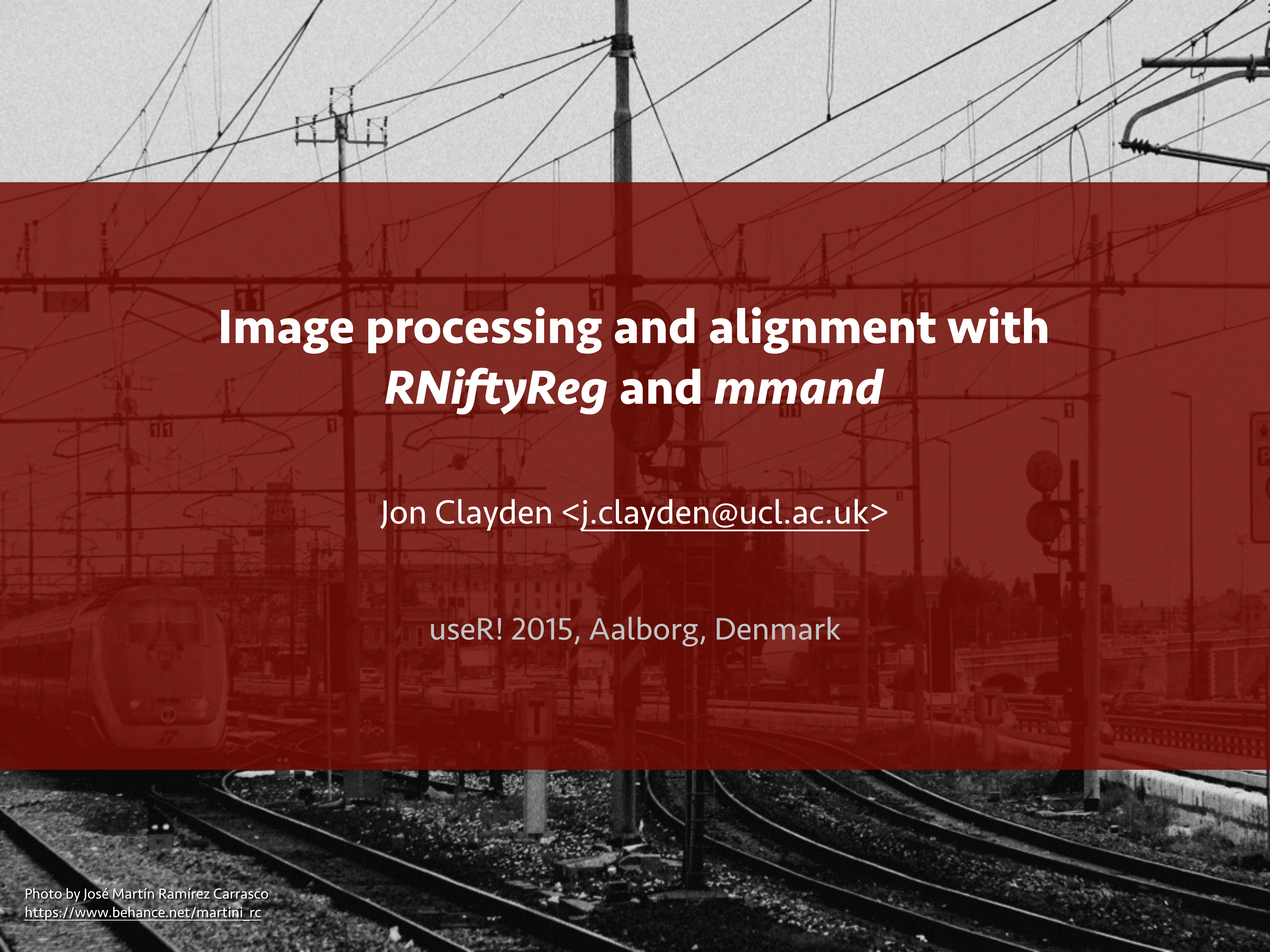


Image processing and alignment with *RNiftyReg* and *mmand*

Jon Clayden <j.clayden@ucl.ac.uk>

useR! 2015, Aalborg, Denmark

Photo by José Martín Ramírez Carrasco
https://www.behance.net/martini_rc

Images

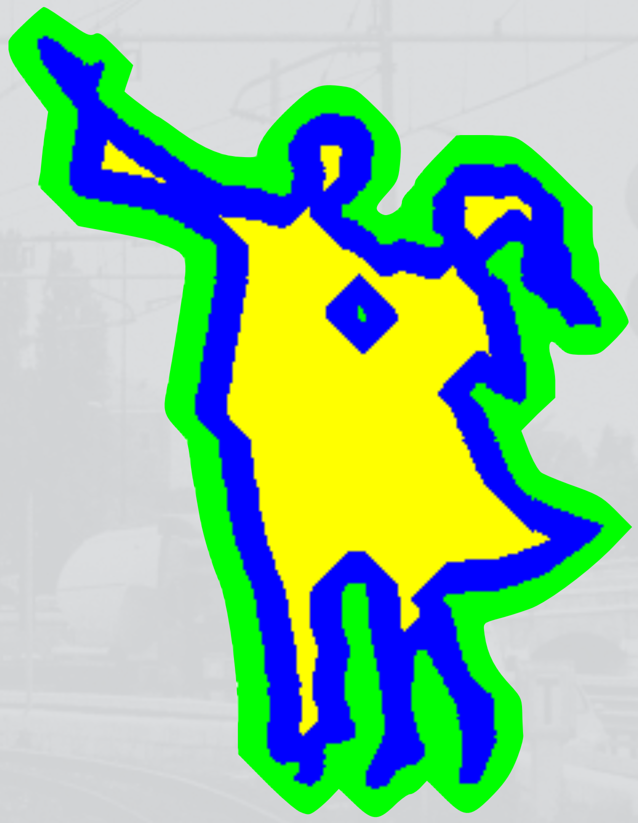
- Produced and analysed across a wide range of disciplines
- Single-channel or multichannel (e.g., RGB)
- Fundamentally numeric arrays of two (or more) dimensions
- Often processed to identify or emphasise features of interest
- Good fit to R's vector-based paradigm

ESA/Rosetta/NAVCAM

Mathematical morphology

- Basis of morphological image processing
- Erosion/dilation: region growing/ shrinking
- Opening/closing: e.g., removing "holes"
- Additional composite processes
- A kernel, or "structuring element", acts like a brush
- The *mmand* package can work in any number of dimensions, with arbitrary kernels

- Bioconductor package *EBImage* also offers (2D) morphology


Wikipedia/Renato Keshet

Binary morphology in 1D

```r
library(mmand)
x <- c(0,0,1,0,0,0,1,1,1,0,0)
```

0	0	1	0	0	0	1	1	1	0	0

```r
kernel <- c(1,1,1)
```

1	1	1

```r
erode(x,kernel)
```

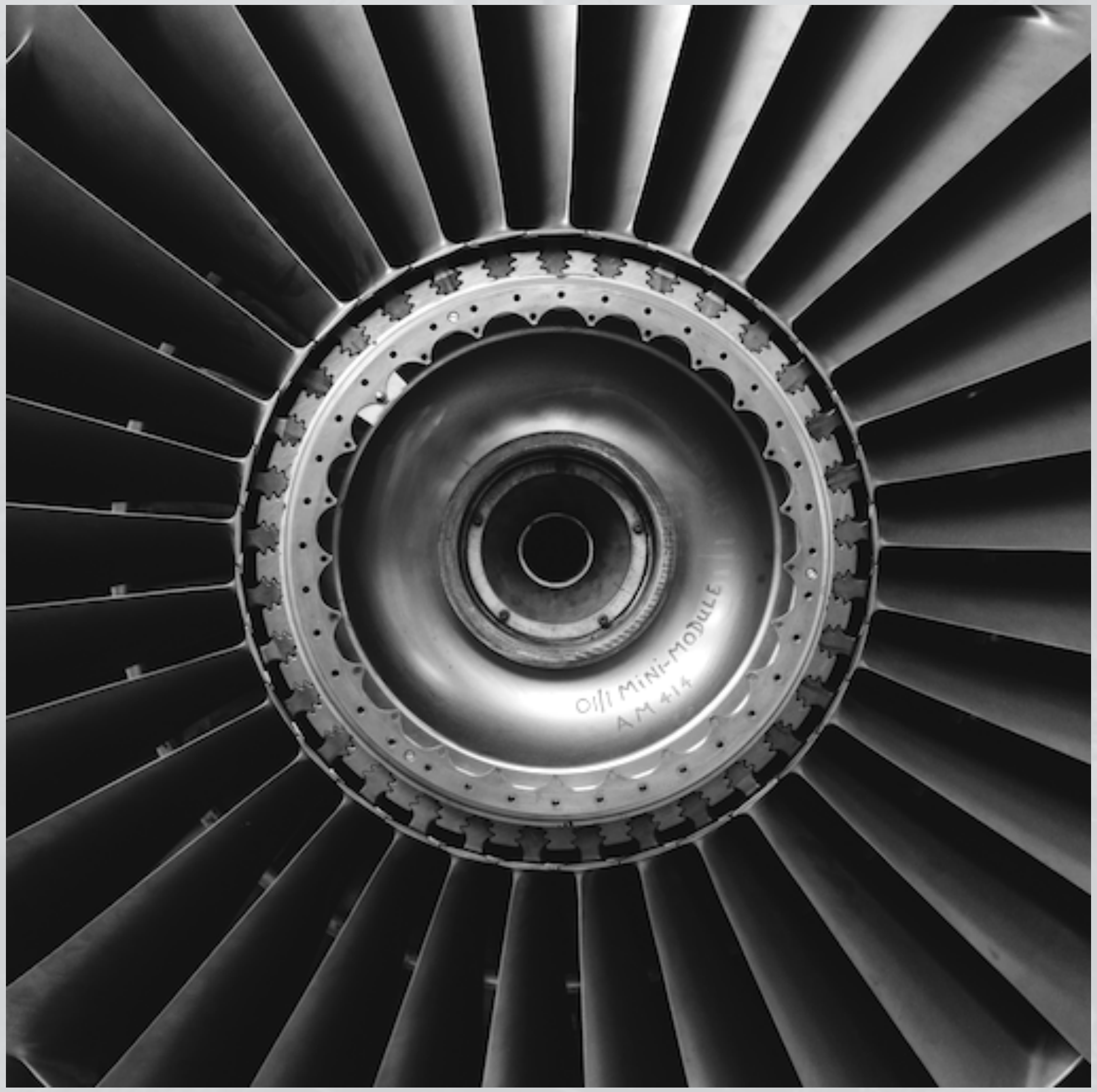
0	0	0	0	0	0	0	1	0	0	0

```r
dilate(x,kernel)
```

0	1	1	1	0	1	1	1	1	1	0

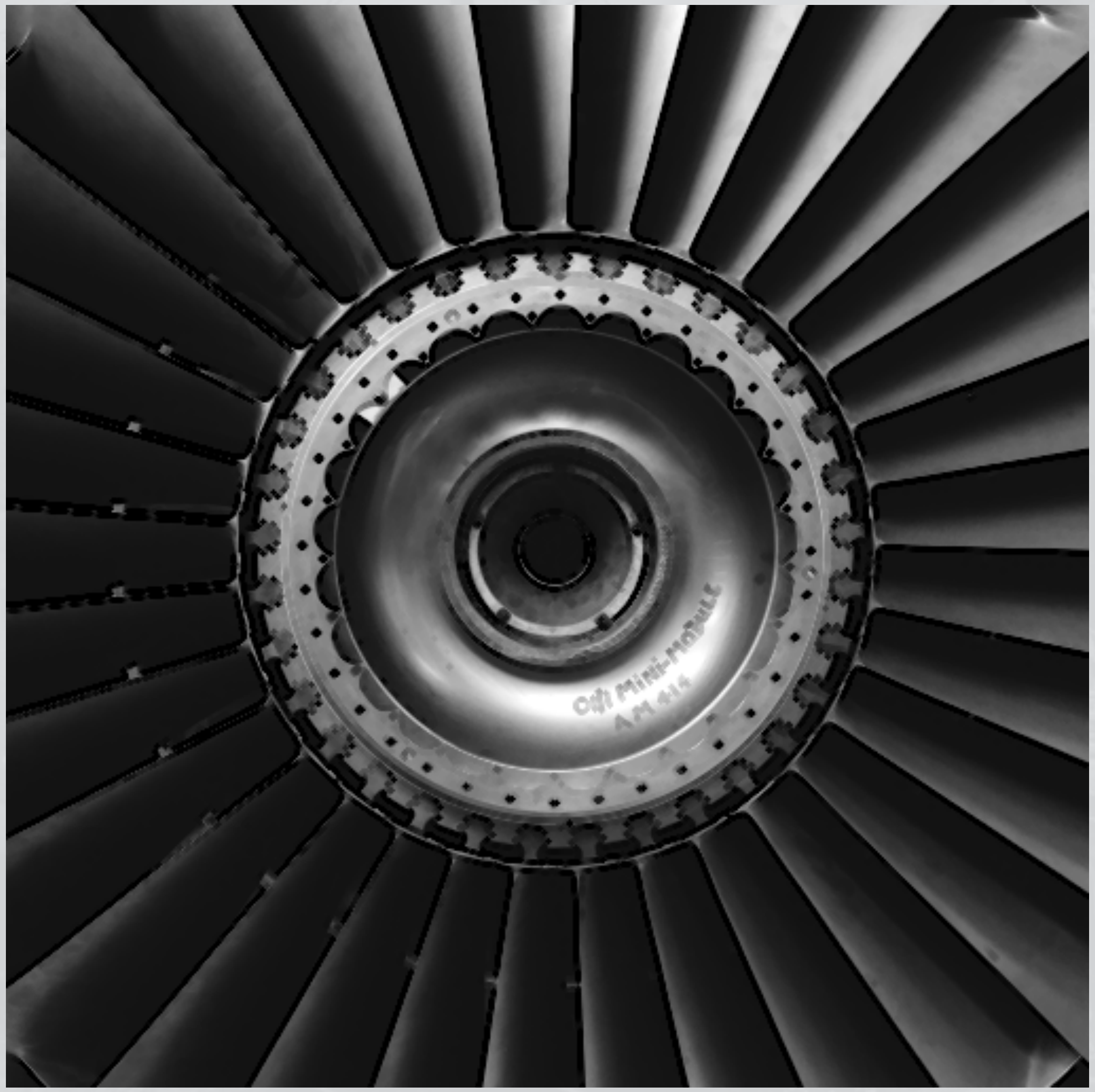
Two dimensions

```
library(png)
fan <- readPNG(system.file("images", "fan.png", package="mmand"))
display(fan)
```



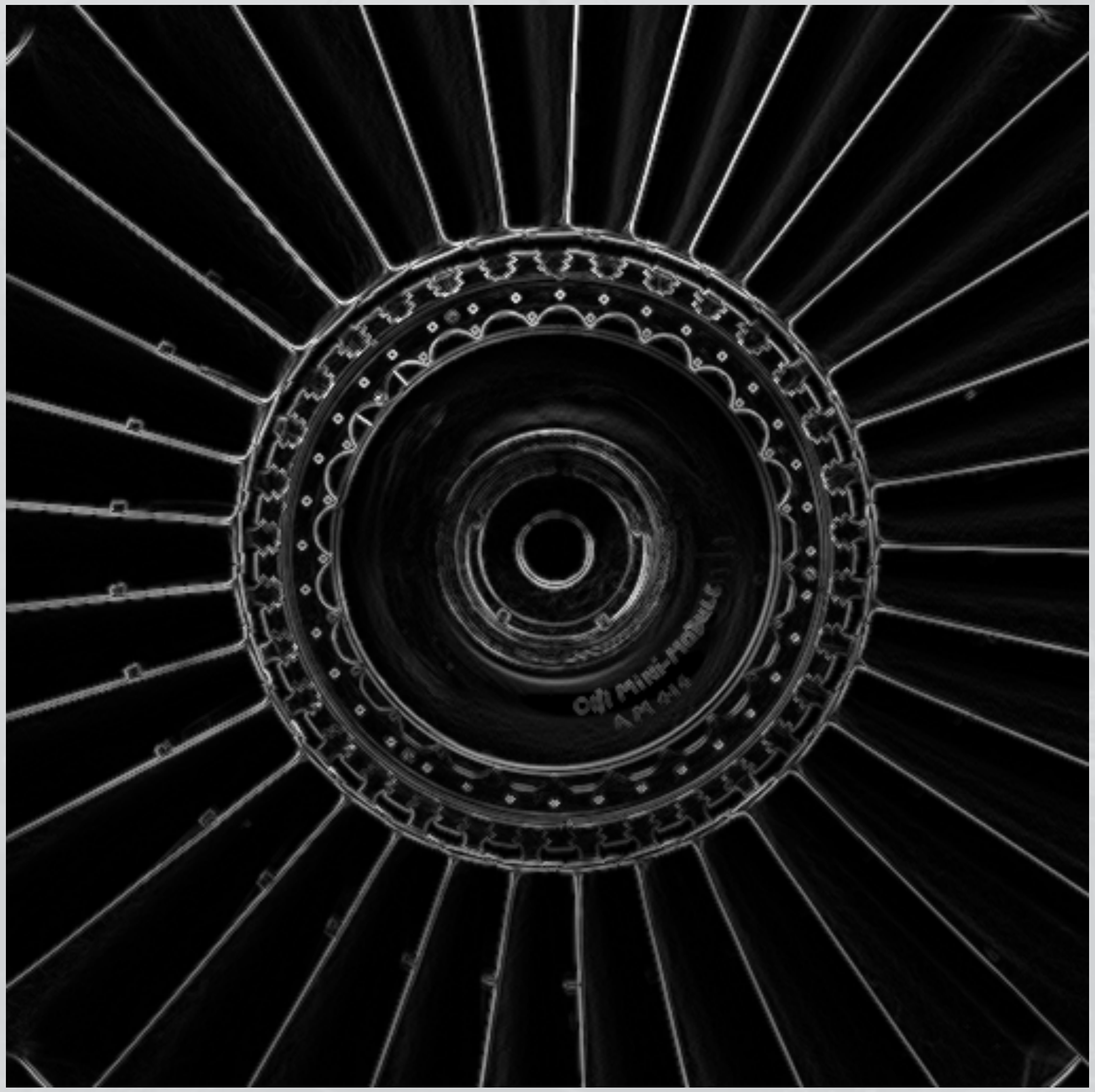
Greyscale morphology in 2D

```
kernel <- shapeKernel(c(3,3), type="diamond")
display(erode(fan,kernel))
```



Morphological gradient

```
kernel <- shapeKernel(c(3,3), type="diamond")
display(dilate(fan,kernel) - erode(fan,kernel))
```



Resampling

- Indexing between elements

```
x <- c(0,0,1,0,0)
x[2.5]
# [1] 0
```

- R truncates 2.5 to 2 and returns the second element
- In some cases there is conceptually a value at location 2.5 but we don't know it
- Best guess is probably that it's 0, or 1, something in between
- Using *mmand* we can interpolate using different sampling kernels

```
# "Nearest neighbour"
resample(x, 2.5, boxKernel())
# [1] 1

# Linear interpolation
resample(x, 2.5, triangleKernel())
# [1] 0.5

# Mitchell-Netravali cubic spline
resample(x, 2.5,
  mitchellNetravaliKernel(1/3,1/3))
# [1] 0.5347222
```

- An entire image of any dimensionality can be resampled similarly
- Allows regridding, upsampling and downsampling

Upsampling a quarter-size image

```r
fan_small <- readPNG(system.file("images", "fan-small.png", package="mmand"))
display(rescale(fan_small, 4, mnKernel()))
```

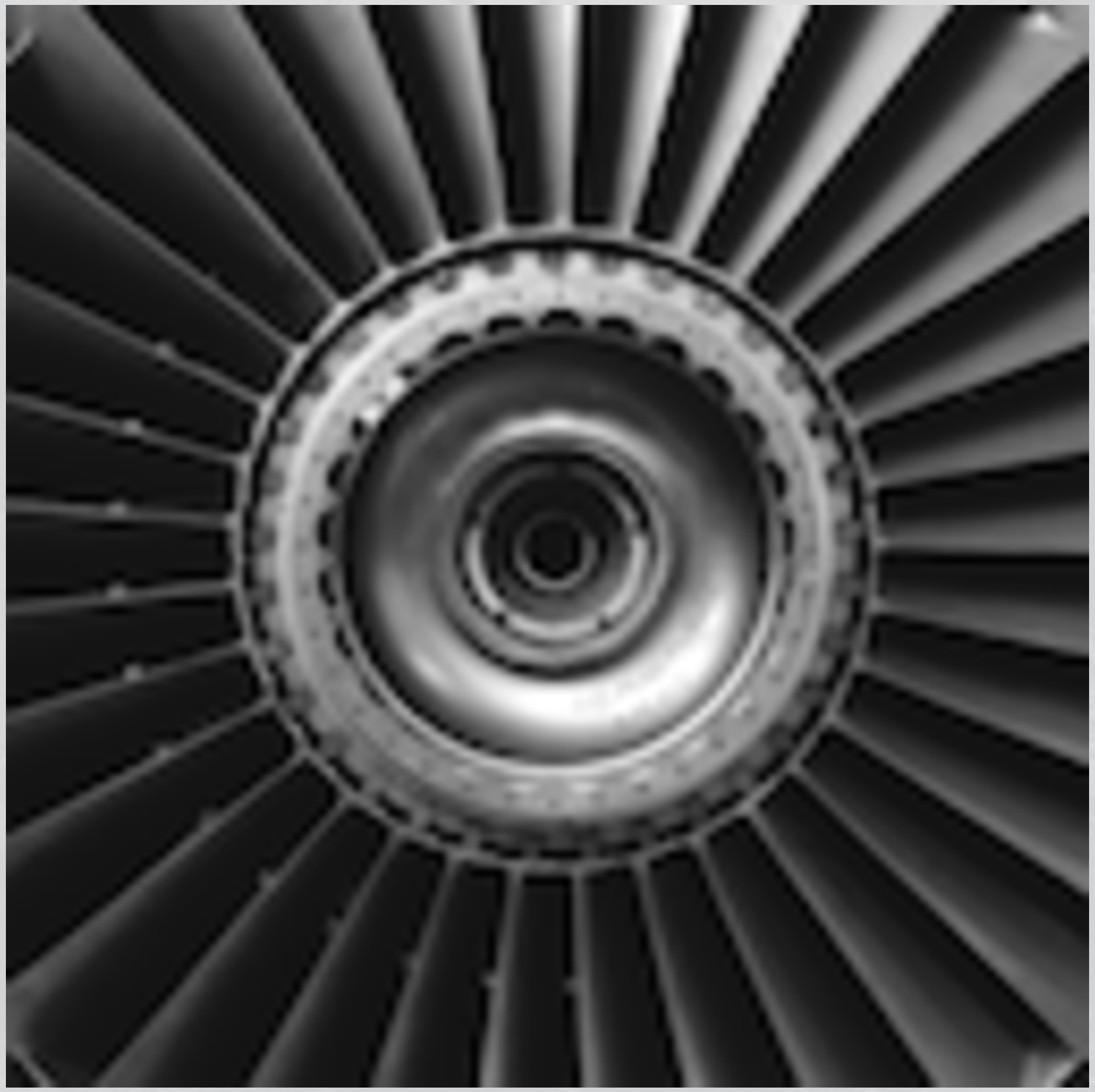
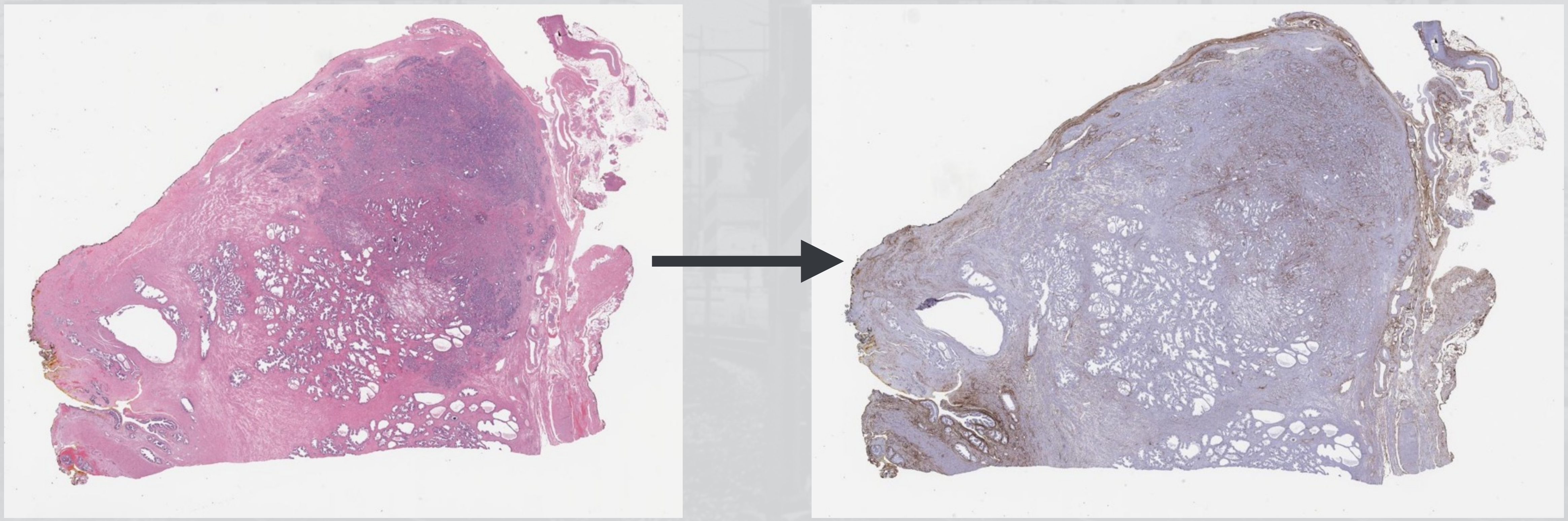


Image registration

- Aligning two related images
- Contrasts may be similar or different
- Pixel information may be combined
- Optimisation over a space of transformations (global/linear or local/nonlinear)
- Resampling to match the target image



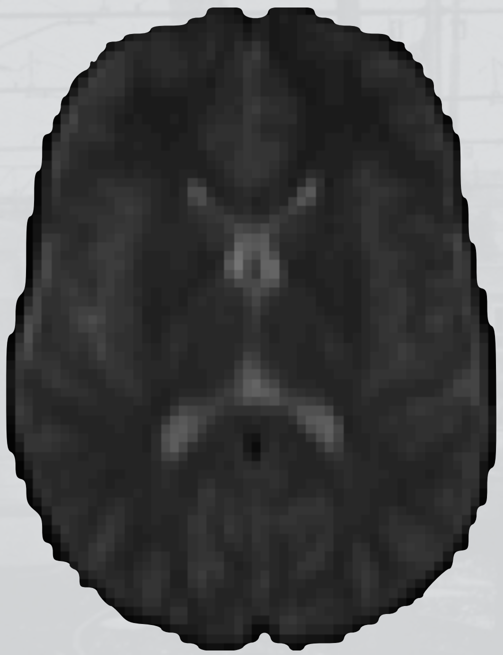
Jiří Borovec, Czech Technical University, Prague

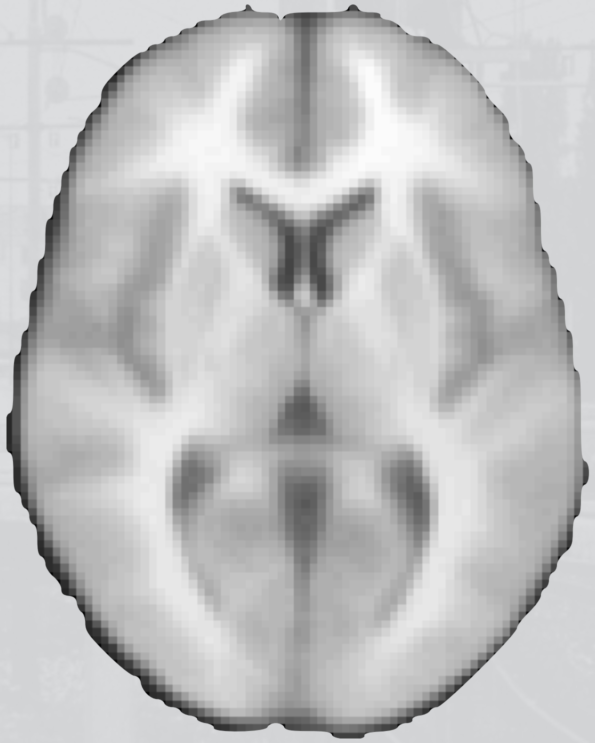
RNiftyReg scope

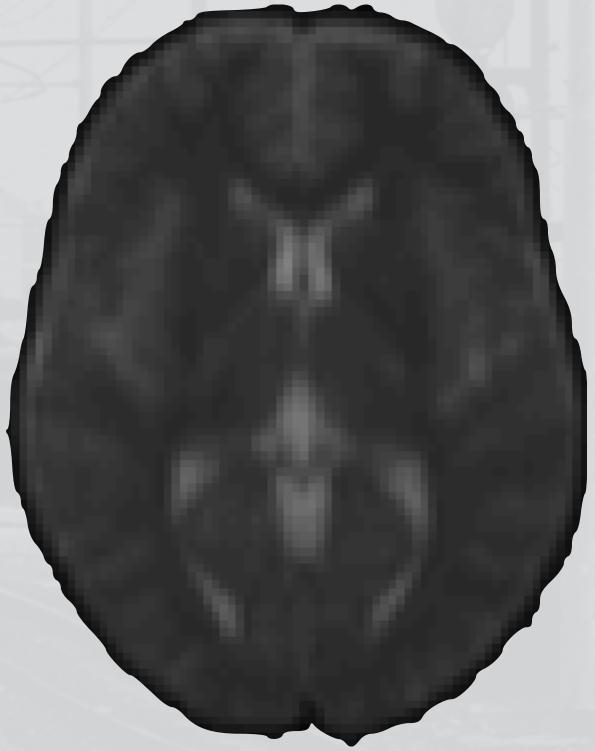
- R wrapper for NiftyReg (C++)
- Affine (linear) and nonlinear registration
- 2D or 3D (target may also be 4D)
- Control over cost function, resampling scheme
- Can apply transformations to other images or points
- Substantial update (to v2.0.0) currently in progress; hopefully coming in July

RNiftyReg usage (3D)

```r
library(oro.nifti)
library(RNiftyReg)
source <- readNIfTI(system.file("extdata","source.nii.gz",package="RNiftyReg"))
target <- readNIfTI(system.file("extdata","target.nii.gz",package="RNiftyReg"))

linear <- niftyreg(source, target, scope="affine")
nonlinear <- niftyreg(source, target, scope="nonlinear", init=linear$affine)
```


source

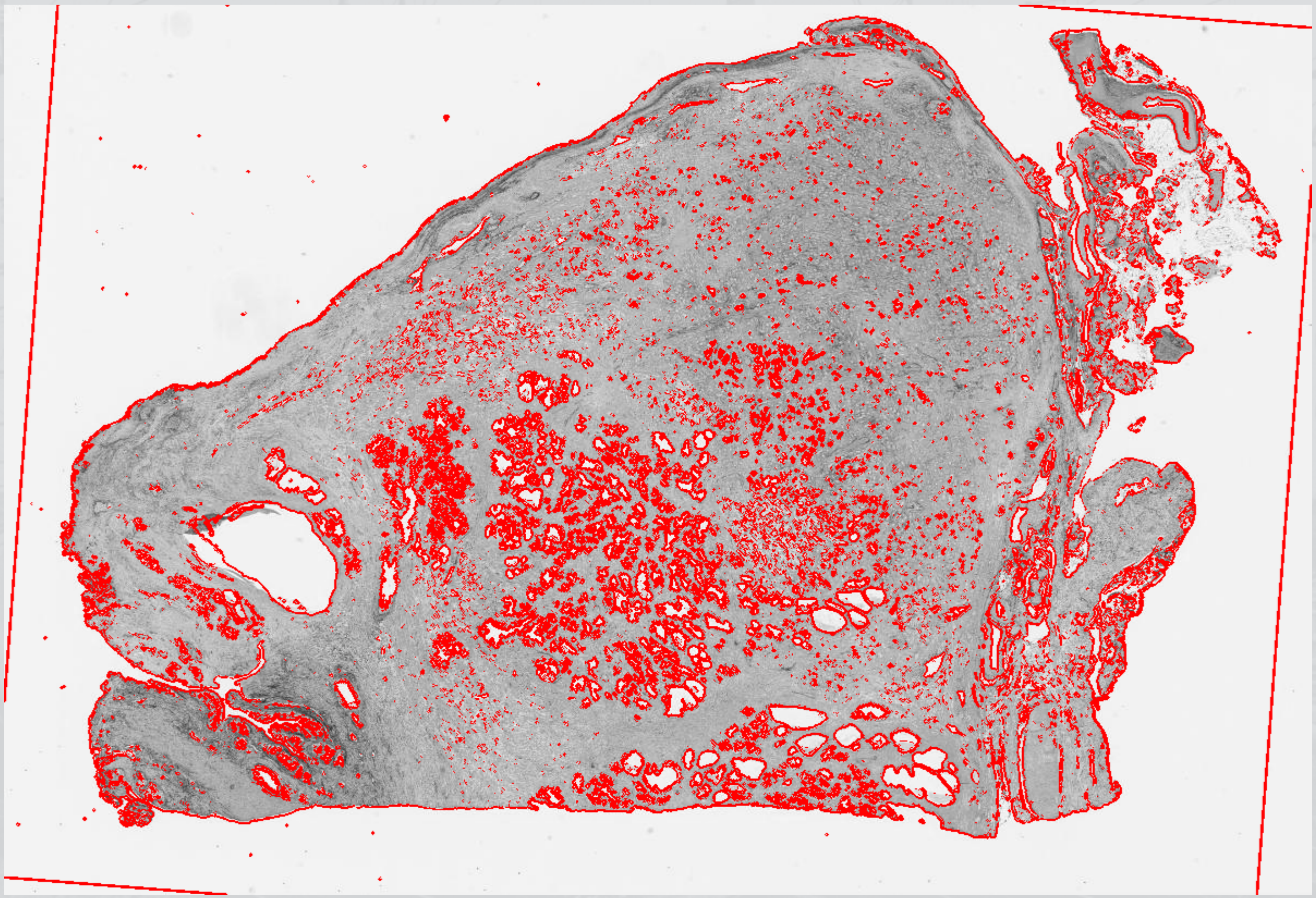

target


result (nonlinear)

Combining the packages: checking registration

```r
library(jpeg)
library(mmand)
library(RNiftyReg)

# Read images and convert to greyscale
source <- readJPEG("source.jpg")
target <- readJPEG("target.jpg")
source <- apply(source, 1:2, mean)
target <- apply(target, 1:2, mean)

# Register images
result <- niftyreg(source, target)

# Calculate morphological gradient
kernel <- shapeKernel(c(3,3), type="diamond")
gradient <- dilate(result$image,kernel) - erode(result$image,kernel)

# Display the results
display(target)
display(threshold(gradient,method="kmeans"), add=TRUE, col="red")
```

Combining the packages: checking registration



Some related packages

- Medical Imaging CRAN task view
- TractoR, a broader R-based platform for medical image analysis
- *EBImage* (BioC)
- *adimpro*, *ripa*, etc. (CRAN)
- *imager* (GitHub)

Contact details

- Email: j.clayden@ucl.ac.uk or code@clayden.org
- Twitter: @jonclayden
- GitHub: https://github.com/jonclayden
- TractoR: http://www.tractor-mri.org.uk