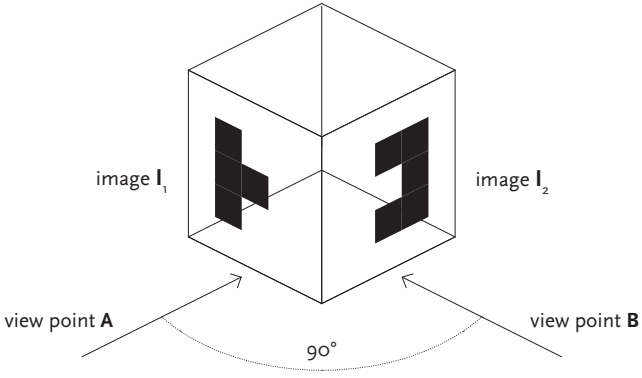
Given two images $\mathbf{I}_1$ and $\mathbf{I}_2$ we wish to map them onto a 3D space **S** such that when viewing $\mathbf{I}_1$ from point **A** only $\mathbf{I}_1$ is visible and when the viewer is positioned 90° at view point **B**, only $\mathbf{I}_2$ is visible.



Given two images that satisfy the property: $\forall\langle x,y\rangle \in \mathbf{I}_1, \exists\langle z,y\rangle \in \mathbf{I}_2$ using any two points $\langle x,y\rangle \in \mathbf{I}_1$ and $\langle z,y\rangle \in \mathbf{I}_2$ we can create a point in 3D space $\langle x,y,z\rangle$ that forms part of both images at view points **A** and **B**. Ignoring perspective (we will come to that later), given a point $\langle x,y,z\rangle$ that is visible from both **A** and **B**, the point $\langle x\pm1,y,z\rangle$ will change the image visible from **A**, but not from **B**. Conversely the point $\langle x,y,z\pm1\rangle$ will change the image visible from **B**, but not from **A**. It is this principle that we use to build up the two images.

As each image must have the same range of y values (height of the actual image, not the canvas) we can calculate the size of S to be sum of points in $\mathbf{I}_1$ and $\mathbf{I}_2$ - height of image.

$$\mathbf{I}_1 = \texttt{[(2,2),(2,3),(2,4),(3,3)]} = 4$$
$$\mathbf{I}_2 = \texttt{[(3,2),(3,4),(4,2),(4,3),(4,4)]} = 5$$
$$h = 3$$
$$|\mathbf{S}| = 4 + 5 - 3 = 6$$

```
#!/bin/python3

from os.path import join
from scipy.misc import imread
from collections import defaultdict
from itertools import cycle, islice

path = "/Users/boliver/Desktop"

imgA = imread(join(path,'Untitled-1.bmp'), flatten=0)
imgB = imread(join(path,'Untitled-1.bmp'), flatten=0)

def bmp2points(bitmap):
    pointmap = defaultdict(list)
    for (y,line) in enumerate(bitmap):
        for (x,point) in enumerate(line):
            if point: pointmap[y].append(x)
    return pointmap

def combine(pointsA, pointsB):
    xyz = []
    for (y,xs) in pointsA.items():
        if y in pointsB:
            zs = pointsB.get(y)
            zipped = islice(zip(cycle(xs), cycle(zs)), max(len(xs),len(zs)))
            xyz = xyz + [(x,y,z) for (x,z) in zipped]
    return xyz

def combineBitmaps(bitmapA, bitmapB):
    return combine(bmp2points(bitmapA), bmp2points(bitmapB))
```