelsson 2019-05-15

1 Intro

Partial volume correction can be performed on regions-of-interest (ROI) or pixelwise on the image.

In the methods implemented here, the following has to be known

- the ROIs have to be accurately known, for instance by segmentation from MRI
- image resolution has to be known in x, y, z directions. This can be determined by autocorrelation methods, such as SPM and AFNI 3dFWHMx on noise signals. For dynamic data, the residual image when using principal-component filtering in imlook4d may be used.

For all the partial-volume correction methods implemented here, a partial-volume effect (pve) map is created, which describes how a homogenous activity (of value 1) inside the ROI would be imaged measured with the a known image resolution.

The pve-map is used in multiple ways:

- **RC: recovery coefficient** (RC) of a ROI is determined by simply averaging the pve-map pixels inside a ROI. This can be extended to calculate the cross talk between all ROIs, which is called the geometric-transfer matrix (GTM)
- **GTM: “true” ROI values** and the GTM are related to the measured ROI values through a matrix equation. This equation can be solved for the true ROI values, Rousset 1998. This can be calculated using the pve-map, or by known weight factors, (which can be determined using the pveWeights function).
- **MTC: partial-volume image correction** can be performed by the Multi-target correction (MTC), Erlandsson 2006. This method uses the GTM method to obtain true ROI values. It then treats each ROI as a target and uses a Müller-Gärtner method (MGM) to subract the spill-in of homogenous uptake from all other ROIs. A pve-corrected image is created by repeating this procedure for all ROIs

The following has to be fulfilled to make the above methods work:

- ROIs can be considered to have homogenous uptake. The exception to this is for the MTC, where the target ROI non-homogeneous uptake is calculated. For MTC, this basically means that the spill-in from non-target ROIs are considered similar to that from homogeneous neighbor ROI.
- all pixels in the image are assigned to one ROI (at least in the area of the brain that we are looking at).

The rest of this document describes how to perform partial-volume correction on images with the all-in-one solution. Appendix A contains information on performing specific operations

2 Performing PVE-corrections on images

This example illustrates how to obtain a MTC pve-corrected image matrix using matrices exported from imlook4d. The matrices can be obtained from any method, such as SPM reader, but from imlook4d the matrix of ROI indeces will be called `imlook4d_ROI`, and the pixel values `imlook4d_Cdata`. Also the ROI values are needed, which in imlook4d is called `imlook4d_ROI_data.mean`

```
  % Start with exporting matrices and ROI values from imlook4d
  Export
  ROI_data_to_workspace
  % Get voxelsize in mm.  Can also be specified as vector vox = [ 2 2 2 ];
  vox = voxel_size(imlook4d_current_handles)
  %Convert known resolution in mm fwhm=[ 3.59, 3.40, 4.32], to fwhm in pixels
  fwhm_pixels = pixels( [ 3.59, 3.40, 4.32], vox);
  % Convert from fwhm to sigma
  sigma_pixels = fwhm_pixels / 2.35;

 % Calculate PVE-corrected using MTC
 %(MGM on all ROIs with GMT-corrected true ROI values)
  [C, P, TACT] = pveCorrection( 'MTC',imlook4d_Cdata, imlook4d_ROI, sigma_pixels);
  % Display corrected matrix
  imlook4d(C);
```

The GMT-corrected “true” ROI values can be viewed in TACT matrix, where columns are ROIs, and rows are values for each frame The pve-map can be displayed using imlook4d(P), where the pve-maps are shown for each ROI (frame slider in imlook4d to change ROI).

3 References

Rousset OG, Ma Y, Evans AC.,
Correction for partial volume effects in PET: principle and validation,
J Nucl Med. 1998 May;39(5):904-11.

Müller-Gärtner HW, Links JM, Prince JL, Bryan RN, McVeigh E, Leal JP, Davatzikos C, Frost JJ.,
Measurement of radiotracer concentration in brain gray matter using positron emission tomography: MRI-based correction for partial volume effects,
J Cereb Blood Flow Metab. 1992 Jul;12(4):571-83.

Erlandsson K, Wong A T, van Heertum R, Mann J J and Parsey R V 2006
An improved method for voxel-based partial volume correction in PET and SPECT,
NeuroImage **31** T84

Appendix A

1 Introduction

This appendix describes the use of the underlying functions to the pveCorrection function. These can be used standalone to extract different data.

2 Performing PVE-corrections on images

The below functions has been wrapped in a single function called pveCorrection. This uses combinations of the functions described below (read on for details, read only this section to get started).

```
% Erlandsson 2006
% Calculate PVE-corrected image ROI, by ROI, using Multi-Target Correction
(MTC).
% This is MGM on all ROIs with true ROI values calculated using Geometric
Transfer Matrix, GTM)
[C, P, TACT] = pveCorrection( 'MTC',imlook4d_Cdata, imlook4d_ROI, sigma_pixels);
% Calculate PVE-corrected using MGM-percentile (MGM on all ROIs with ROI values
from 90% percentile)
[C, P, TACT] = pveCorrection( 'MGM-percentile',imlook4d_Cdata, imlook4d_ROI,
sigma_pixels, 90);
```

3 Performing PVE-corrections on ROI data

PVE-correction of ROI data can be done in many ways, with the assumption that all pixels within a ROI have the same constant value. Knowing the true mean value for each ROI, is crucial for these PVE-correction method to work. The problem is that all ROI mean values are incorrect, since they are averaged over pixels that are affected by partial-volume effects.

The true ROI value can be measured in the central parts of a large region (percentile method), or calculated by finding the recovery fraction from the region influenced by spill-out (Geometric Transfer Matrix, GTM, method).

The partial-volume effect map (pve-map) which is created by a convolution with a Gaussian 3-dimensional point spread function (PSF) is needed.

4 Create pve-map

The spill-out from all ROIs is needed to perform PVE corrections. Such a map can be created by the function pveMap. The output P can be used for both image PVE correction and ROI PVE correction. If only ROI PVE corrections should be used, then the preferred method is to use pveWeights (Correct using GTM with W from pveWeights).

```
function P = pveMap( ROIs, PSF)
%
% This function calculates a partial-volume-map for ROIs
% Assumptions are:
% - neighboring regions can be treated as homogenous
% - scanner resolutions in x,y,z directions are.
%
% INPUT:
%   ROIs    3D matrix with ROI indeces
%   PSF     [sigmaX, sigmaY, sigmaZ] in pixels.
%           Convert from FWHM: sigma = FWHM / 2.35
%           Convert from mm to pixels: divide by sigma(pixels) = sigma(mm) /
pixelsize(mm)
%
% OUTPUT:
%   P       PVE map with ROI in 4th dimension
%
% Example:
  % Get voxelsize in mm.  Can also be specified as vector vox = [ 2 2 2 ];
  vox = voxel_size(imlook4d_current_handles);
   % Convert known resolution in mm fwhm=[ 3.59, 3.40, 4.32], to fwhm in pixels
  fwhm_pixels = pixels( [ 3.59, 3.40, 4.32], vox);
  % Convert from fwhm to sigma
  sigma_pixels = fwhm_pixels / 2.35;
  % Create pve-map
  P = pveMap( imlook4d_ROI, sigma_pixels);
```

Great care has been taking in analyzing the matlab convolution operation. It turns out that the length of the convoluted image dimensions have to be even; otherwise a shift occurs in the images. See matlab code of pveMap.

5 Obtaining the ROI values

Knowing the true mean values for each ROI, is crucial for the method to work. The problem is that all ROIs in the image are wrong, since they are affected by partial-volume effects.

5.1 Use mean values

This is simply done by getting the time-activity curve for the ROIs. Smaller regions will not be correct.

```
function measTACT = tactFromMatrix(M,ROIs);
% INPUTS
% - M     3D or 4D matrix
% - ROIs 3D matrix with ROI indeces
%
% OUTPUTS
% - TACT   Time-activity curve extracted by method MTC or MGM-percentil
```

5.2 Use highest pixels

Using the mean of the highest 10% or 20% of the pixels within the ROI, gives central voxels for high-uptake regions, and thus a value closer to the correct uptake. This may still work for larger low-uptake regions, also when neighboring a high-uptake region, because the fraction of pixels suffering from spill in is small. This method may fail for small low-uptake regions close to a high uptake region. Therefore, using mean values for most low-uptake background ROIs is used except for using this highest-pixel method for the isolated ROIs with high uptake.

The highest-pixel function:

```
function TACT= percentileTACT( ROIs, M, p)
%
% This function gives a time-activity curve using the mean of  the highest
% pixels (as defined by fraction parameter)
%
% INPUT:
%   ROIs        3D matrix with ROI indeces
%   M           3D or 4D matrix
%   p           Percentile to use for mean calculations. 90 is the highest 10 %
of pixel values
%
% OUTPUT:
%   TACT        Time activity matrix [frame, ROI]
%
% Example:
  % First, Export from imlook4d.
  TACT = percentileTACT( imlook4d_ROI, imlook4d_Cdata, 90)
```

5.3 Correct using GTM with P from pveMap

The Geometric Transfer Matrix (GTM) method (Rousset, 1998) calculates true mean uptake values by solving the cross-talk due to partial-volume effects between regions. This requires known ROIs

```
function trueTACT = gtm( measTACT, ROIs, P)
%
% This function calculate true uptake values using the geometric transfer
% matrix (GTM) method, by Rousset (1998).
%
%
% INPUT:
%   measTACT measured ROI values [frame, roi].  1D if static scan.  2D if
dynamic scan.
%   ROIs    3D matrix with ROI indeces
%   P       Recovery map, pixel by pixle.  Create by pveMap command
%
% OUTPUT:
%   trueTACT   PVE-corrected true TACT values
%
% Example:
  % Export, and ROI_data_to_workspace from imlook4d.
  %
  % Get voxelsize in mm.  Can also be specified as vector vox = [ 2 2 2 ];
  vox = voxel_size(imlook4d_current_handles)
  %Convert known resolution in mm fwhm=[ 3.59, 3.40, 4.32], to fwhm in pixels
  fwhm_pixels = pixels( [ 3.59, 3.40, 4.32], vox);
  % Convert from fwhm to sigma
  sigma_pixels = fwhm_pixels / 2.35;
  % Create pve-map
  P = pveMap( imlook4d_ROI, sigma_pixels);
  % Get pve-corrected ROI values
  TACT = gtm( imlook4d_ROI_data.mean, imlook4d_ROI, P);
```

5.4 Correct using GTM with W from pveWeights

For PVE correction of ROIs (not images), the output from pveMap is unnecessary big, especially for large number of ROIs. Therefore, it is better to calculate only the weight factors for the GTM model, and apply them.

Calculating weights:

```
function W = pveWeights( ROIs, PSF)
%
% This function calculates  partial-volume weights for cross talk between all
ROIs
% using Geometric Transfer Matrix (GTM) method
% (Rousset OG, Ma Y, Evans AC, J Nucl Med. 1998 May;39(5):904-11).
% Assumptions are:
% - neighboring regions can be treated as homogenous
% - scanner resolutions in x,y,z directions are.
%
% INPUT:
%   ROIs    3D matrix with ROI indeces
%   PSF     [sigmaX, sigmaY, sigmaZ] in pixels.
%           Convert from FWHM: sigma = FWHM / 2.35
%           Convert from mm to pixels: divide by sigma(pixels) = sigma(mm) /
pixelsize(mm)
%
% OUTPUT:
%   W       PVE weights between ROIs (2D matrix)
%
% Example:
%   % Export from imlook4d.
%   % Get voxelsize in mm.  Can also be specified as vector vox = [ 2 2 2 ];
   vox = voxel_size(imlook4d_current_handles);
    % Convert known resolution in mm fwhm=[ 3.59, 3.40, 4.32], to fwhm in pixels
   fwhm_pixels = pixels( [ 3.59, 3.40, 4.32], vox);
   % Convert from fwhm to sigma
   sigma_pixels = fwhm_pixels / 2.35;
   % Create pve-weights
   W = pveWeights( imlook4d_ROI, sigma_pixels);
```

The ROI values can then be PVE-corrected by the following operations :

```
% Dynamic data
[tact, NPixels, stdev, maxActivity, roisToCalculate ] = ...
   generateTACT(imlook4d_current_handles,imlook4d_ROI);
   numberOfFrames = size(tact,2);
   for i = 1:numberOfFrames
        pvc_tact(:,i) = (pveFactors' \ tact(:,i) );
   end

% Static data
ExportROIs; % Saves pveFactors to workspace
imlook4d_ROI_data.pve = (pveFactors' \ imlook4d_ROI_data.mean')';
```

6 PVE-correct an image

The pve correction requires known time-activity data for each ROI, the ROI matrix, the PET matrix, and the point-spread function resolution.

This was implemented as the function

```
function [C, P] = pve( TACT, ROIs, M, P)
```

```
%
% This function calculates a partial-volume-corrected image.
% Assumptions are:
% - neighboring regions can be treated as homogenous
% - scanner resolutions in x,y,z directions are.
%
% INPUT:
%   TACT    true ROI values [frame, roi].  1D if static scan.  2D if dynamic
scan.
%   ROIs    3D matrix with ROI indeces
%   M       3D or 4D matrix
%   P       Recovery map, pixel by pixle.  Create by pveMap command
%
% OUTPUT:
%   C       PVE-corrected matrix
%   P       PVE map with ROI in 4th dimension
%
% Example:
  % Export, and ROI_data_to_workspace from imlook4d.

  % Get voxelsize in mm.  Can also be specified as vector vox = [ 2 2 2 ];
  vox = voxel_size(imlook4d_current_handles);
  % Convert known resolution in mm fwhm=[ 3.59, 3.40, 4.32], to fwhm in pixels
  fwhm_pixels = pixels( [ 3.59, 3.40, 4.32], vox);
  % Convert from fwhm to sigma
  sigma_pixels = fwhm_pixels / 2.35;
  % Create pve-map
  P = pveMap( imlook4d_ROI, sigma_pixels);
  % Create pve-corrected image
  C=pve( imlook4d_ROI_data.mean, imlook4d_ROI, imlook4d_Cdata, P); imlook4d(C);
```

This method is quite slow due to the convolution operation in MATLAB, but there may be optimized ways to solve this (http://www.mathworks.com/matlabcentral/newsreader/view_thread/309585).