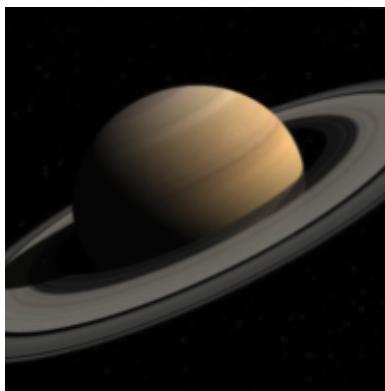


## Image Registration 2D

```
In[1]:= {i1, i2} = {, };
```

```
res = ImageAlign[i1, i2]
```

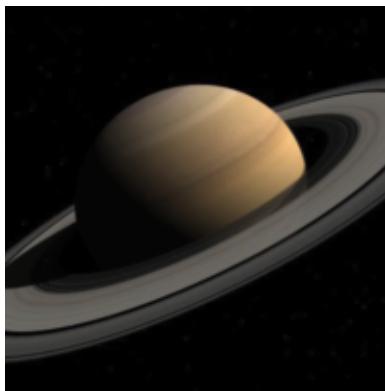


Out[1]=

```
In[2]:= {s, t} = FindGeometricTransform[, 
```

```
Out[2]= {0.279804, TransformationFunction[
$$\begin{array}{ccc|c} 0.00015142 & 1.00294 & -0.107685 \\ -1.00016 & 0.00172335 & 199.911 \\ \hline -2.56417 \times 10^{-6} & 9.9811 \times 10^{-6} & 1. \end{array}\}]}$$
 ] }
```

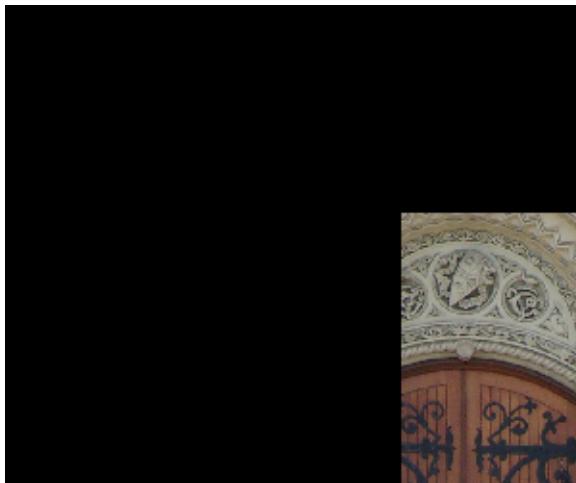
```
In[3]:= regImg = ImagePerspectiveTransformation[i2, t, DataRange → Full, Padding → 0]
```



Out[3]=

```
In[4]:= {p1, p2} = {, };
```

```
In[6]:= ImageAlign[p1, p2, Method -> "Fourier", Background -> Black]
```



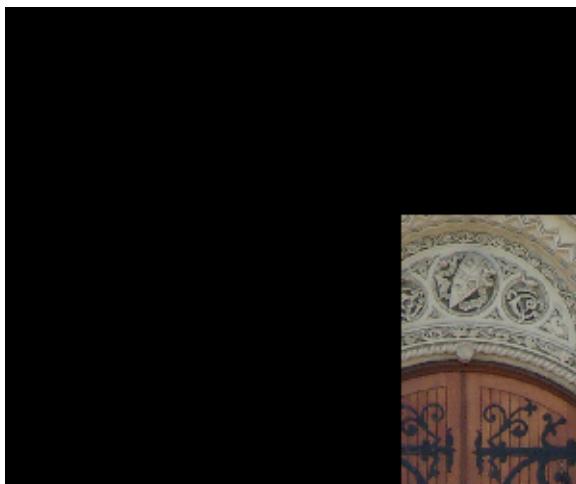
Out[6]=

FindGeometricTransform can take method → {"ImageAlign", "ImageAlignMethod"}

```
In[7]:= {err, trans} = FindGeometricTransform[p1, p2, Method -> {"ImageAlign", "Fourier"}]
```

Out[7]= {0.0000160713, TransformationFunction[ $\begin{pmatrix} 1 & 0 & 206 \\ 0 & 1 & -108 \\ 0 & 0 & 1 \end{pmatrix}$ ]}]

```
In[8]:= regImg = ImagePerspectiveTransformation[p2, trans, DataRange -> Full, Padding -> 0]
```



Out[8]=

```
In[9]:= mask = Closing[RegionBinarize[regImg, MaxDetect[regImg], 0.60], 2];
```

```
In[6]:= HighlightImage[p1, {Blue, mask}]
```



```
In[7]:= ImageAdjust[p1 + regImg]
```

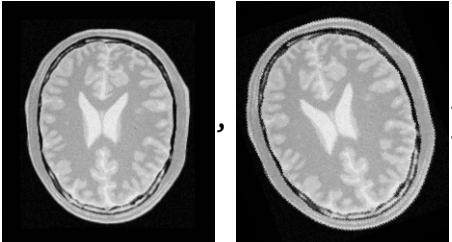


```
In[8]:= SetAlphaChannel[p1, ColorNegate@regImg]
```



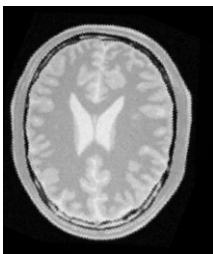
(\* translation, rigid, affine and similarity transformations are available \*)

```
In[1]:= {p1, p2} = {
```



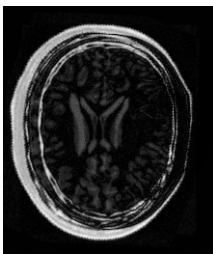
```
, };
```

```
In[2]:= res = ImageAlign[p1, p2, TransformationClass -> "Rigid", Background -> Black]
```



Out[2]=

```
In[3]:= diff1 = ImageAdjust@ImageDifference[Image[res, "Real64"], Image[p1, "Real64"]]
```

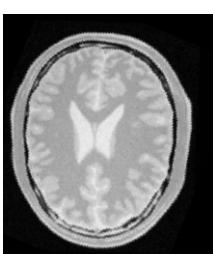


Out[3]=

```
In[4]:= {err, transF} =
FindGeometricTransform[p1, p2, Method -> {"ImageAlign"}, TransformationClass -> "Rigid"]
```

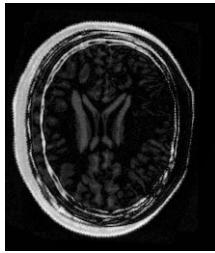
```
Out[4]= {0.00348832, TransformationFunction[ $\left(\begin{array}{cc|c} 0.926667 & 0.375883 & -48.5789 \\ -0.375883 & 0.926667 & 48.8606 \\ \hline 0. & 0. & 1. \end{array}\right)$ ]}
```

```
In[5]:= regImg = ImagePerspectiveTransformation[p2, transF, DataRange -> Full, Padding -> 0]
```



Out[5]=

```
In[1]:= diff2 = ImageAdjust@ImageDifference[Image[regImg, "Real64"], Image[p1, "Real64"]]
```

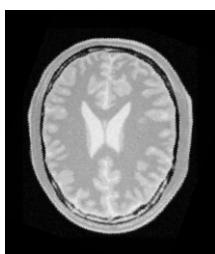


Out[1]=

```
In[2]:= diff1 == diff2
```

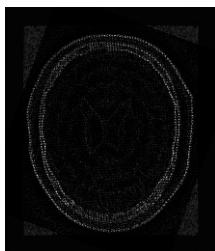
Out[2]= True

```
In[3]:= res2 = ImageAlign[p1, p2, TransformationClass -> "Affine", Background -> Black]
```



Out[3]=

```
In[4]:= diff3 = ImageAdjust@ImageDifference[Image[p1, "Real64"], Image[res2, "Real64"]]
```

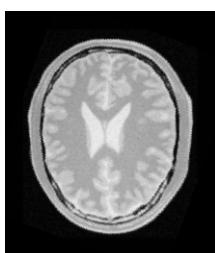


Out[4]=

```
In[5]:= {err, transF2} =
FindGeometricTransform[p1, p2, Method -> {"ImageAlign"}, TransformationClass -> "Affine"]
```

```
Out[5]= {0.000116006, TransformationFunction[ $\left(\begin{array}{cc|c} 0.853587 & 0.312593 & -24.3483 \\ -0.312464 & 0.85341 & 51.673 \\ \hline 0. & 0. & 1. \end{array}\right)$ ]}
```

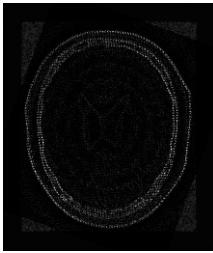
```
In[6]:= regImg2 = ImagePerspectiveTransformation[p2, transF2, DataRange -> Full, Padding -> 0]
```



Out[6]=

```
In[6]:= diff4 = ImageAdjust@ImageDifference[Image[p1, "Real64"], Image[regImg2, "Real64"]]
```

Out[6]=



```
In[7]:= diff3 == diff4
```

Out[7]= True

(\*overlay aligned images \*)

```
In[8]:= i1 = ; i2 = ;
```

```
In[9]:= aligned = ImageAlign[i1, i2, Background -> Black]
ImageCompose[aligned, {i1, 0.75}]
```

Out[9]=



Out[10]=



Use `FindGeometricTransform` to find the geometric transformation that aligns images:

```
In[11]:= i1 = ; i2 = ;
```

```
In[6]:= {e, tr} = FindGeometricTransform[i1, i2]
Out[6]= {0.341998, TransformationFunction[ $\left(\begin{array}{cc|c} 0.537851 & -0.0101117 & 153.442 \\ -0.13068 & 0.916323 & 0.645807 \\ \hline -0.0016047 & 0.000214252 & 1. \end{array}\right)$ ]}
```

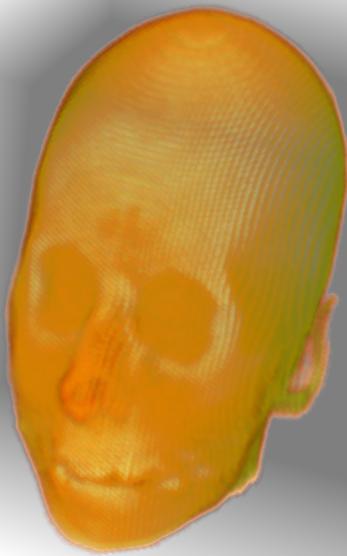
Stitch images together by transforming one and composing on top of the other one:

```
In[7]:= {w, h} = ImageDimensions[i2];
tmp = ImagePerspectiveTransformation[i2, tr,
    DataRange → Full, PlotRange → {{0, First@tr[{w, 0}]}, {0, h}}];
ImageCompose[tmp, i1, Round@(w, h) / 2]
```



## Image Registration 3D

```
In[17]:= regImg = ImageAlign[, , TransformationClass -> "Rigid"]
```



Out[17]=

```
In[4]:= {err, transF} = FindGeometricTransform[, 
```

```
, Method -> "ImageAlign", TransformationClass -> "Rigid"]
```

$$\text{Out}[4] = \left\{ 0.00941642, \text{TransformationFunction} \left[ \begin{array}{ccc|c} 0.862041 & -0.00142434 & -0.506836 & 33.5308 \\ 0.357732 & 0.710108 & 0.606445 & -34.9491 \\ 0.359044 & -0.704092 & 0.612652 & 35.7758 \\ \hline 0. & 0. & 0. & 1. \end{array} \right] \right\}$$

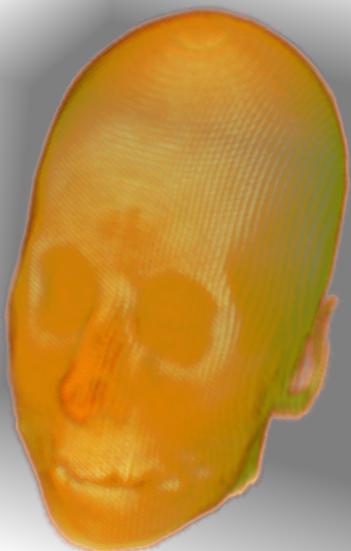
In[7]:= **regImgTrans =**

**ImagePerspectiveTransformation[**



**, transF, DataRange → Full, Padding → 0]**

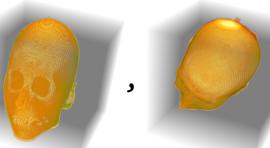
Out[7]=



In[20]:= **regImgTrans - regImg**

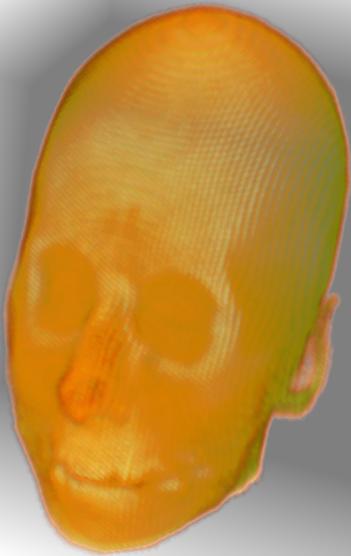
Out[20]=

```
In[18]:= regImg2 = ImageAlign[
```



```
] (*unspecified transformation class*)
```

```
Out[18]=
```



A comparison of the manually selected “rigid” transformation class versus the automated selection of transformation class shows that the result is actually quite good !!

`regImgTrans - regImg2`

Out[22]=

