Rigid Body Image Realignment in image space vs. k-space

Stefan Skare
Karolinska University Hospital
Stockholm, Sweden

It is natural to think of rigid body motion correction between similar images (2D) or image volumes (3D) as a task performed in the image domain. To find the amount of motion in units of voxels (or mm) and degrees, the image to be registered (i.e. corrected) is compared against some reference image (often the first one in the time series). The best estimate (guess) of the motion that has occurred corresponds to the set of translations and rotations that give the lowest sum-of-squares difference (or some other metric) between the reference image (I_{ref}) and the image to be registered (I_{moved}):

$$\arg\min\left[SoS(\mathbf{x})\right] = \arg\min\left[\sum_{\text{all voxels}}\left(I_{ref} - \mathrm{I}_{moved}(\mathbf{x})\right)^2\right] \qquad [1]$$

where $\mathbf{x}_{2D} = \left[T_x, T_y, R_z\right]$ for the 2D case (two translations and one rotation in the 2D plane) and $\mathbf{x}_{3D} = \left[T_x, T_y, T_z, R_x, R_y, R_z\right]$ (full rigid body motion: 3 translations and 3 rotations) for the 3D (image volume) case.

We need to use some search algorithm (engine) that iteratively tries different values for **x** until the images look as similar as possible. For the 2D case, **x** contains three parameters, and the search space will therefore be a 3-dimensional parameter space, in which we want to find the best solution. For 3D motion correction, this search space becomes instead 6-dimensional.

The estimation of these motion parameters - and the correction of the data using these motion parameters - may also be performed in k-space. In some situations, performing motion estimation in k-space is the only choice, such as when a navigator is added to a pulse sequence for the purpose of motion detection, but where the navigator data does not cover enough of k-space to allow for a Fourier transformation to an image that could be used in Eq. [1]. Examples of such navigators include e.g. the cloverleaf (1), orbital (2), and spherical (3) navigators. For PROPELLER data, motion correction has been performed both in k-space (4) and in the image domain (5).

To understand how motion detection and correction can be performed in k-space, we first need to understand the Fourier shift theorem. This theorem states that a shift of data in one domain (for motion we have these shifts, or translations, in the image domain) corresponds to an added linear phase ramp in the Fourier transformed domain (i.e. here in k-space). This is illustrated in Fig. 1 below. A shift (translation) of the image data of e.g. 1 pixel to the right in the horizontal direction corresponds to an added phase ramp in k-space of $-\pi \rightarrow \pi$, i.e. where the left-most column of k-space has a phase change of $-\pi$ (or -180 deg.) and the right-most column of k-space has a phase change of $+\pi$ (or +180 deg.). For a translation in the image domain of e.g. 4 pixels vertically and -3 pixels horizontally, the corresponding k-space will first have a vertical phase ramp added of $\underbrace{-4\pi}_{\text{top row}} \rightarrow \underbrace{4\pi}_{\text{bottom row}}$, but also a horizontal phase ramp of $\underbrace{3\pi}_{\text{leftmost column}} \rightarrow \underbrace{-3\pi}_{\text{rightmost column}}$ added to it, summing up to a combined phase ramp along a near-diagonal direction in k-space. Hence, from the k-space data we will use for motion detection (estimation), we need to be able to determine this slope and direction of the motion induced phase ramp. See the Matlab example at the end of this document.

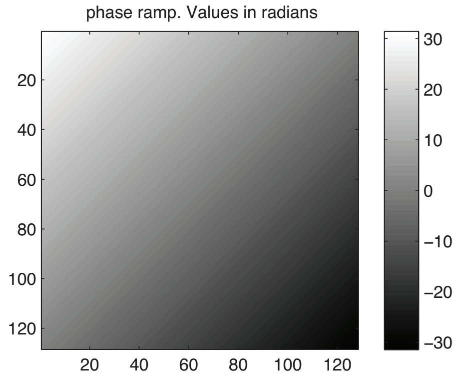
A rotational motion is the same in k-space and the image domain, i.e. if we detect α degrees of rotational motion in k-space, it will be the same in the image domain. Using something else than square image pixels and square image FOV (corresponds to square pixels in k-space), one have to keep the voxel size in mind when rotating.

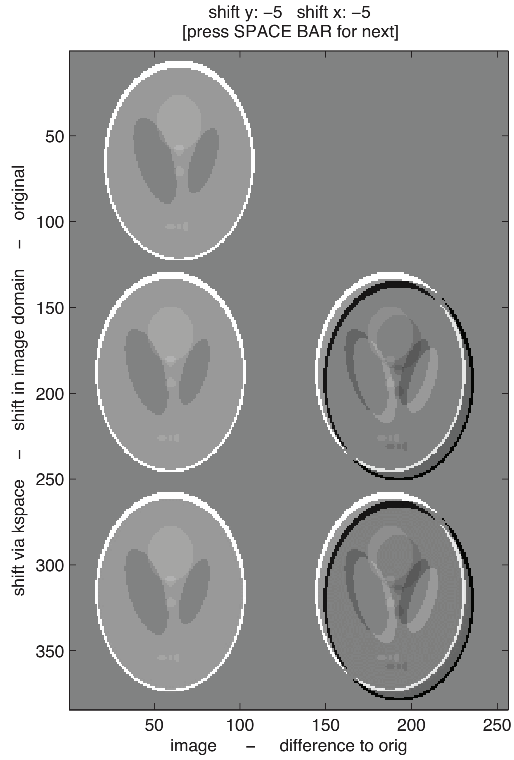
Searching for the optimal **x** in k-space can be done in two stages, unlike when estimating the motion in the image domain (where all 3 (2D) or 6 (3D) parameters need to be estimated together). By taking the magnitude of k-space, all phase is removed from the k-space data and hence also phase ramps (all translation effects in the image domain). Hence, on a magnitude (2D) k-space, we only estimate one rotation parameter, i.e. we are performing a 1-dimensional search for the best rotation. Once this rotation is found, the translation parameters can be estimated using the complex rotation corrected k-space. This technique is well described in (4,6). Moreover, Refs (7,8) explains how rotations can be made by using successive shear operations.

Image translation demonstration using Matlab (or Octave)

```
res = 128;
I = phantom(res); % create a Shepp-Logan phantom image of size res x res in variable 'I'

close all; % close all figures (if any)

k = fftshift(fft2(fftshift(I))); % Fourier transform of the original image to k-space

for shift_x = -5:5:5
   % shifts (translations) from -5 to +5 pixels (in steps of 5)
   % in the horizontal direction

   for shift_y = -5:5:5
      % shifts from -5 to +5 pixels (in steps of 5)
      % in the vertical direction

      % shift the image in the image domain. 'circshift' only supports integer shifts
      Ish = circshift(I, [shift_y shift_x]);

      % create a horizontal 1D ramp of values between:
      % -shift_x*pi (-shift_x*180deg) --> +shift_x*pi (+shift_x*180deg)
      phaseramp_onerow = linspace(-pi, +pi, size(k,2))  * shift_x;

      % create a vertical 1D ramp of values
      % between -shift_y*pi to +shift_y*pi
      % Note the "'" sign making this a column vector
      phaseramp_onecol = linspace(-pi, +pi,  size(k,1))' * shift_y;

      % create the combined phaseramp in k-space
      % - first term is a copy of the 1D row ramp to all rows
      % - second term is a copy of the 1D column ramp to all columns
      phaseramp2D = repmat(phaseramp_onerow, [size(k,1) 1]) + ...
         repmat(phaseramp_onecol, [1 size(k,2)]);

      % add this phaseramp to the original kspace
      k_withphaseramp = k .* exp(-i * phaseramp2D);

      % Fourier transformt the new kspace to the image domain
      Ish_doneinkspace = abs(ifftshift(ifft2(ifftshift(k_withphaseramp))));


      % show the data for each shift combination
      figure(1); colormap gray
      subplot(121);
      imagesc(phaseramp2D,10*pi*[-1 1]);
      axis image; title('phase ramp. Values in radians'); colorbar; drawnow

      subplot(122);
      imagesc([I  zeros(size(I)); Ish (Ish-I); Ish_doneinkspace (Ish_doneinkspace-I)],[-1 1]);
      axis image;
      if exist('truesize','file')
         truesize; % one pixel in the image = one pixel on the screen
```

```
        end
        ylabel('shift via kspace    -    shift in image domain    -    original')
        xlabel('image      -     difference to orig');
        title(sprintf('shift y: %g   shift x: %g\n[press SPACE BAR for next]',shift_y, shift_x));

        % wait for key press.
        % In Octave you need to click on the Command Window, then SpaceBar, each time
        pause;
    end
end
```





References

1. van der Kouwe AJ, Benner T, Dale AM. Real-time rigid body motion correction and shimming using cloverleaf navigators. Magn Reson Med 2006;56(5):1019-1032.
2. Fu ZW, Wang Y, Grimm RC, Rossman PJ, Felmlee JP, Riederer SJ, Ehman RL. Orbital navigator echoes for motion measurements in magnetic resonance imaging. Magn Reson Med 1995;34(5):746-753.
3. Welch EB, Manduca A, Grimm RC, Ward HA, Jack CR, Jr. Spherical navigator echoes for full 3D rigid body motion measurement in MRI. Magn Reson Med 2002;47(1):32-41.
4. Pipe JG. Motion correction with PROPELLER MRI: application to head motion and free-breathing cardiac imaging. Magn Reson Med 1999;42(5):963-969.
5. Skare S. Motion Correction of a new T1-w Propeller Sequence (SE-prop). 2012; Melbourne, Australia. p 2458.
6. Maas LC, Frederick BD, Renshaw PF. Decoupled automated rotational and translational registration for functional MRI time series data: the DART registration algorithm. Magn Reson Med 1997;37(1):131-139.
7. Cox RW, Jesmanowicz A. Real-time 3D image registration for functional MRI. Magn Reson Med 1999;42(6):1014-1018.
8. Eddy WF, Fitzgerald M, Noll DC. Improved image registration by using Fourier interpolation. Magn Reson Med 1996;36(6):923-931.