Parallelized reconstruction of spiral Fourier velocity encoding MRI data on multicore processors

Rosana Ribeiro Lima (rosanariblim@gmail.com)
João Luiz Azevedo de Carvalho (joaoluiz@pgea.unb.br)

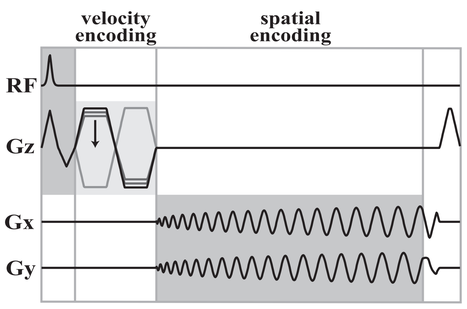
Medical Imaging and Signal Processing Group
Department of Electrical Engineering
University of Brasília, Brasília–DF, Brazil

Introduction

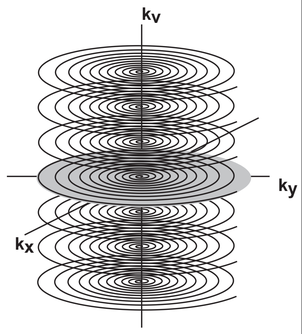
- Flow MRI can be used to assess valvular disease [1] and carotid wall shear stress [2]
- Phase contrast: fast, but has issues with partial voluming
- Fourier velocity encoding: robust to partial voluming, but slow
- Scan time in FVE can be reduced using spiral trajectories in $\mathbf{k_x}$-$\mathbf{k_y}$ (spatial encoding) [1]
- Spiral FVE: long reconstruction time, due to its high dimensionality and non-Cartesian sampling
- Reconstruction time can be reduced using parallel computing

Spiral FVE

- Multidimensional data: $\mathbf{m(x, y, z, v, t)}$



Spiral FVE pulse sequence [1]



Spiral FVE k-space trajectory: a stack-of-spirals in $\mathbf{k_x}$-$\mathbf{k_y}$-$\mathbf{k_v}$ [1]

- Reconstruction:
  - NUFFT [3] along $\mathbf{k_x}$-$\mathbf{k_y}$
  - FFT along $\mathbf{k_v}$

Parallelized Reconstruction in Matlab

- Use a "parfor" loop instead of a "for" loop
- Variable types allowed: temporary, broadcast, loop, sliced, and reduction [4]
- Specific implementation
- Bias time for initialization: works better for large scale processing

incorrect

```
X = zeros(3,12);
parfor m = 1:3
    for n = 1:12
        X(m,n) = m + n;
    end
    disp(X(m,1))
end
```

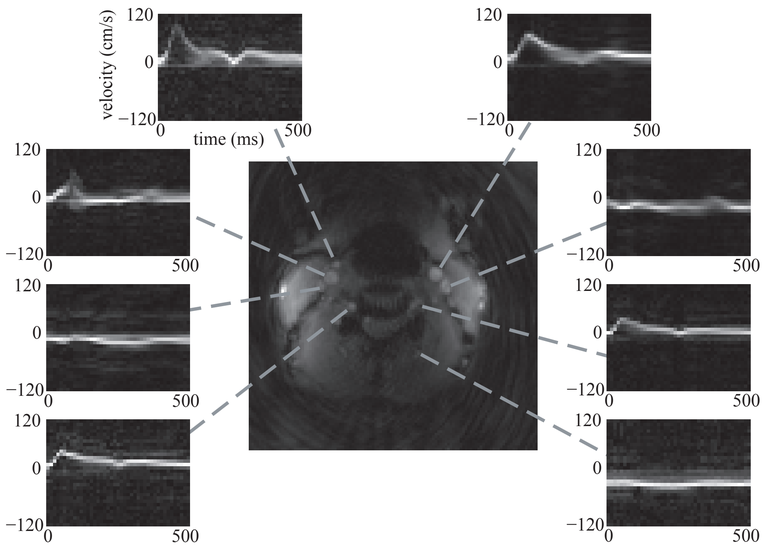
correct

```
X = zeros(3,12);
parfor m = 1:3
    p = zeros(1,12);
    for n = 1:12
        p(n) = m + n;
    end
    disp(p(1))
    X(m,:) = p;
end
```

Example of incorrect and correct usages of parfor, performing the same function

Reconstructed Data

- Multi-slice CINE spiral FVE scans
- Spatial resolution: $1.4 \times 1.4 \times 5$ mm^3
- 8 variable-density spiral readouts (4 ms each)
- Velocity resolution: 5 cm/s (32 velocity encodes)
- Temporal resolution: 12 ms (43 cardiac phases)
- 5 axial slices
- Scan time: 2.4 min/slice (256 hbs @ 105 bpm)

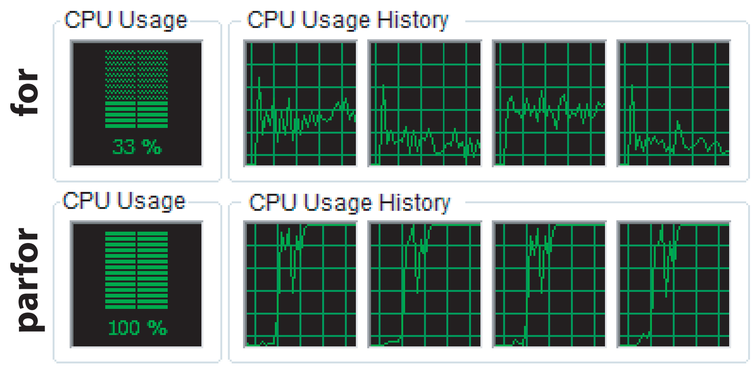


Time-velocity distributions from an axial slice prescribed at the neck of a healthy volunteer

- Reconstruction time measured for:
  - Matlab R2008a and R2011a
  - Quad-core and dual-core processors
  - 2D (**x**,**y**), 4D (**x**,**y**,**v**,**t**), and 5D (**x**,**y**,**z**,**v**,**t**) data

Parfor Loops

- Parfor restrictions → changes in the algorithm
  - Externalization of parfor loop
  - Change of phase and slice loops order (turns slice into a loop variable)
  - Redefinition of $\mathbf{m(x, y, k_v, c)}$ variable inside parfor (makes it a temporary variable)
- Allocation of code in the outer loops whenever possible



CPU usage for a quad-core processor while running a sequential implementation of the reconstruction algorithm (for), and a parallelized implementation (parfor)

Results and Discussion

Reconstruction times (in seconds) for the sequential algorithm ("for" loop) and the parallelized approach ("parfor" loop)

	MATLAB R2008a					
	dual-core processor			quad-core processor		
	for	parfor	reduction	for	parfor	reduction
$m(x,y)$	0.8	1.1	no	0.3	0.9	no
$m(x,y,v,t)$	171.2	107.8	37%	87.5	49.8	43%
$m(x,y,z,v,t)$				435.7	275.5	37%

	MATLAB R2011a					
	dual-core processor			quad-core processor		
	for	parfor	reduction	for	parfor	reduction
$m(x,y)$	0.7	1.0	no	0.3	0.9	no
$m(x,y,v,t)$	107.1	90.3	16%	65.2	40.1	38%
$m(x,y,z,v,t)$				334.9	218.3	35%

- Parfor was unable to reduce reconstruction time for the small datasets, because of initialization overhead
- Matlab 2011a presented significantly faster reconstruction times
- Speed-up achieved from using parallelized reconstruction in Matlab 2011a was less significant
  - Newer versions provide improved multicore support for many built-in functions
  - Increased CPU usage even within traditional "for" loops

Conclusion

- Parallelized reconstruction is a simple and practical approach for speeding-up MRI reconstruction
- It can be especially useful when dealing with multidimensional data, non-Cartesian sampling, and/or iterative reconstruction (e.g., compressed sensing)

References

[1] Carvalho JLA and Nayak KS. MRM 57:639, 2007
[2] Carvalho JLA et al. MRM 63:1537, 2010
[3] Fessler JA and Sutton BP. IEEE TSP 51:560, 2003
[4] http://www.mathworks.com/help/

Financial Support

- PAEX/CAPES
- PIBIC/UnB/CNPq
- DEG/UnB
- PGEA/ENE/FT/UnB
- Edital MCT/CNPq 014/2010 – Universal