Arithmetic Operations

Arithmetic operations between images are array operations which means that arithmetic operations are carried out between corresponding pixel pairs. The four arithmetic operations are denoted as

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

It is understood that the operations are performed between corresponding pixel pairs in f and g for $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$ where, as usual, M and N are the row and column sizes of the images. Clearly, s,d,p,v are images of size $M \times N$ also. Note that image arithmetic in the manner just defined involves images of the same size.

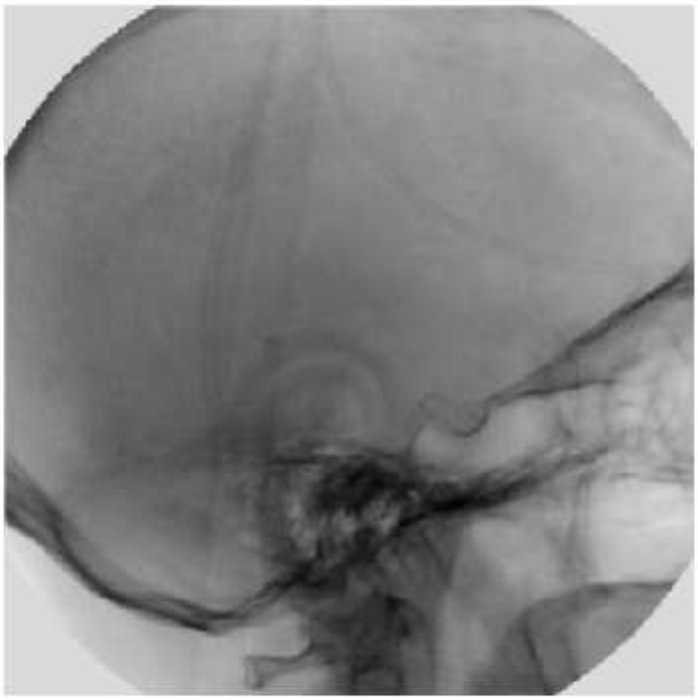
Example 2.6 Image subtraction for enhancing differences

A frequent application of image subtraction is in the enhancement of differences between images.

The first image shows a mask X-ray image of the top of a patient's head prior to injection of an iodine medium into the bloodstream.

In [27]:

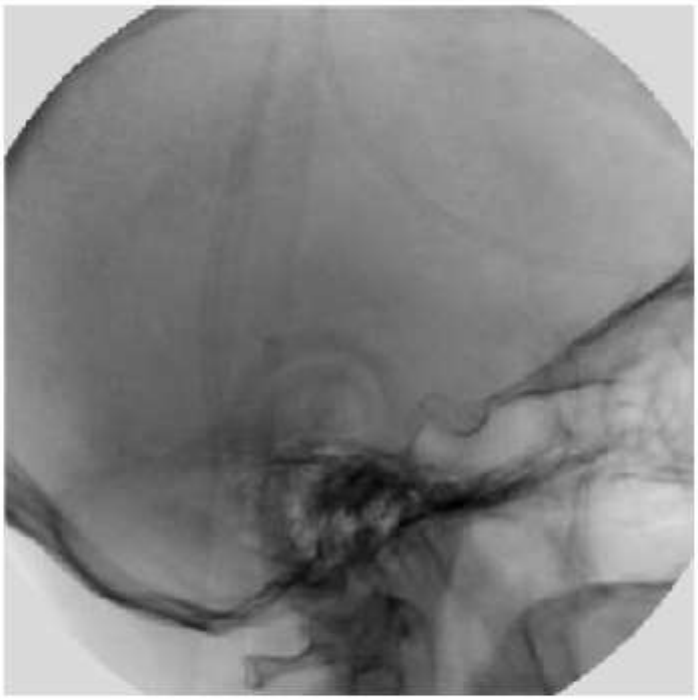
```
f = imread('Fig0228(a).tif');
f = tofloat(f);
imshow(f);
```



The second image is a sample of a live image taken after the medium was injected.

In [28]:

```
g = imread('Fig0228(a).tif');
g = tofloat(g);
imshow(g);
```



The third image is difference between the first image and the second image. Some fine blood vessel structures are visible in this image.

In [ ]:

```
s = f - g;
imshow(imcomplement(s), []);
```

The difference is clear in the 4th image, which was obtained by enhancing the contrast in 3rd image. The 4th image is a clear "map" of how the medium is propagating through the blood vessels in the subject's brain.

In [ ]:

```
ss = imadjust(s, stretchlim(s), [1 0]);
imshow(ss, []);
```

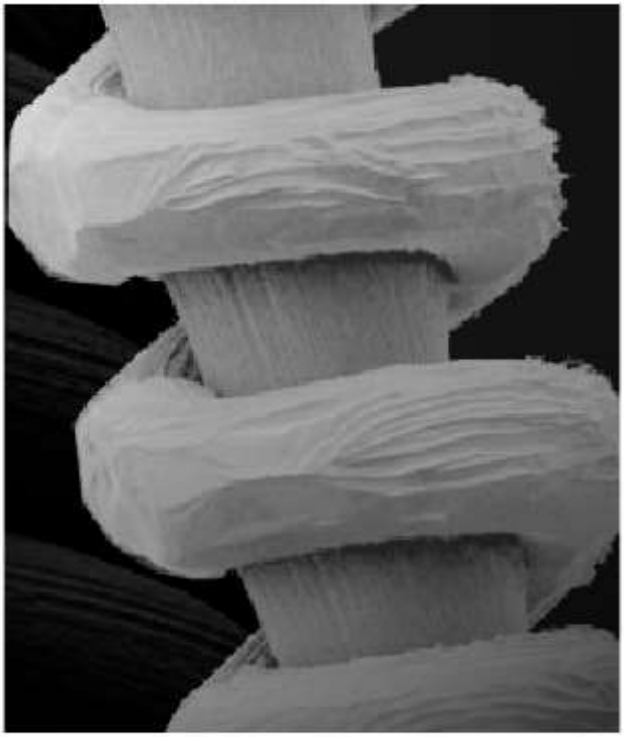
Example 2.7 Using image multiplication and division for shading correction.

An important application of image multiplication (and division) is shading correction. Suppose that an imaging sensor produces images that can be modeled as product of a "perfect image", denoted by $f(x,y)$, times a shading function, $h(x,y)$; that is, $g(x,y) = f(x,y)h(x,y)$. If $h(x,y)$ is known, we can obtain $f(x,y)$ by multiplying the sensed image by the inverse of $h(x,y)$. If $h(x,y)$ is not known, but access to the imaging system is possible, we can obtain an approxmation to the shading function by imaging a target of constant intensity. The following pictures show an example of shading correction.

In [21]:

```
g = imread('Fig0229(a).tif');
g = tofloat(g);
imshow(g);
```

警告: 图像太大，无法在屏幕上显示；将以 67% 显示
> In images.internal.initSize (line 71)
  In imshow (line 336)



In [22]:

```
h = imread('Fig0229(b).tif');
h = tofloat(h);
imshow(h);
```

警告: 图像太大，无法在屏幕上显示；将以 67% 显示
> In images.internal.initSize (line 71)
  In imshow (line 336)

In [24]:

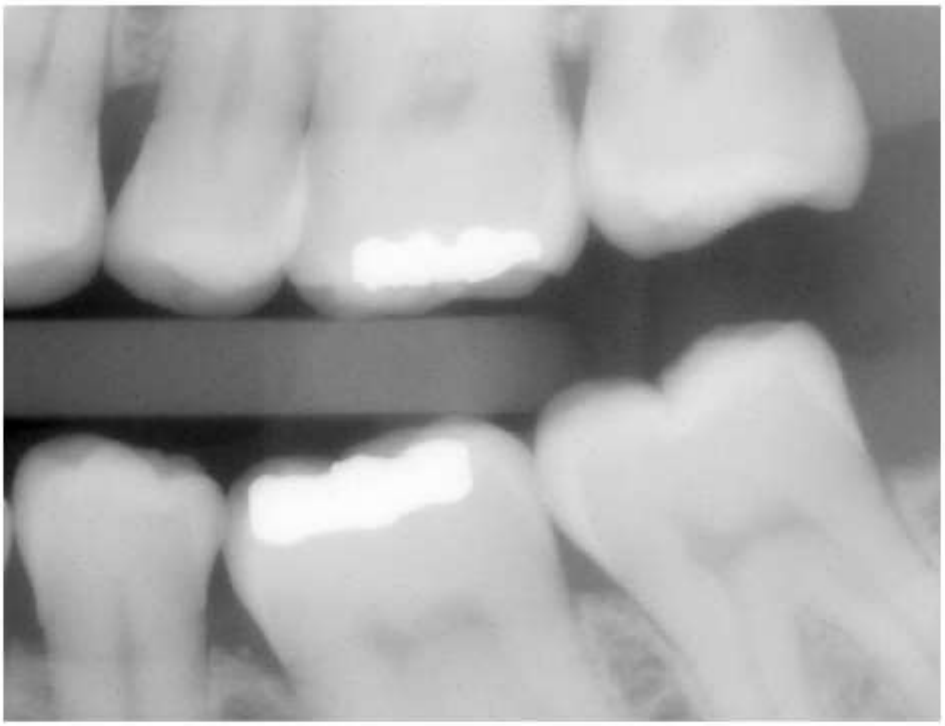
```
f = imdivide(g, h);
imshow(f);
```

警告: 图像太大，无法在屏幕上显示；将以 67% 显示
> In images.internal.initSize (line 71)
  In imshow (line 336)

Another common use of image multiplication is in masking, also called region of interest (ROI), opertions. The process, illustrated in the following images, consist simply of multiplying a given image by a mask image that has 1s in the ROI and 0s elsewhere. There can be more than one ROI in the mask image, and the shape of the ROI can be arbitrary, although rectangular shapes are used frequently for ease of implementation.

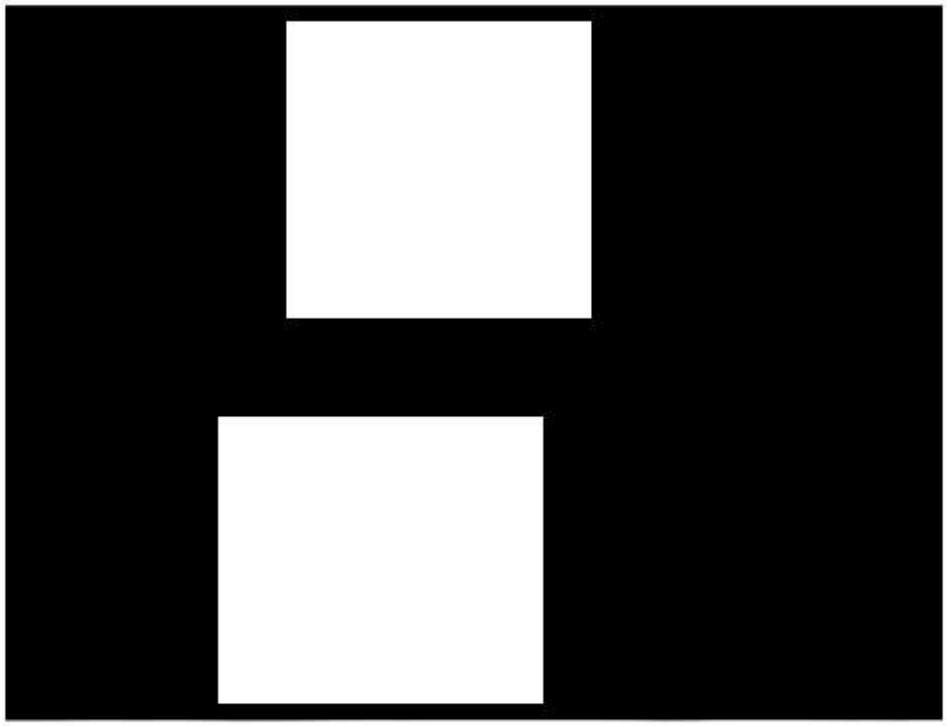
In [25]:

```
f = imread('Fig0230(a).tif');
f = tofloat(f);
imshow(f);
```



In [26]:

```
roi = imread('Fig0230(b).tif');
roi = tofloat(roi);
imshow(roi);
```



In [27]:

```
g = immultiply(f, roi);
imshow(g);
```