Liver vein segmentation
for contrast-enhanced MR images

László Ruskó, GE Healthcare, Szeged

Imagination at work.

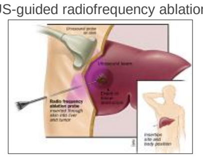
Overview

- Motivations
- Liver segmentation
- Vessel segmentation
- Challenges of vessel segmentation
- Improvement opportunities

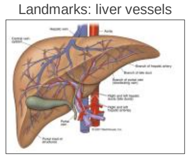
2

Motivation
Ultrasound – MRI image registration

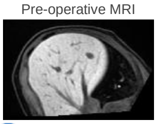
US-guided radiofrequency ablation



Landmarks: liver vessels

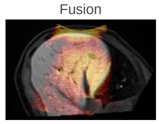


Pre-operative MRI



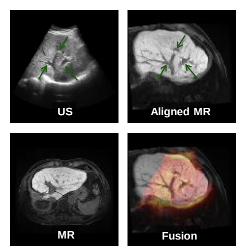
Intra-operative US

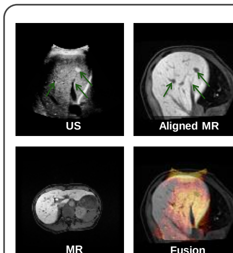
Fusion



3

Examples of automated registration



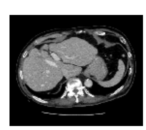


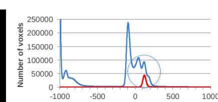
4

Liver segmentation

Liver segmentation
Characteristics of EOB-enhanced MR LAVA

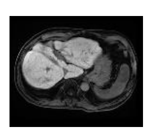
Contrast-enhanced CT

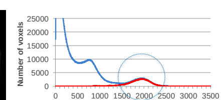




Liver (red) intensity is represented by the last peak of the image histogram (blue), and it is always found in a well defined range.

EOB – MR LAVA

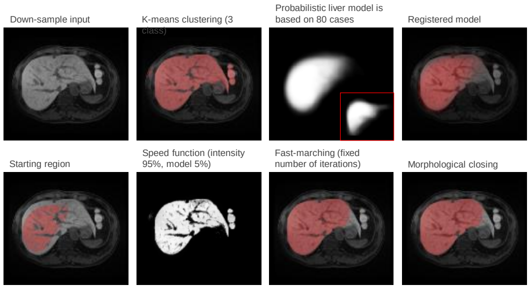




Liver intensity is represented by the last peak of the histogram, and its position can change among cases.

6

Liver segmentation
Model-based approach



Reliable ROI for vein segmentation

7

Model registration – ITK code snippet

```
typedef itk::MeanSquaresImageToImageMetric < ImageType, ImageType > MetricType;
typedef itk::TranslationTransform < double, 3 > TranslateType;
typedef itk::NearestNeighborInterpolateImageFunction< ImageType, double > InterpolatorType;
typedef itk::RegularStepGradientDescentOptimizer OptimizerType;
typedef itk::ImageRegistrationMethod < ImageType, ImageType > RegistrationType;

metric = MetricType::New( );                    // Define image difference metric
trans = TranslateType::New( );                  // Define registration transform
interpolator = InterpolatorType::New( );        // Define interpolator
optimizer = OptimizerType::New( );              // Define optimizer
registration = RegistrationType::New( );        // Define registration method

registration->SetMetric( metric );              // Set metric
registration->SetTransform( trans );            // Set transform
registration->SetInitialTransformParameters( trans->GetParameters() );
registration->SetInterpolator( interpolator );  // Set interpolator
registration->SetOptimizer( optimizer );        // Set optimizer
registration->SetFixedImage( fix_image );       // Set fixed image (clustered input image)
registration->SetMovingImage( mov_image ); // Set moving image (liver model)
registration->SetFixedImageRegion( fix_image->GetLargestPossibleRegion() );

optimizer->SetMaximumStepLength( 10.0 );  // Set optimizer parameters
optimizer->SetMinimumStepLength( 1 );
optimizer->SetNumberOfIterations( 25 );

registration->Update( );                        // Start the registration
                                                // Final transform = registration->GetTransform()
```

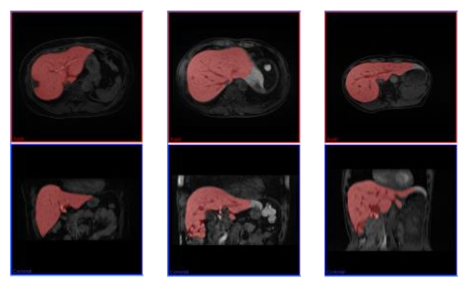
8

Liver segmentation – ITK code snippet

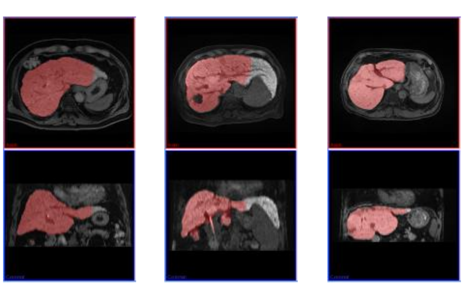
```
typedef itk::FastMarchingImageFilter< ImageType, ImageType > FastMarchingFilterType;
typedef FastMarchingFilterType::NodeContainer NodeContainer;

// Starting region is defined by the core of the registered model
…
NodeContainer::Pointer alives = NodeContainer::New();    // Internal voxels of starting region
NodeContainer::Pointer trials = NodeContainer::New(); // Contour voxels of starting region
…
// Speed of contour propagation is defined based on intensity similarity to starting region and
probabilities in the registered model
…
ImageType::Pointer speed = …;
…
FastMarchingFilterType::Pointer fastMarching = FastMarchingFilterType::New();

fastMarching->SetAlivePoints( alives );          // Set starting region
fastMarching->SetTrialPoints( trials );

fastMarching->SetInput( speed );                 // Set speed function

fastMarching->SetStoppingValue( stop_value );    // Set maximal number of iterations

fastMarching->Update();                          // Start the segmentation
                                                 // Result = fastMarching->GetOutput();
```

9

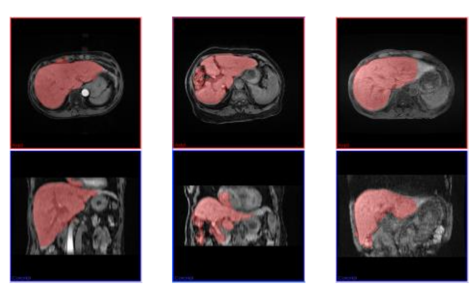
Liver segmentation
Results – normal cases



Liver segmentation
Results – cirrhotic cases



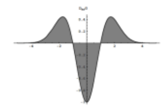
Liver segmentation
Results – low contrast cases



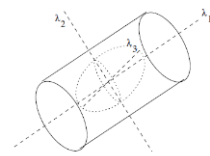
Vessel segmentation

Vessel segmentation
Hessian filtering

$$\frac{\partial}{\partial x}L(\mathbf{x}, s) = s^{\gamma} L(\mathbf{x}) * \frac{\partial}{\partial x}G(\mathbf{x}, s)$$



$$|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$$



$2D$		$3D$			orientation pattern
λ_1	λ_2	λ_1	λ_2	λ_3	
N	N	N	N	N	noisy, no preferred direction
		L	L	H-	plate-like structure (bright)
		L	L	H+	plate-like structure (dark)
L	H-	L	H-	H-	tubular structure (bright)
L	H+	L	H+	H+	tubular structure (dark)
H-	H-	H-	H-	H-	blob-like structure (bright)
H+	H+	H+	H+	H+	blob-like structure (dark)

Vessel segmentation
Frangi's multi-scale vessel enhancement filtering

$$\mathcal{R}_{\mathcal{A}} = \frac{(\text{Largest Cross Section Area})/\pi}{(\text{Largest Axis Semi-length})^2} = \frac{|\lambda_2|}{|\lambda_3|}$$

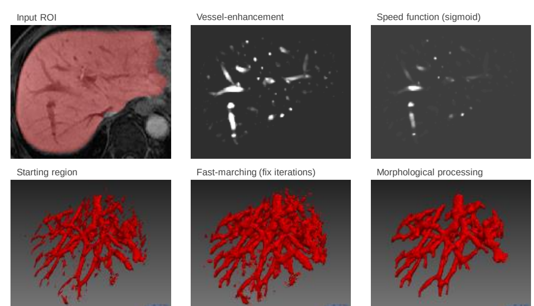
$$\mathcal{R}_{\mathcal{B}} = \frac{\text{Volume}/(4\pi/3)}{(\text{Largest Cross Section Area}/\pi)^{3/2}} = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}$$

$$\mathcal{S} = \|\mathcal{H}\|_F = \sqrt{\sum_{j \leq D} \lambda_j^2}$$

$$\mathcal{V}_o(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0, \\ (1 - \exp\left(-\frac{\mathcal{R}_{\mathcal{A}}^2}{2\alpha^2}\right)) \exp\left(-\frac{\mathcal{R}_{\mathcal{B}}^2}{2\beta^2}\right)(1 - \exp\left(-\frac{\mathcal{S}^2}{2c^2}\right)) & \end{cases}$$

$$\mathcal{V}_o(\gamma) = \max_{s_{min} \leq s \leq s_{max}} \mathcal{V}_o(s, \gamma)$$

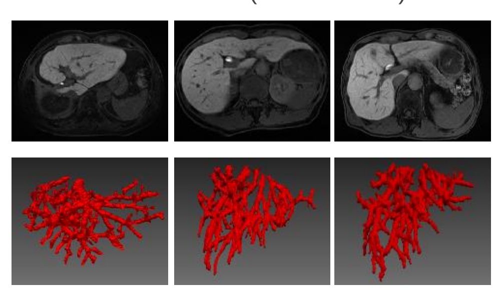
Vessel segmentation
Based on vessel-enhancement



Vessel enhancement – ITK code snippet

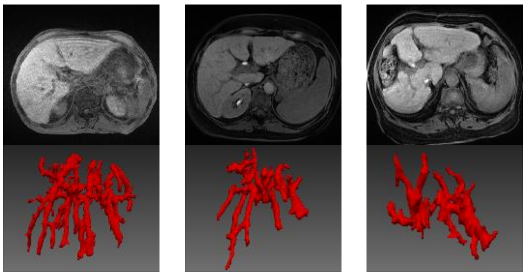
```
// Create Hessian filter
typedef itk::HessianRecursiveGaussianImageFilter< ImageType > HessianFilterType;

HessianFilterType ::Pointer hessianFilter = HessianFilterType ::New();
hessianFilter->SetInput( input );                // Set input  (MR image)
hessianFilter->SetSigma( sigma );                // Set radius of the tubular structure to enhance
hessianFilter->Update();

// Create vesselness filter based on "3D Multi-scale line filter for segmentation and visualization
of curvilinear structures in medical images", Yoshinobu Sato et al.
typedef itk::Hessian3DToVesselnessMeasureImageFilter<float> VesselnessFilterType;

VesselnessFilterType::Pointer vesselnessFilter = VesselnessFilterType ::New();
vesselnessFilter->SetInput( hessianFilter->GetOutput() );
vesselnessFilter->Update();

// Create intensity rescale filter to normalize vesselness values
typedef itk::RescaleIntensityImageFilter< FloatImageType, ImageType > RescaleFilterType;

RescaleFilterType ::Pointer rescale = RescaleFilterType::New();
rescale->SetInput( vesselnessFilter->GetOutput() );
rescale->SetOutputMaximum( 1000 );
rescale->SetOutputMinimum( 0 );
rescale->Update();                               // Result is returned by rescale->GetOutput();
```


Vessel segmentation
Results – Good contrast (40% of cases)



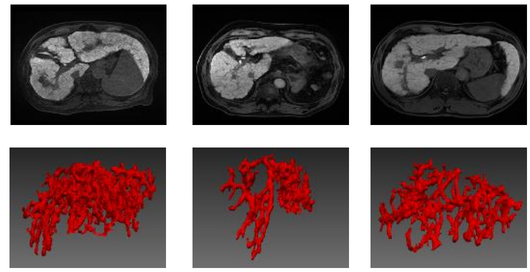
Good result

Vessel segmentation
Results – Low contrast (30% of cases)



Under-segmented result

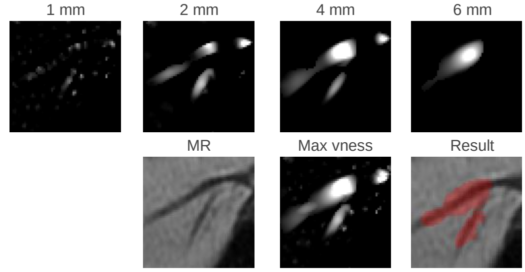
Vessel segmentation
Results – Cirrhotic cases (30% of cases)



Under-segmented result

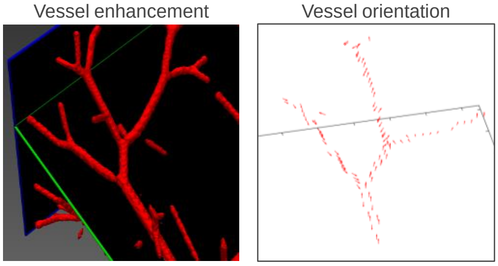
Challenges

Hessian vessel enhancement
Challenge 1: disconnected vessel tree



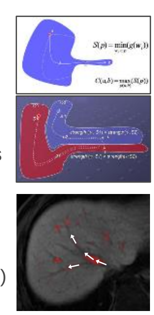
Vesselness can be nearly zero at junctions

Hessian-based vessel enhancement
Idea: Incorporate vessel orientation



Segmentation approach
Fuzzy connectedness*

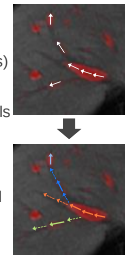
- Theoretical background
  - Graph-based approach based on the similarity of neighboring voxels
  - Computes the connected components finding the optimal routes
- Advantages
  - Extend concept: incorporate local direction vector when similarity is computed (+ intensity and vesselness)
  - Fast and easy to implement



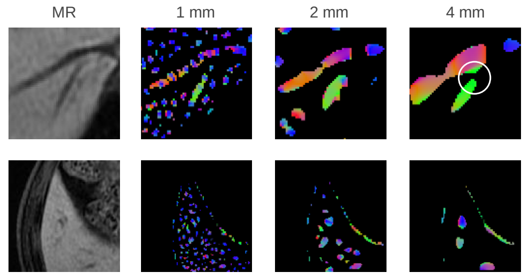
*Fuzzy Connectedness and Image Segmentation, J. K. Udupa, P. K. Saha, Proceedings of the IEEE, VOL. 91, NO. 10, OCTOBER 2003

Proposed method
Combine intensity with vessel direction

- Pre-process
  - Compute vesselness and local vessel directions
  - Identify the most likely vessel voxels (starting points)
- Iterative segmentation
  - Incorporate intensity difference of neighboring voxels
  - Allow **faster contour propagation in local vessel direction** (slow in perpendicular directions)
  - **Propagate local vessel direction** from starting points into less confident voxels of its neighborhood
- Post-process
  - Morphological filtering and connected components

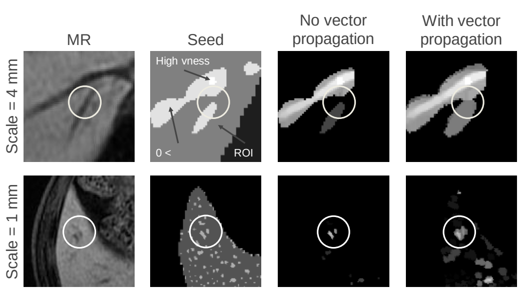


Compute vessel orientation
Preliminary results



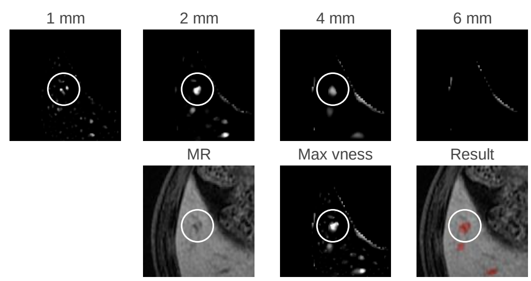
Vectors located at 2 sides of a disjoint branch point in the same direction

Segmentation based on intensity and directions
Preliminary results



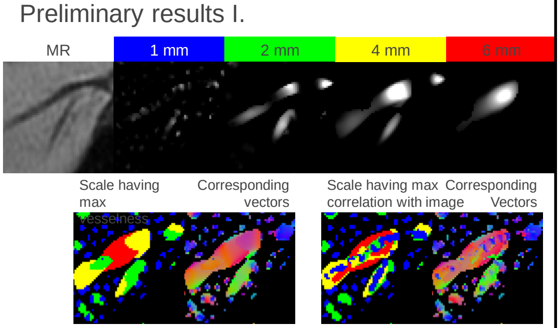
Promising results using one scale only

Multi-scale vessel enhancement
Challenge 2: leak between branches



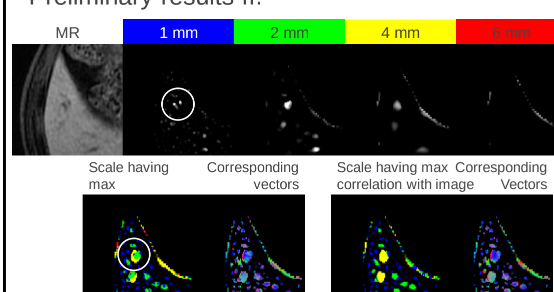
Larger scales blur the small vessels

Scale selection
Preliminary results I.



Difficult to combine the information of different scales

Scale selection
Preliminary results II.

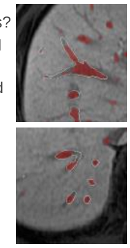


Difficult to combine the information of different scales

Future work and open questions

- Method
  - How to separate confident directions from noise?
  - How to combine the information of different scales?
  - Vector propagation to all neighbors or only in local direction?
  - What is the optimal balance between intensity and direction?
- Validation
  - Quantitative evaluation on synthetic images
  - Qualitative evaluation on real images
- Optimization
  - Reduce running time (1.5m >> 10-15s)
  - Reduce extreme memory usage (24 x input size)

Preliminary results



31

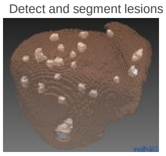
Improvement opportunities

Lesion detection and segmentation

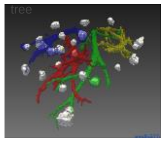
Liver segmentation and morphological closing

Intensity normalization based on liver mode

Detect and segment lesions



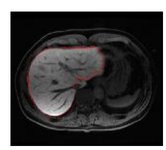
Remove lesions from vessel tree

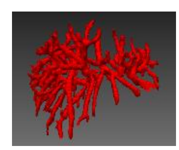


33

Normalize intensity heterogeneity

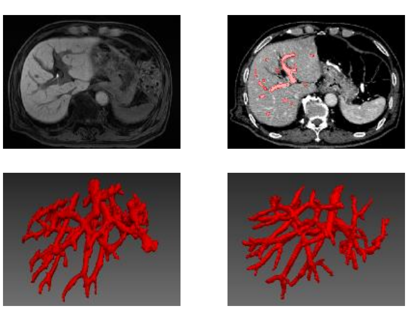
Normalize intensity based on the difference of the local average and the liver mode





34

Adaption to CT modality



35