UFAZ 2018-19 DSA2 Project

Flattening the pot

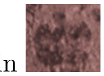
Last year, the DSA project consisted of understanding how BMP images were constructed, in order to extract the central part of an image.

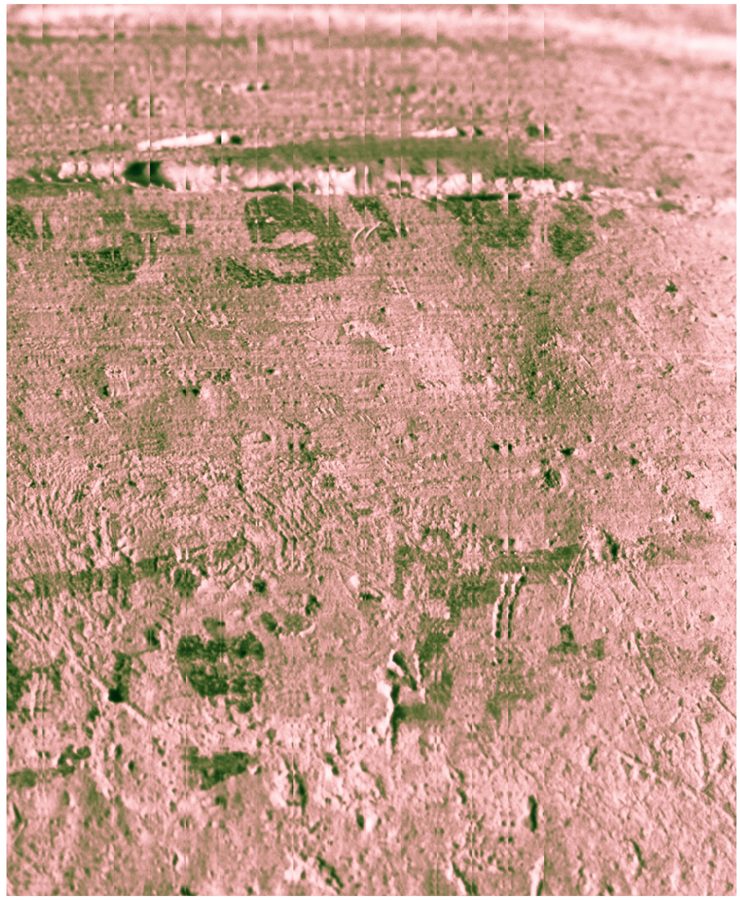
During my visit to UFAZ last July, I went back to the museum with correct equipment and I took 360 infrared photos of the pot, with a 1^o rotation between each photo. I then extracted a strip of 480x2230 pixels of each photo resulting in 360 strips that now need to be stitched together (I did a bit of colour processing so that ink markings can show up in green so that they can be better seen).

Quality image stitching is a very complex process, but fortunately, many panorama stitching software exist that can be used for this (look for “panorama stitching” in Google for instance).

However, there is a problem with the stitching if we use vertical strips of this pot, because, as you can see in the image above, the pot is not a cylinder.

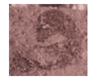
Take a part of the pot below (taken from original image L1020651).

I have stitched 15 strips to reconstruct the same part of the pot. However, if I stitch the strips so as to have the correct aspect ratio for the small pumpkin  on the bottom left of the image, you see that the top of the image is elongated :

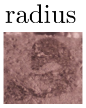


Worse than being elongated: the strips on the top of the image do not match, meaning that the pixels are doubled... This is not good at all!!!

This is related to what scientist call in geometry the phenomena of stereo-graphic projection when you pass from sphere to plan you cannot have the real dimension (you already see that everyday with the earth map). You See more video's in different languages here with this link, if you like, before starting your project to understand what is the real challenge.

Now, if I stitch the strips so that the reversed E sign  looks good:

Not only the bottom part of the image is shrunk, but where it is shrunk, we have lost some information (overlapped pixels): the pumpkin is smashed! And the reversed E does not look good either.

As said before, this comes from the fact that the vase is not a cylinder. A 1^o rotation means that the image will move by more pixels on the middle of the vase than on the top, because the radius of the vase is different in the middle than on the top. (This is the case too for the  letter because the letter is not on a vertical part of the vase.

How can **you** solve this problem?

This is the object of your DSA2 project : you need to resize the strips so that they have a variable width matching the variable radius of the vase, so that a 1^o rotation corresponds to the same number of pixels on top of the image or on the bottom of the image.

You will find the the https://tinyurl.com/ycw727zd directory some zipped files: AZCropped-Pot.zip corresponds to 360 480x2230 strips extracted from the 360 images of the pot, with a 1^o rotation between each image. Because BMP is an image format that is much more easy to work with than .jpg images, you will find the same images in .bmp format in the CroppedPotBMP.zip file, but as you will see, it is very large (800MB) so it will take you a very long time to download it.

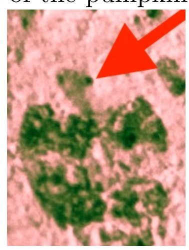
So that you can work without loading the whole file, I extracted 16 BMP images that I put in 16Strips.zip (only 33MB).

Please use these images for the project : once your program is working for those 16 images, it will work for all 360 images.

Now, here is what you need to do.

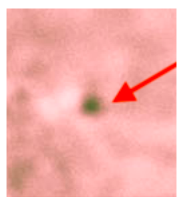
Open the 16 images, and take a notable feature of the image.

For instance, take the tip of the stem of the pumpkin :



Note its position in image AzCroppedPot25.bmp (474, 1548) and look at its position in image AzCroppedPot26.bmp (408, 1548). This means that at height 1548 of the image, a 1^o rotation means a shift by 66 pixels.

Now, take another notable feature of the image at another height, for instance the right part of the black spot on top of the image:



You see that the coordinates of this spot change from (358,227) in image AzCroppedPot25.bmp to (320,225) in image AzCroppedPot26.bmp, so it only moved by 38 pixels. This is because the radius of the vase at height 225 is 0.576 times the radius of the vase at height 1548. Note that there is a small variation of height of the spot between images 25 and 26: 227 in image 25 and 227 in image 26. We do not care much of vertical variations. Horizontal variations is what we want to correct for (the stitching is horizontal, not vertical).

What this means is that if we want strips to match when flattening the vase, we must have an equal shift of pixels for features of the vase at the bottom of the image and at the top of the image.

So we must elongate the pixels at height 225 by a factor $1/.576 = 1.737$ compared to pixels at height 1548.

What about pixels at height 1000? Well, it depends on the radius of the vase at this height.

Work to be done:

1. Choose a step (every 200 pixels? because the image is 2230 pixels high, this means 10 steps) and for each step, find a scaling factor that you will store in a float array of size 10. (Take the larger number of pixels shift as being factor 1 so that all the found values are < 1: when you invert them to create the Pixel Scaling Factor *psf*, all values will become > 1). The Pixel Scaling Factor to apply to each pixel will be :

$$psf = \frac{1}{Scaling\ Factor}$$

   Thing to do in step 1: **Create an array of 10 floats that contains the pfs at each step.**

2. Then, supposing that the maximal *psf* value is 1.75, this means that at this height, the width of 1 pixel needs to become 1.75. This means that the strip width of the modified image now needs to be $1.75 \times 480 = 840$ pixels.

   Thing to do in step 2: **Create a new BMP image of height 2230 and width 480× maximal *psf* called fTarget**.

3. Thing to do in step 3: **Open a strip BMP image (take the file name from the command line) called fSource**.

4. Now, for all lines of the image, enlarge the pixels of each line by a factor corresponding to the *psf* of their height. Suppose, for instance, that for height 0 to 200 pixels, the *psf* factor is 1.75. This means that for all lines from 0 to 200, each pixel of the source BMP strip must be copied onto 1.75 pixel (only on the x axis : we keep the pixel as it is on the y axis: this makes things much smpler).

   How do you increase a pixel size by 1.75? 1.75 is 1 + .75. Thins means that the colour of the first fTarget pixel will keep the RGB colours of the first fSource pixel.

   For the second fTarget pixel of the line, we need to know the colour of the second fTarget pixel of the line, and make a **weighted addition** for each colour: Red of second fTarget pixel = .25× Red value of fSource first Pixel + .75* Red value of fSource second pixel. The same for Green and for Blue.

   Things to do for step 4:

   (a) **Determine what is the formula to apply to obtain the value of the 3rd pixel, knowing that the color of a target pixel is the weighted sum of the colors of the pixels covered on the source.**

   (b) **determine what is the formula for the n^{th} target pixel**.

   (c) Don't forget that by doing this, different lines of the target file will not be of the same length. Because you have written last year a program that extracts strips from the center of an image, **the lines must be centered** and after the lines are resized, you will be able to use your program of last year to extract a center strip of 480 pixels from the target file. It is these strips from elongated pixels that can be stitched perfectly by a panorama software.

5. Thing to do for step 5: Out of two input files : an input file `xxx.bmp` containing a strip and an input file `psf` containing the values of the array, **write the complete algorithm and program to resize the pixels depending on their height on the image and write the result in a file called `xxxResized.bmp`**. Don't forget that the lines must be centered.

In order to get additional points, you can write a small script that will:

1. run your `resize` program on all files of a directory.
2. use your program of last year to extract a 480 pixels strip from the resized files
3. glue all the images together to finally obtain... **THE FLATTENED POT!**

You must of course provide the source code of all the files. You can work in groups of 2. You can discuss between groups, but each group must write his/her own code.

Note: the top of the pot will be elongated compared to the bottom. This is normal. If the viewer wants to examine the top part of the pot, he will have to resize the image horizontally so that the scale factor makes things look ok for the top.

But by doing this the bottom of the pot will be shrunk. But... contrarily to before, shrunk oixels will not have disappeared: the information will still be there.

Refinements:

- The 1^o rotation is approximative. You could determine the pfs over several images rather than between 2 consecutive images. You could look at the shift between 10 images and divide by 10.
- You can of course have more than 10 steps.
- You could find a function to match the evolving radius of the pot and use the function to compute the pfs for each line.