Documentation for the VAMP project

Pavlo Shchelokovskyy

August 5, 2011

Contents

1 Development Notes

1.1 Extraction of data from images

For now everything is done in Python with different additional packages.

The pseudocode-like draft for the algorithm for processing single image looks like this:

1. load picture to array
2. get user input
3. calculate positions of pipette walls
4. calculate pipette radius
5. calculate the axis of the system
6. extract data along the axis from the picture
7. find positions of 3 characteristic points on the axis
8. calculate all derived parameters (vesicle radius, tip length, surface area, volume)
9. calculate dilations and tensions and fit them with some defined function

Python packages aside from Python Standard Library used or (probably) going to be used in this program:

- *numpy* — basic package for numeric stuff, defines *array* data type
- *scipy* — depends on *numpy* and provides a lot of mathematical functions, which I most probably will need [1]
- *wxPython* — GUI programming
- *psyco* — JIT compiler for Python, may use it to speed up the execution
- *Python Image Library (PIL)* — basic image processing functionality, but may eventually drop it (see Section 1.1.1)
- *matplotlib* — for plotting results

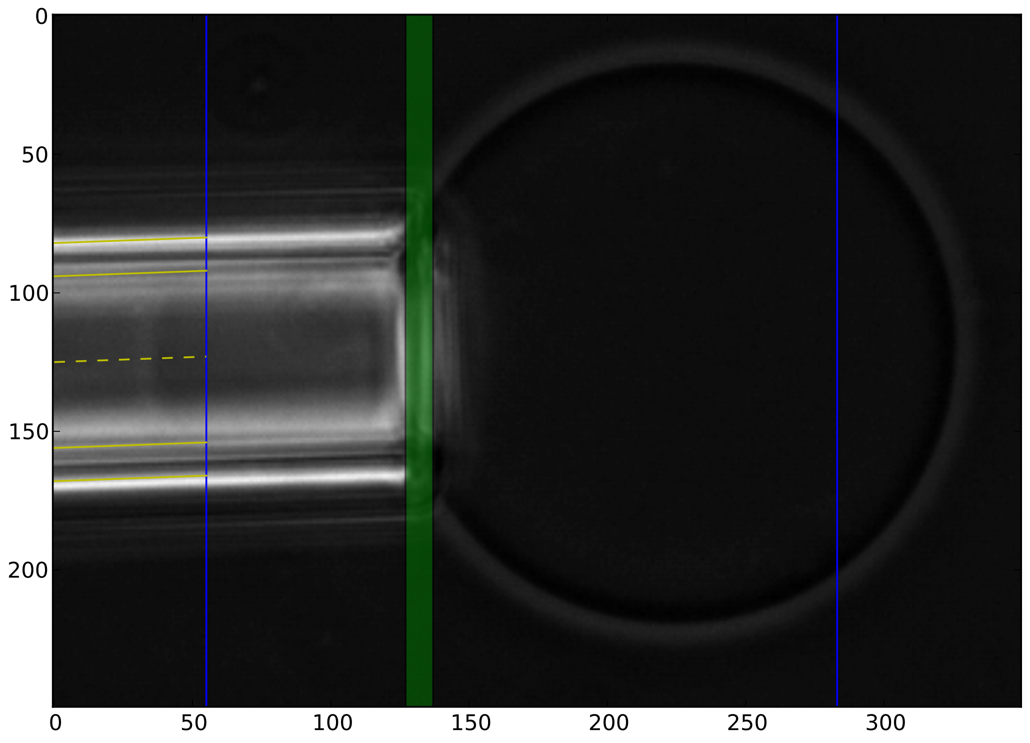


Figure 1: User supplied parameters visualized on the aspirated vesicle image. Left blue line is x_{a0}, right blue line is x_{v0}, green region is approximate pipette tip position, yellow lines define the approximate pipette walls positions, yellow dashed line is approximate pipette axis.

1.1.1 Load picture to array

This is done with *scipy* and *PIL*, which allows for direct conversion from image file to *numpy* array.

On the other hand, *wxPython*'s wxImage object could be probably used, thus eliminating the need of *PIL* package, since I'm going to use *wxPython* for GUI programming anyway. Unfortunately, wxImage still doesn't support *numpy*-like array interface as of now, but this is planned feature for addition. Or buffers interface seems to be able to do this …

1.1.2 Get user input

First of all user should crop the image so that everything but the vesicle is left out and rotate the picture so that the pipette *sticks from left*. Let us assume the size of the image along the x-axis is now N. User supplies several sets of coordinates to define approximate regions of features locations (see Figure 1):

- overestimated position of the aspirated vesicle part x_{a0} so that the tip of the aspirated vesicle part must be always to the left from this, $x_a \in (0, x_{a0})$;
- overestimated position of the outer part of the vesicle x_{v0} so that the tip of the outer part of the vesicle is always to the right of it, $x_v \in (x_{v0}, N)$;
- approximate position of the pipette tip, x_{p1}, x_{p2} so that the pipette tip is always in this region, $x_p \in (x_{p1}, x_{p2})$;
- approximate positions of pipette walls.

1.1.3 Define pipette walls

A cross-sections of the pipette are taken in two points $x = 0$ and $x = x_{a0}$. The pipette walls are then taken as global minima in the regions corresponding to the approximate pipette walls positions for each cross-section (see Figure 2), producing sets of coordinates $(x_1, y_1), (x_2, y_2)$ and $(x_3, y_3), (x_4, y_4)$. The y-positions can be further defined more accurately with subpixel resolution (see Section 1.1.6).

1.1.4 Calculate pipette radius

Distance (unsigned) from point (x_0, y_0) to line $y = kx + b$ is given by:

$$\delta = \left| \frac{kx_0 + b - y_0}{\sqrt{k^2 - 1}} \right| . \tag{1}$$

Thus the diameter of the pipette for single image is calculated as mean of distances from every of 4 points to the line corresponding to the opposing wall. Then these values are again averaged between all images.

1.1.5 Calculate the middle line (axis) of the system

Having obtained the set of reference points from Section 1.1.3 one builds a system axis as a bisector between straight lines formed by pipette inner walls. This is done by simply defying the line going through 2 middle points $\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$ and $\left(\frac{x_3+x_4}{2}, \frac{y_3+y_4}{2}\right)$. Then the axis $y = kx + b$ is defined by coefficients

$$k = \frac{y_1 + y_1 - y_3 - y_4}{x_1 + x_2 - x_3 - x_4},\ b = \frac{y_1 + y_2}{2} - k\frac{x_1 + x_2}{2} . \tag{2}$$

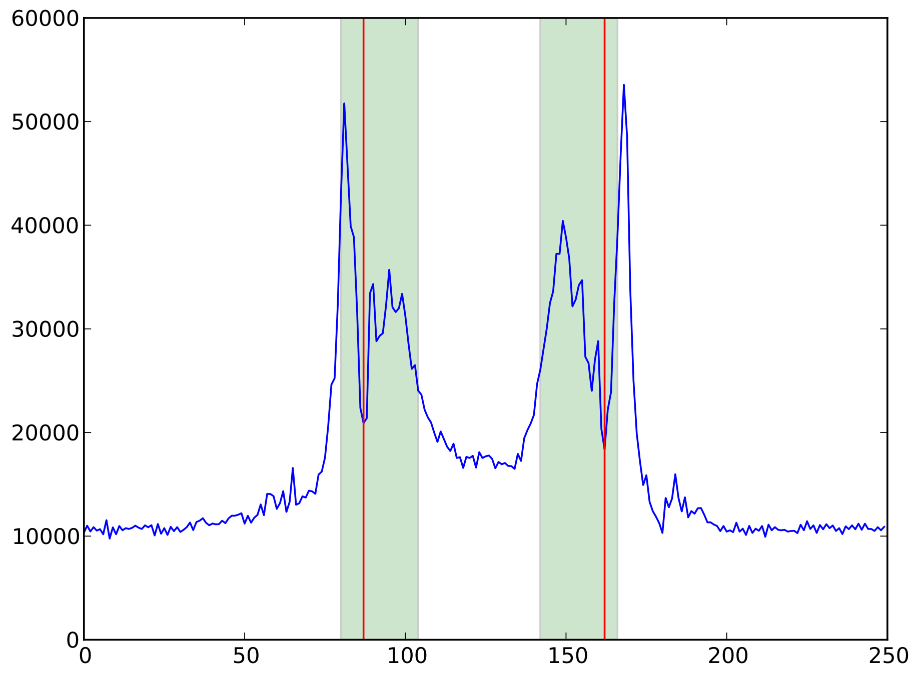


Figure 2: Brightness profile of the pipette cross-section at $x = x_{a0}$. Green regions correspond to user-supplied approximate pipette wall positions, red vertical lines denote the actual found positions of pipette walls.

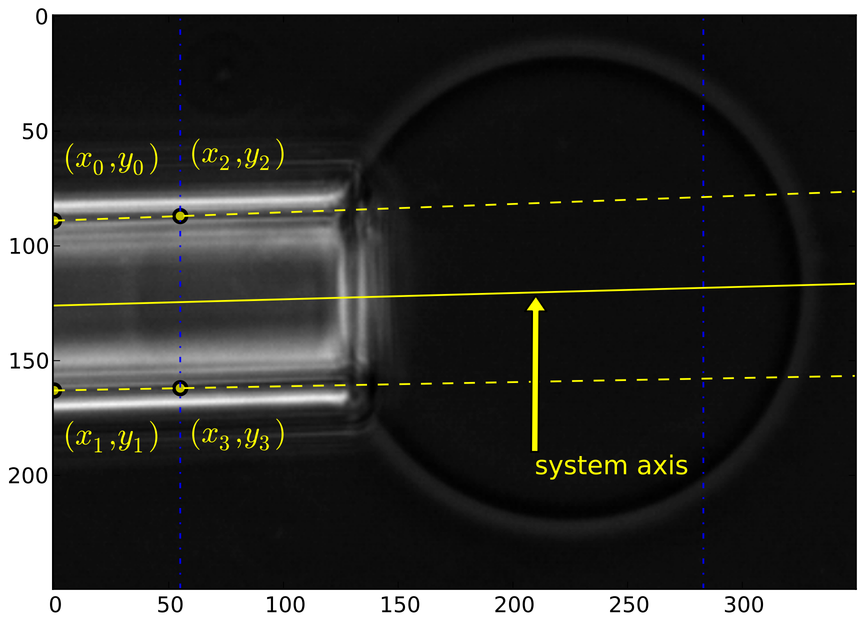


Figure 3: Sketch of general situation of pipette system alignment on the image

After that the brightness profile along this axis is extracted from image with special function `scipy.ndimage.map-coordinates`, which returns spline-interpolated values of brightness “between” pixels. To define the set of points at which interpolated brightness should be evaluated first intersections of the system axis with boundaries of image are calculated. To define the number of points for evaluation the biggest of differences in x or y coordinates between intersection points is taken.

Small pitfall — since at the end we need real distances between features, an attention have to be paid to keep the scale of profiles equal. The above procedure does not keep the scale, because the line has only as many points as there are along one of it's coordinates, but it should be $\sqrt{x^2+y^2}$. Hence, the distance between 2 consecutive points on brightness profile along arbitrary tilted straight line with number of evaluated points as defined above corresponds to distance ϵ in pixels, where

$$\epsilon = \begin{cases} \sqrt{1+k^2}, & \text{if } k < 1 \\ \sqrt{1+\frac{1}{k^2}}, & \text{if } k \geq 1 \end{cases}, \tag{3}$$

so that $1 \leq \epsilon \leq \sqrt{2}$. Thus the procedure of extracting brightness profile must also return the ϵ value along with profile itself to use it in final steps of determining distances. This is not necessary for profiles used for location of pipette walls, since by design only untilted profiles along pixel lines are used there, thus the scale in preserved.

1.1.6 Find positions of 3 characteristic points on the axis

We are interested in 3 points on the axis of the system — intersection of axis with outer vesicle, aspirated tip of a vesicle and the tip of the pipette. These manifest themselves as peaks (positive or negative) or steep jumps on the brightness profile along the axis. What exactly feature corresponds to what point of interest depends on the type of images one analyses.

- **Phase contrast images**

  For phase contrast images (see Figure 4), the pipette tip ought to be a local minima or (in case of too small pipette and/or too bright illumination) a global maxima. Vesicle sides in turn are featured as steep jumps from dark to bright (so the inflection point is a choice for a feature position).

  The pipette tip is first roughly located as the global maximum or minimum in the pipette tip search region, giving the position of the pipette

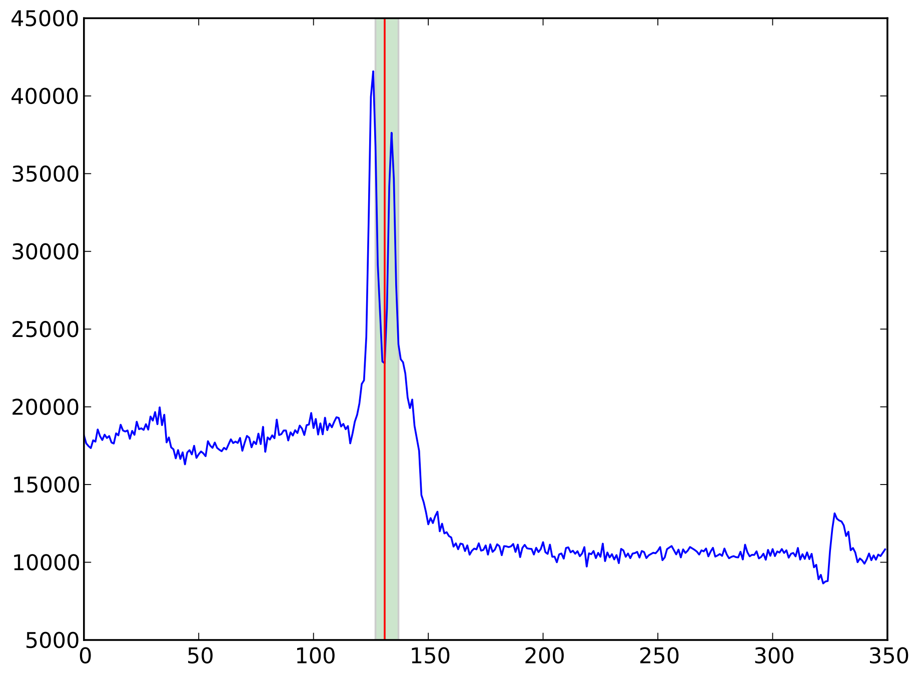


Figure 4: Brightness profile along the system axis for phase contrast image at Figure 3. Green region corresponds to the user-supplied approximate pipette tip position. Red line denotes actual pipette tip position found. Other features are also clearly distinguishable.

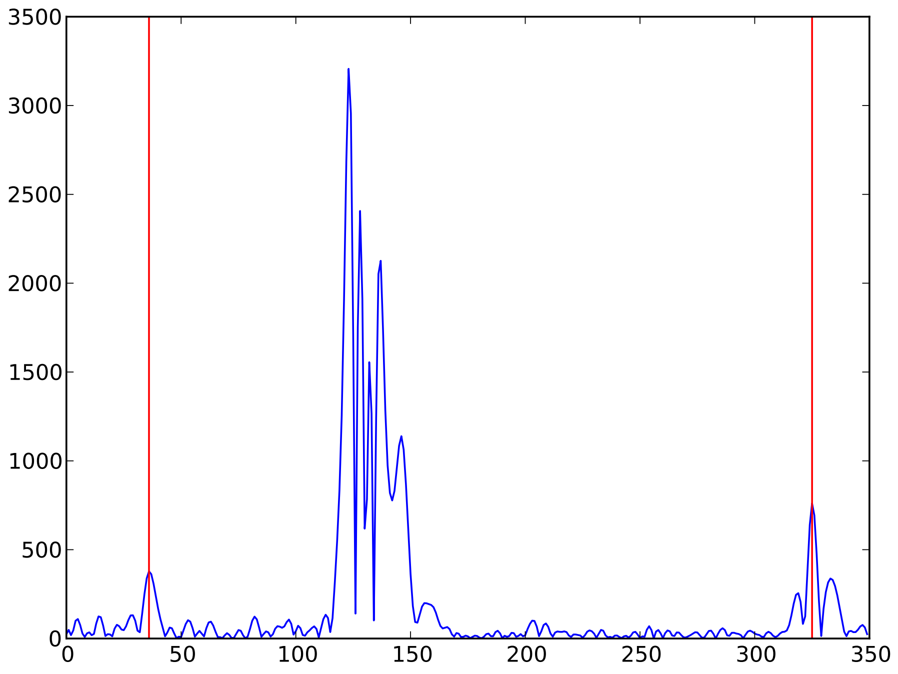


Figure 5: Result of applying gradient magnitude filter to gaussian-presmoothed profile from Figure 4. Red lines correspond to located features positions.

tip with *pixel resolution*. To get a *subpixel resolution* of peak position its vicinity is fitted with some suitable profile (see Section 1.1.6) with parameters of the fit giving the value desired.

As for inflection points, the rough positions are found by first applying a smoothing filter to the brightness profile, taking absolute value of smoothed profile's derivative (Savitzky-Golay [2, 3] and Gauss smoothing/derivative filters are implemented for now) and than using user input for overestimated feature positions (see Section 1.1.2) to define left and right search regions where global maxima for these regions are found, giving positions of steep jumps with *pixel resolution* - see Figure 5. For *subpixel resolution* the profile in the vicinity of estimated position must be fitted with suitable function with well-defined inflection point (see Section 1.1.6).

- **DIC images**

For DIC images the aspirated tip is local minimum and the outer edge is local maximum in their respective search regions (or vice versa, depending on choice of polarizations) and pipette tip is (as with phase contrast images) either a local minimum or global maximum in the pipette tip search region. The fitting procedure by itself is the same as for peaks in Section 1.1.6.

- **Subpixel resolution**

  NOT YET IMPLEMENTED

  Positions of features with subpixel resolutions can be found by making a non-linear least squares fit (`scipy.optimize.leastsq`, Levenberg-Marquardt implementation) of brightness profile in the vicinity of the pixel-resolved location of the feature using some suitable function. According to [4] brightness jumps on the edge of an object observed with phase contrast microscopy are described by integral sine function, with inflection point being the exact position of the object's edge. As for the peaks, for them the Gaussian bell function can be used with position of its maximum defining feature position with subpixel resolution.

1.1.7 Calculation of derived parameters

Given positions of outer vesicle edge as x_v, pipette tip x_p and aspirated vesicle tip x_t plus the pipette radius r together with their corresponding errors δx_v, δx_p, δx_t and δr and assuming total cylindrical symmetry of the system along the system axis and spherical form of the vesicle outside the pipette one can calculate the desired parameters (see Figure 6).

Denoting length of the vesicle part outside the pipette as $l_v = |x_p - x_v|$ with error $\delta l_v = \delta x_p + \delta x_v$ and the length of the aspirated tip as $l_a = |x_p - x_t|$ with error $\delta l_a = \delta x_p + \delta x_t$, the radius of the outside vesicle is

$$R = \frac{{l_v}^2 + r^2}{2l_v} \quad .$$

Corresponding error is then

$$\delta R = \sqrt{\frac{r^2}{{l_v}^2}(\delta r)^2 + \frac{1}{4}\left(1 - \frac{r^2}{{l_v}^2}\right)^2 (\delta l_v)^2} \quad .$$

The surface of the inner part is

$$S_{\text{in}} = \begin{cases} 2\pi r l_a, & \text{if } l_a \geq r \\ \pi\left({l_a}^2 + r^2\right), & \text{if } 2R - l_v \leq l_a < r \end{cases} \quad ,$$

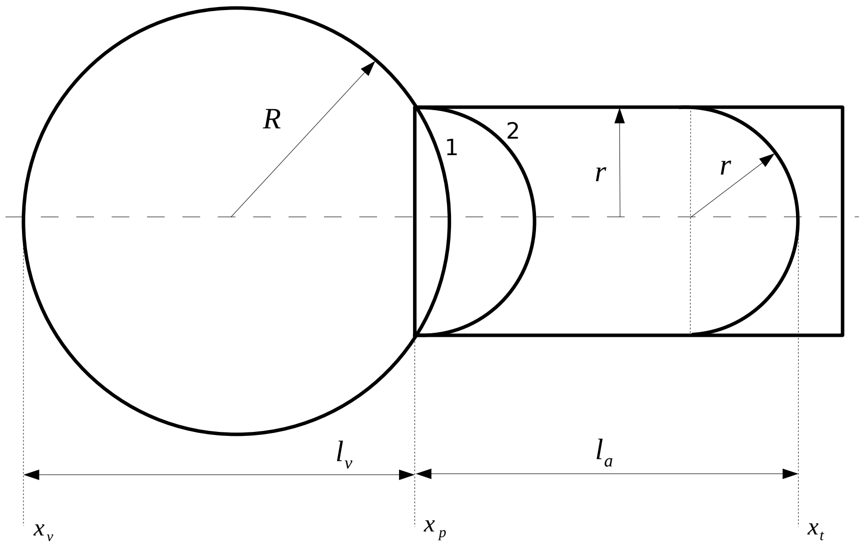


Figure 6: Sketch of a vesicle aspirated into the pipette with explanation of parameters.
1: $l_a = 2R - l_v$.
2: $l_a = r$

and its volume is

$$V_{\text{in}} = \begin{cases} \pi r^2 \left(l_a - \frac{r}{3}\right), & \text{if } l_a \geq r \\ \frac{1}{6}\pi l_a \left(3r^2 + {l_a}^2\right), & \text{if } 2R - l_v \leq l_a < r \end{cases} ,$$

with errors

$$\delta S_{\text{in}} = \begin{cases} 2\pi\sqrt{{l_a}^2 \left(\delta r\right)^2 + r^2 \left(\delta l_a\right)^2}, & \text{if } l_a \geq r \\ 2\pi\sqrt{{l_a}^2 \left(\delta l_a\right)^2 + r^2 \left(\delta r\right)^2}, & \text{if } 2R - l_v \leq l_a < r \end{cases}$$

and

$$\delta V_{\text{in}} = \begin{cases} \pi r\sqrt{r^2 \left(\delta l_a\right)^2 + \left(2l_a - r\right)^2 \left(\delta r\right)^2}, & \text{if } l_a \geq r \\ \frac{\pi}{2}\sqrt{4r^2{l_a}^2 \left(\delta r\right)^2 + \left(r^2 + {l_a}^2\right)^2 \left(\delta l_a\right)^2}, & \text{if } 2R - l_v \leq l_a < r \end{cases} .$$

The surface of the outer vesicle part is

$$S_{\text{out}} = 2\pi R l_v = \pi\left({l_v}^2 + r^2\right) \quad ,$$

and it's volume is

$$V_{\text{out}} = \frac{1}{6}\pi l_v \left(3r^2 + {l_v}^2\right)$$

with errors

$$\delta S_{\text{out}} = 2\pi\sqrt{{l_v}^2 \left(\delta l_v\right)^2 + r^2 \left(\delta r\right)^2}$$

and

$$\delta V_{\text{out}} = \frac{\pi}{2}\sqrt{4r^2{l_v}^2 \left(\delta r\right)^2 + \left(r^2 + {l_v}^2\right)^2 \left(\delta l_v\right)^2} \quad .$$

Total area and volume of the vesicle are then $S = S_{\text{out}} + S_{\text{in}}$ and $V = V_{\text{out}} + V_{\text{in}}$ with errors $\delta S = \delta S_{\text{out}} + \delta S_{\text{in}}$ and $\delta V = \delta V_{\text{out}} + \delta V_{\text{in}}$ respectively.

1.2 Data Analysis

1.2.1 Classical procedure

Having combined data acquired from image analysis together with corresponding pressure values obtained in experiment one finds elastic properties of membrane by fitting low-tension regime with logarithm and high-tension regime with linear functions. Also some corrections pointed out in [5, 6] could be accounted for here, by trying to fit the whole range of pressures with one general function.

Currently tensions are calculated according to simple model after Evans:

$$\tau = \frac{PR_p}{2\left(1 - \frac{R_p}{R_v}\right)} \tag{4}$$

Below is the list of currently implemented models/fittings.
Dilations:

- simplest model $\alpha = \frac{S-S_0}{S_0}$
- corrected for initial aspiration after [5]

$$\alpha = \left[\frac{1}{2}\left(\frac{R_p}{R_{v0}}\right)^2 \frac{\Delta L}{R_p} + \left[1 - \frac{3}{4}\left(\frac{R_p}{R_{v0}}\right)^3 \frac{\Delta L}{R_p}\right]^{\frac{2}{3}} - 1\right]\gamma \tag{5}$$

  where $\gamma = 1 - \dfrac{2R_pL_0 + R_p^2}{4R_{v0}^2}$.

- approximated correction for initial aspiration after [5] (made for testing)

$$\alpha = \frac{1}{2}\left[\left(\frac{R_p}{R_{v0}}\right)^2 - \left(\frac{Rp}{R_{v0}}\right)^3\right]\frac{\Delta L}{R_p}\gamma \tag{6}$$

  where γ as in equation 5.

Fittings (done with Orthogonal Distance Regression):

- Simple model for bending rigidity: $\alpha = \frac{1}{8\pi\kappa}\ln\frac{\tau}{\tau_0}$
- Simple model for stretching elasticity: $\alpha = \frac{\tau}{K} + \alpha_0$

1.2.2 Modern procedure

The overall ultimate goal is to implement a MCMC (Markov Chain Monte Carlo) algorithm found in [5].

2 User Guide

Being in active development, the state of the program might be quite different from what is written here!

As it is now (not packaged in self-contained executable) you need the following software to be installed on your computer to run VAMPy:

- Python 2.7.2 (`http://www.python.org`
- NumPy 1.6.1 (`http://www.scipy.org`
- SciPy 0.9 (`http://www.scipy.org`
- matplotlib 1.0.1 (`http://matplotlib.sourceforge.net/`)
- wxPython 2.8.12.1 (`http://www.wxpython.org`
- Python Image Library 1.1.7 (`www.pythonware.com/products/pil/`)

All of the above is Open Source and freely available from respective websites. Version numbers are the ones with which the software was developed and tested, but it might also work with older versions and hopefully will work with future versions of Python language and packages. The program is cross-platform, it runs on Windows (tested), Linux/Unix (tested) and Mac (not tested) as long as all dependencies are met.

2.1 What you can do with it

Warning: list of features is actively changed!

Currently you can:

- Open images of aspirated pipette and view them as stack of images.
- Crop the unnecessary parts of the image, rotate it as needed and save this settings for future reuse.
- Load corresponding pressure file or read pressures from image filenames.
- Analyze the geometry of the image and obtain plots for total area of the vesicle, its volume etc.
- Check suspicious points by analyzing single images to see whether the image analysis succeeded.

- Obtain the plot of tension vs dilation and corresponding value for the bending rigidity or stretching elasticity from fitting of data.
- Export results of geometry analysis and tensions calculations to ASCII file.
- Load manually measured geometry data and applied pressure from text files, calculate corresponding dilations and tensions and fit them for getting bending rigidity and stretching elasticity.
- As the above but load already manually calculated tensions and dilations from text file.

2.2 User Interface

- Main window (figure 7)

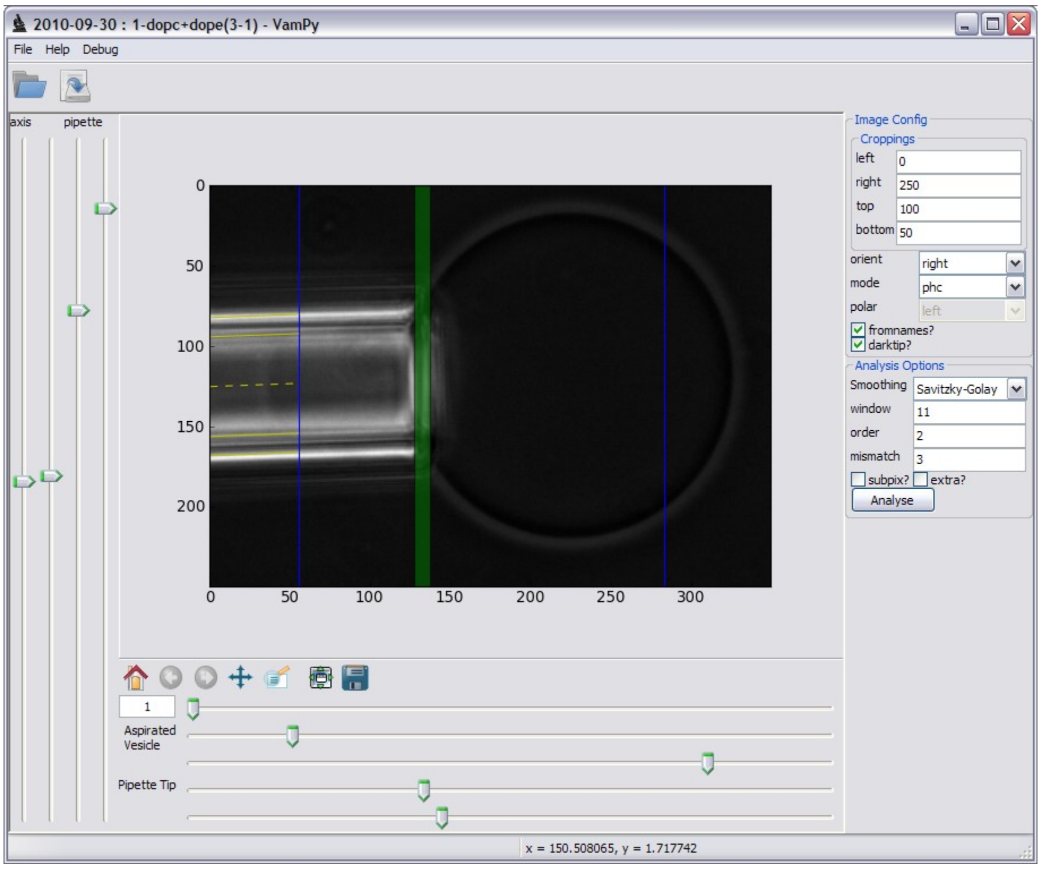

2010-09-30 : 1-dopc+dope(3-1) - VamPy

Image Config

Croppings

left	0
right	250
top	100
bottom	50
orient	right
mode	phc
polar	left

[✓] fromnames?
[✓] darktip?

Analysis Options

Smoothing	Savitzky-Golay
window	11
order	2
mismatch	3

[ ] subpix? [ ] extra?

x = 150.508065, y = 1.717742

Figure 7: VAMPy main window

1. Menu Bar - quite self-explanatory
   - File - open directory, exit application
   - Help - About info, Help (not implemened yet, shows the same info as "About"
   - Debug - Debug single image (for analysing image recognition performance), reload (for developers - reloads math modules without restarting GUI)
2. Tool Bar - contains buttons for some actions ("Open images folder" and "Save image config")
3. Image panel:
   (a) Image with region lines and pipette lines
   (b) Image toolbar (zoom, pan, adjust subplots, go back and forward between views, save plot as image - does not affect analysis, only representation)
   (c) Image slider - scrolls through the images
   (d) Region sliders - control region lines for aspirated tip, outer vesicle and pipete mouth
   (e) Axis sliders - control the estimation of the pipette axis position
   (f) Pipette sliders - control the estimation (radius and walls thickness) of pipette position
4. Image Config panel
   (a) Croppings - used to crop the image
   (b) Orientation - for rotation the image
   (c) Mode - type of the image (phase contrast or dic)
   (d) Polar - type of the dic image
   (e) fromnames - how to load pressures, from images filenames (checked) or from pressure file (unchecked)
   (f) darktip - what to count for pipette mouth, brightest(unchecked) or the darkest (checked) pixels
5. Analsis Options panel
   (a) Smoothing – type of applied smoothing (Savitzky-Golay or Gauss readily available)
   (b) window - full window size for the smoothing filter (odd integer since point plus same number of points to the left and to the right)

(c) order - used for smoothing the image while processing (polinomial order for Savitzky-Golay, blurring strength for Gauss)
(d) mismatch - the maximal difference between pixel and subpixel resolution to tolerate subpixel routine as successful *(not implemented yet, ignored)*
(e) extra - output extra information as NumPy saved arrays (NPZ) and python pickles *(not implemented yet, ignored)*
(f) subpixel - use subpixel resolution *(not implemented yet, ignored)*
(g) Analyse - start processing

6. Status Bar - when the cursor is in the image/plot shows current coordinates

- Geometry window (figure 8)

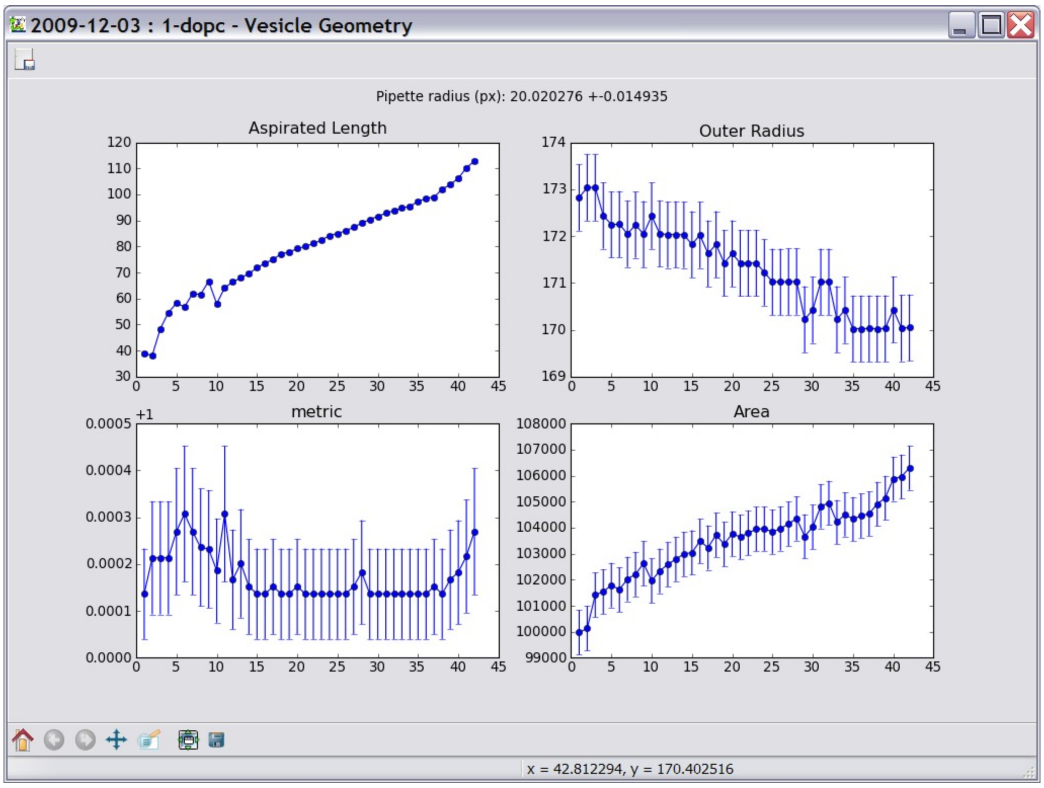
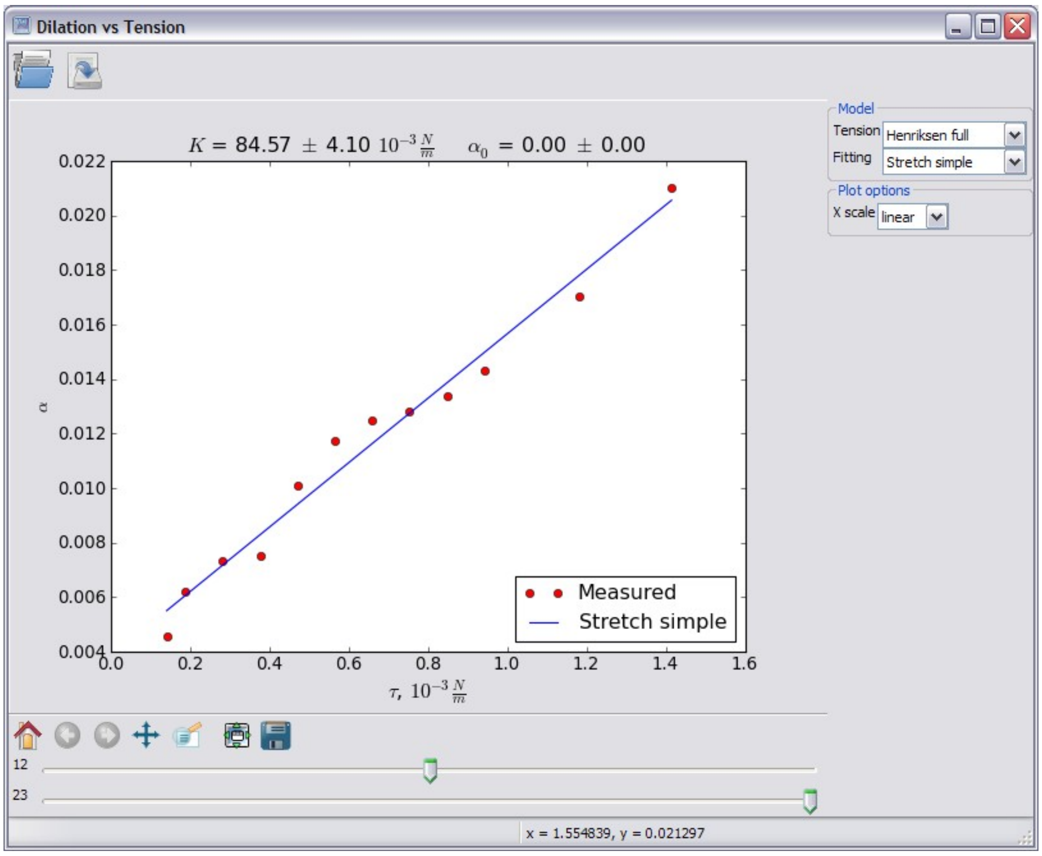

2009-12-03 : 1-dopc - Vesicle Geometry
Pipette radius (px): 20.020276 +-0.014935
Aspirated Length
Outer Radius
metric
Area
x = 42.812294, y = 170.402516

Figure 8: VAMPy vesicle geometry window

1. Toolbar - buttons to open external geometry file and to save all the calculated data to the ASCII file

2. Plots - plots of various calculated parameters - currently aspirated length, outer vesicle radius, vesicle total volume and metric (related to the tilt of the axis with 1 being perfectly horizontal). Calculated pipette inner radius is also displayed.
3. image toolbar - the same as on main window
4. status bar - the same as on main window

- Fitting window (figure 9)



K = 84.57 ± 4.10 $10^{-3}\frac{N}{m}$ α_0 = 0.00 ± 0.00

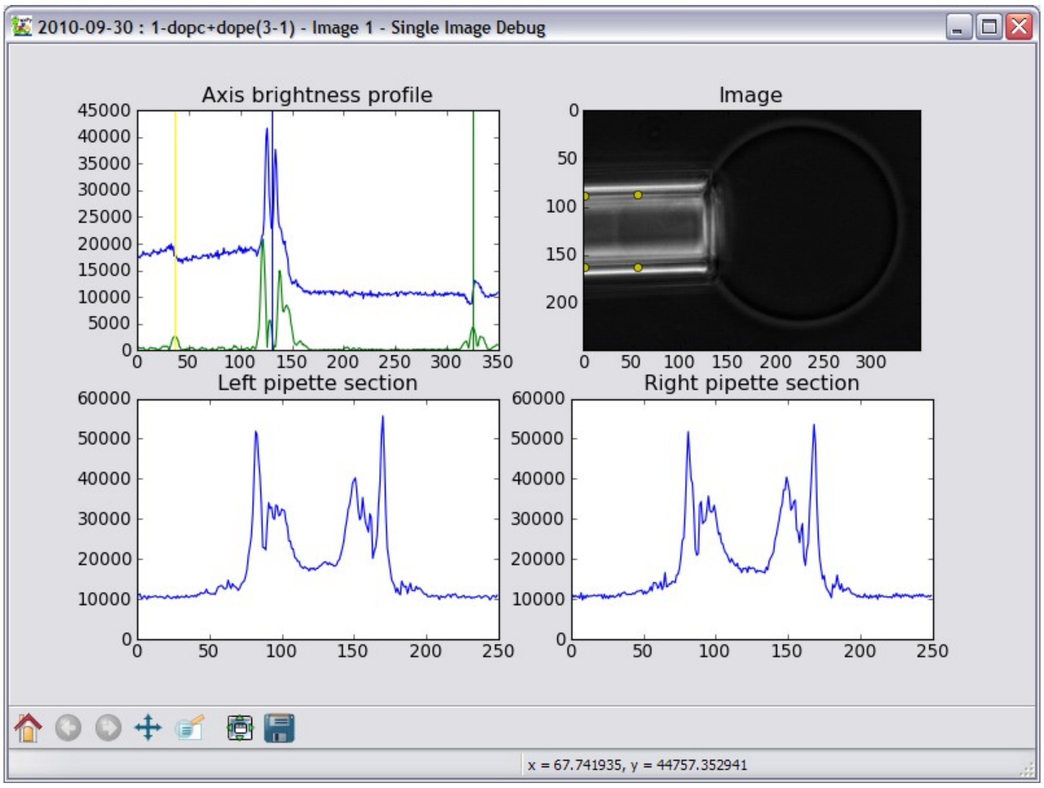
x = 1.554839, y = 0.021297

Figure 9: VAMPy fitting window

1. toolbar - buttons to open external file with tensions/dilations and to save calculated tensions and dilations to ASCII file (for example to analyse/fit it with other tools)
2. plot of the dilation vs tension - both experimental points and fitted line are shown, with fitted values printed on top
3. range slider - choose what range of values to fit

4. Model parameters
    (a) Tension model used for calculation of tensions/dilations
    (b) Model to fit data with
5. plot options - currently only choice for x coordinate as linear or log10
6. image toolbar - the same as on main window
7. status bar - the same as on main window

- Debug window (figure 10)


2010-09-30 : 1-dopc+dope(3-1) - Image 1 - Single Image Debug

x = 67.741935, y = 44757.352941

Figure 10: VAMPy single image debug window

1. plot of various values for the single chose image - currently axis brightness profile and absolute value of its smoothed derivative with found features positions, found positions of pipette walls, brightness profile across pipette used to find pipette walls
2. image toolbar - the same as on main window
3. status bar - the same as on main window

2.3 Work flow

1. If it is the first time you run the software on this particular computer, make sure that all dependencies are installed (see **??**).
2. Run the program by launching file “VamPy.pyw” (on Windows) or from command line ’python VamPy.pyw’ (Linux/Unix)
3. Open desired folder with images to analyze (Menu→File→Open... or Ctrl+O or Toolbar button). You will also be prompted for file type to load (currently only PNG or TIF images are supported).
4. If the folder contained a file “vampy.cfg”, the program will try to load image settings (cropping, rotation) and sliders positions from it.
5. Crop the image as desired - the smaller the image, the faster program works and less noise/artifacts it might encounter while analyzing images. You can use image slider to scroll through images and make sure that you do not crop out any significant parts. Sides of cropping are relative to original image
6. Rotate image so that *pipette sticks from the left side.* This is very important!
7. Choose the type of image you are analyzing - phase contrast or differential contrast. For latter, choose also “polar” parameter, which is the side from where a “virtual light” is shining in your picture - if the left part is bright and the right has “shadows”, choose “left” and vice versa.
8. Check the option ’fromnames’ if the pressures must be read from image filenames (as created with LabVAMP program). If not, the pressure file used for the experiment will be asked for
9. decide for what will be counted as pipette mouth - it can be either a most bright part in the region of the pipette mouth or the most dark one (’darktip’ unchecked or checked respectively) - ideally pipette tip produces a dark line, but with small pipette/bright illumination it might be blurred already beyond recognition, so you will have to define the pipette tip from the brightest pixels
10. Adjust region, axis and pipette sliders so that on every image in the stack:
    - Dashed line goes approximately along the pipette axis

- The darkest part of the pipette walls is always inside the estimation for pipette walls (yellow lines)
- The tip of the aspirated part is always to the left of the left vertical blue line
- The outer vesicle part crossing the pipette axis is always to the right of the right blue line
- The green region is as narrow as possible but always covering the pipette mouth

11. You can save image settings - crop, rotation, position of sliders etc to use them next time you open these images, just press “Save image info” button in the toolbar. Settings will be saved as a simple tab separated ASCII-file “vampy.cfg” in the folder from where current images were loaded.

12. Other parameters that affect the analysis:

    - “smoothing” - what algorithm to use for smoothing and derivating the brightness profile
    - “window”- full width of the window in the smoothong filters that need it, otherwise ignored
    - “order”- smoothing strength, has different meaning in different filters (for example polinomial order in Savitzky-Golay, strength of blur in Gaussian)
    - “Extra” - return a lot of additional info from image analysis and save them to external files (not fully implemented yet)
    - “Subpix” - check to have a fitting routines for subpixel resolution (not fully implemented yet).
    - “Mismatch” - determines when to discard the subpixel value for pixel value (as with subpixel resolution itself, not fully implemented yet).

13. Press the “Analyse” button. If you haven’t checked “fromnames” check box, you will be prompted for the location of the pressure protocol file.

14. The program automatically determines the situation when there are more than one image for each pressure and handles it correctly if all images are present, otherwise it produces an error message.

15. In any case in the next dialog you will also be prompted for the next parameters:

- Stage number - which column in the pressure protocol file (part of the name of the image file) to use as a pressures for these images.
- Scale factor - conversion from pixels to micrometers, can be measured with the microscope ruler. Default value is for TELI CS-3960DCL camera and overall magnification of 20X.
- Pressure accuracy - needed for the calculation of tension accuracy (and thus accuracy of output parameters of fitting), determined by the accuracy of the stages moving the water vessels. Default is for PI M-535.21 linear stage.

16. After some processing time two windows will pop out - “Vesicle Geometry” and “Dilation vs Tension”
17. “Vesicle Geometry” shows you values and plots of various extracted/calculated parameters already averaged between images corresponding to the same pressure, such as measured pipette diameter, total vesicle volume, area, outer vesicle radius etc .
18. “Dilation vs Tension” window presents the corresponding plot together with its fit with one of the chosen models/algorithms, with fitted value printed on top of the plot.
19. By using sliders you can control the region which is fitted.
20. You can also inspect how well the image recognition worked on any particular image by choosing “Debug$\rightarrow$Debug image” from the menu of the main window. The Debug window will open, allowing you to check the algorithm performance.

2.4 Limitations

Current problems/limitations/target for improvements:

- Not thoroughly tested for dic-images
- No subpixel resolution for single images
- No extra data available to user

References

[1] Eric Jones, Travis Oliphant, Pearu Peterson, et al. Scipy: Open source scientific tools for python, 2001–.

[2] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[3] Jean Steinier, Yves Termonia, and Jules Deltour. Smoothing and differentiation of data by simplified least square procedure. *Analytical Chemistry*, 44(11):1906–1909, 1972.

[4] A. Bitler, A. Barbul, and R. Korenstein. Detection of movement at the erythrocyte's edge by scanning phase contrast microscopy. *Journal of Microscopy*, 193(2):171–178, 1999.

[5] J. R. Henriksen and J. H. Ipsen. Measurement of membrane elasticity by micro-pipette aspiration. *The European Physical Journal E - Soft Matter*, 14(2):149–167, June 2004.

[6] J.-B. Fournier, A. Ajdari, and L. Peliti. Effective-area elasticity and tension of micromanipulated membranes. *Phys. Rev. Lett.*, 86(21):4970–4973, May 2001.