Enamel resampling code in python

Hi Paul,

My brother Gregory has built this python code which can help with comparing mineralization in teeth as they grow larger. The following few slides will help explain what the code does.

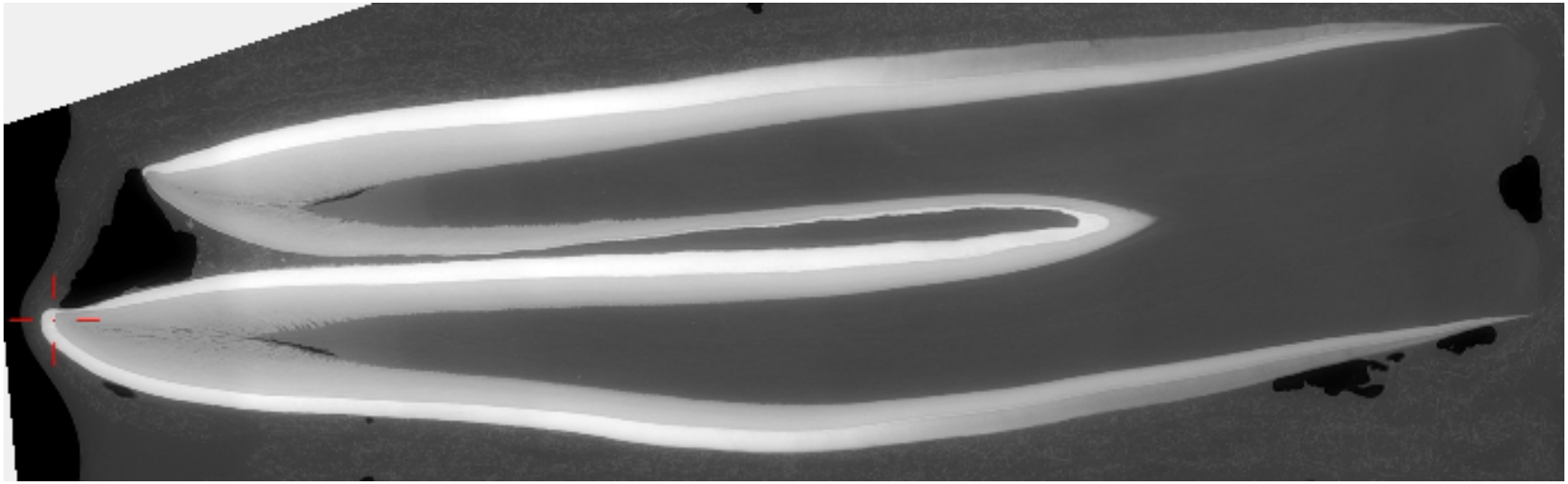
I'd like the images and graphs produced by this code to contribute to a portion of the poster for AAPAs.

Because images produced using 13um scans look better, I'll use these for display, but data graphs will likely be derived from 45um resolution scans.

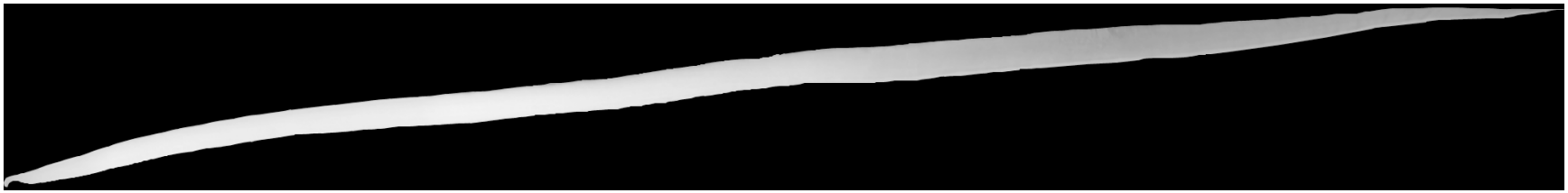
```
#numpy.empty(shape, dtype=float, order='C') returns a new array:
#the array's type and shape are defined, but entries are not initialized.
#Here, markerPos is the array [(number of markers), 2], without defined entries.
#markerDeriv is the array [number of markers], without defined entries.

def place_markers(x, spl, spacing=5.):
    fineness = 10
    xFine = np.linspace(x[0], x[-1], fineness*len(x))
    yFine = spl(xFine)
    derivFine = np.empty(len(xFine), dtype='f8')
    for i,xx in enumerate(xFine):
        derivFine[i] = spl.derivatives(xx)[1]
    derivFine = filters.gaussian_filter1d(derivFine, fineness*spacing)
    dx = np.diff(xFine)
    dy = np.diff(yFine)
    dist = np.sqrt(dx*dx + dy*dy)
    dist = np.cumsum(dist)
    nMarkers = int(dist[-1] / spacing)
    if nMarkers > 1e5:
        raise ValueError('nMarkers unreasonably high. Something has likely gone wrong.
            ')
    markerDist = np.linspace(spacing, spacing*nMarkers, nMarkers)
    markerPos = np.empty((nMarkers, 2), dtype='f8')
    markerDeriv = np.empty(nMarkers, dtype='f8')
    cellNo = 0
    for i, (xx, d) in enumerate(zip(xFine[1:], dist)):
        if d >= (cellNo+1) * spacing:
            markerPos[cellNo, 0] = xx
            markerPos[cellNo, 1] = spl(xx)
            markerDeriv[cellNo] = derivFine[i+1] #spl.derivatives(xx)[1]
            cellNo += 1

    return markerPos, markerDeriv

def get_image_values_2(img, markerPos, DeltaMarker, step=0.1):
    ds = np.sqrt(DeltaMarker[:,0]*DeltaMarker[:,0] + DeltaMarker[:,1]*DeltaMarker[:,1]
        )
    nSteps = img.shape[0] / step
    stepSize = step / ds
    n = np.linspace(0., nSteps, nSteps+1)
    sampleOffset = np.einsum('i,ik,j->ijk', stepSize, DeltaMarker, n)
    sampleStart = np.empty(markerPos.shape, dtype='f8')
    sampleStart[:,:] = markerPos[:,:]
    nMarkers, tmp = markerPos.shape
    sampleStart.shape = (nMarkers, 1, 2)
    sampleStart = np.repeat(sampleStart, nSteps+1, axis=1)
    samplePos = sampleStart + sampleOffset

    samplePos.shape = (nMarkers*(nSteps+1),2)
    resampImg = imginterp.map_coordinates(img.T, samplePos.T, order=1)
    resampImg.shape = (nMarkers, nSteps+1)

    return resampImg.T[:,:]
```

Sample of code that resamples enamel images into more standard arrays that can be compared.

Before code is run, we locate the developmental plane in VG, capturing the Mes-Ling dentine horn, maximum average B-L breadth, and maximum extension:

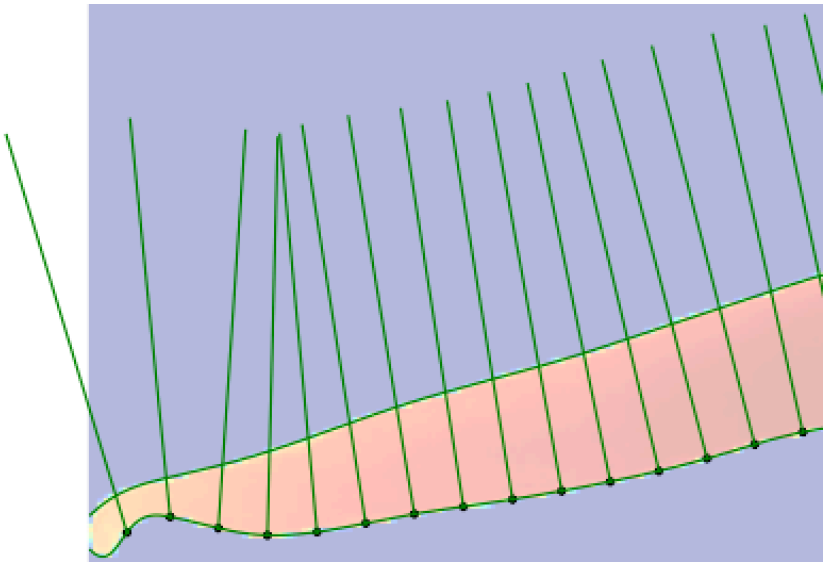


We trace enamel edges in photoshop; this involves partial pixel sampling at periphery right now, though perhaps shouldn't:



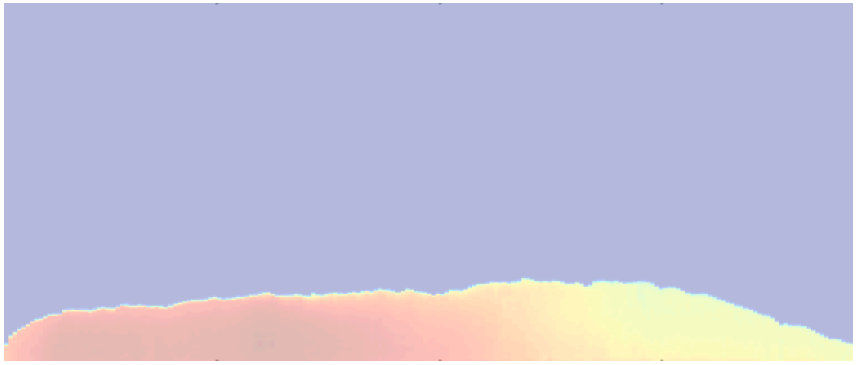
Pixel values outside enamel are set to zero.

In the code, the upper and lower edges of the enamel are located, and a smoothed line is fitted to the lower edge using spline interpolation. Pixel values are then sampled perpendicular to the spline.



Another possibility (not coded) is to sample average areas instead of individual pixels.

Sampled pixel values are then transferred into a new, standardized array.



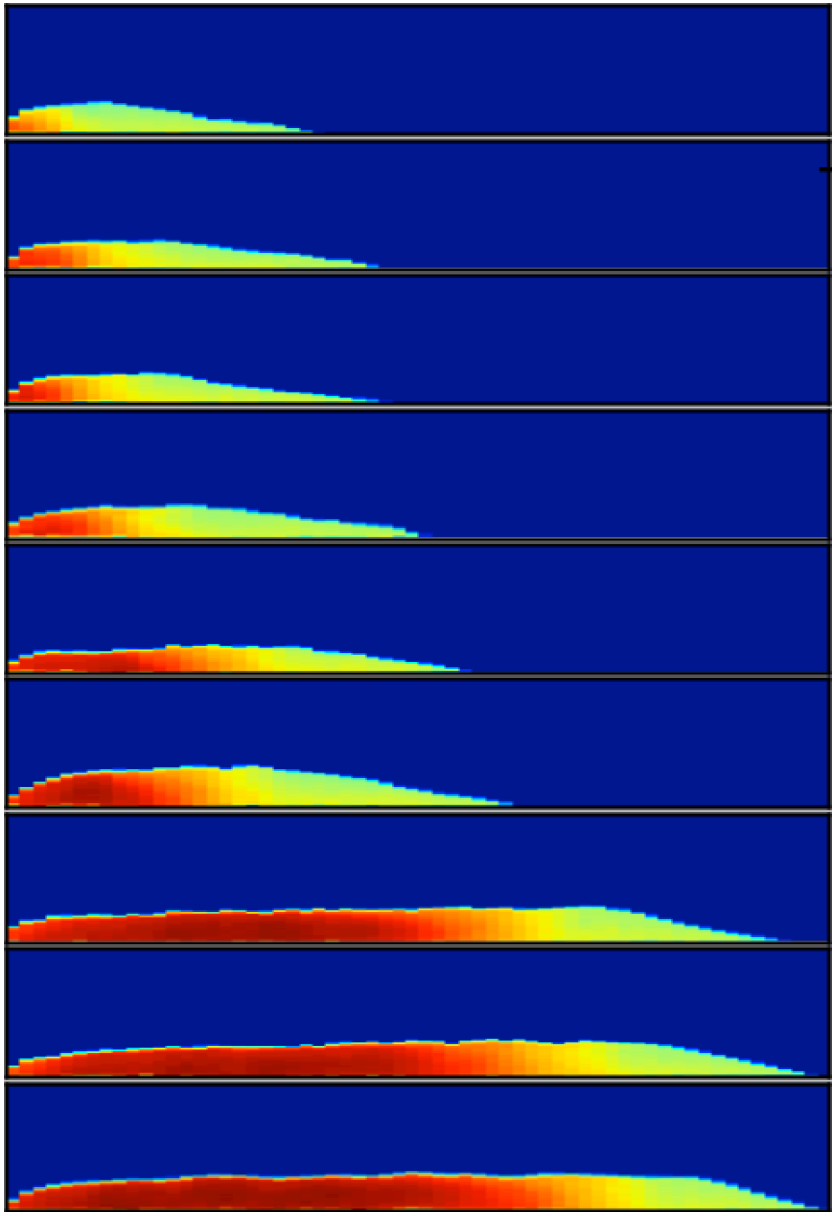
The fineness of sampling in both x and y directions can be specified.

This resampling makes enamel in different teeth far more comparable, but does not remove all variation.

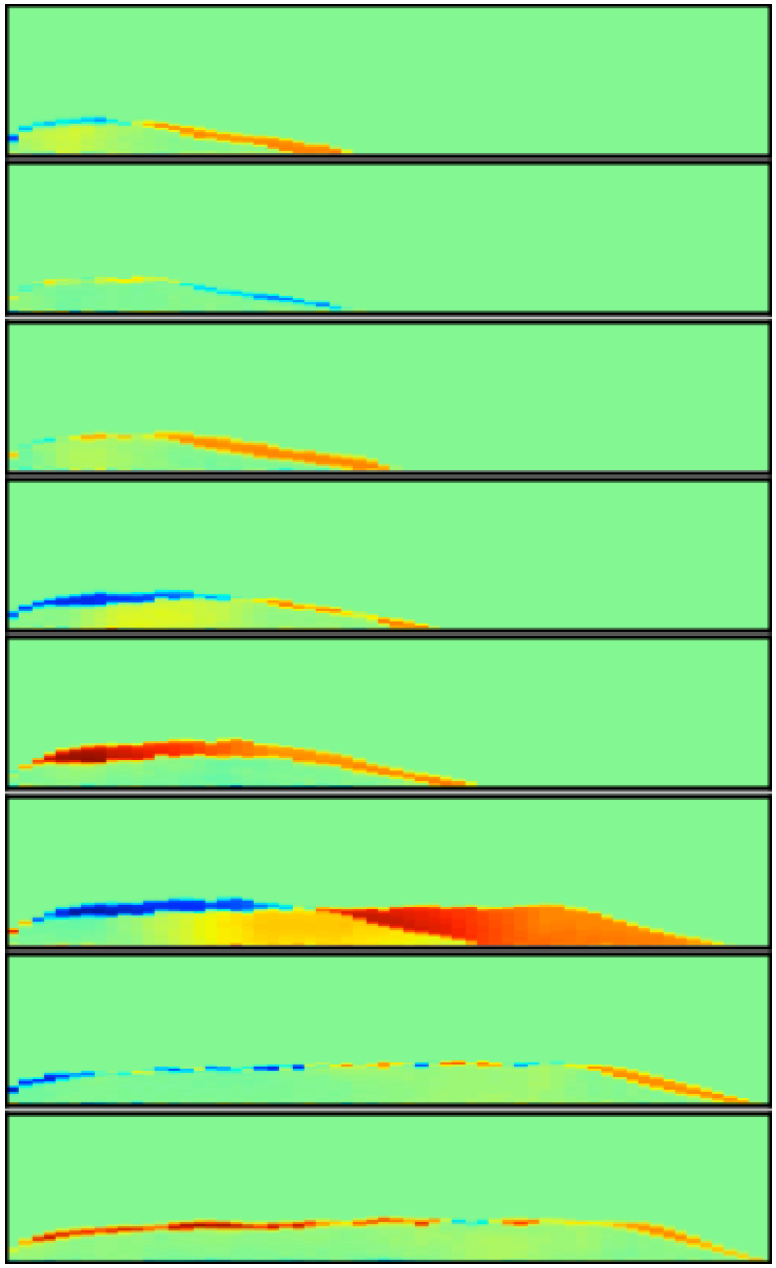
At right are nine 45um resolution resampled enamel arrays, with highly mineralized enamel to the left (cuspal) in red, and developing enamel in yellow and green.

Even when standardized, there's still substantial shape variation, which makes the teeth difficult to compare, especially moving away from the EDJ and towards the outer enamel surface.

There is also variation in growth rates, so that sometimes older teeth are actually smaller than younger ones.



The differences in shape are especially apparent when you subtract one shape (from a younger individual) from another (from an older individual). Red, orange and yellow at right represent mineralized tissue in the older tooth not present in the younger; blue represents tissue present in the younger, not the older. Material should only be added as teeth grow, but the different shapes result in apparent loss.



It's possible, for any point in the resampled arrays, to measure pixel scores as they increase (or sometimes decrease) from younger to older teeth (above right).

It's also possible to measure "added pixel value" as teeth become older (below right). With all 40 first molars at 45um resolution sampled, and sampling closer to the EDJ, we might be able to build a better picture of how mineral is being added over time at any given point.

We'll need to calibrate pixel scores to densities for the poster.