A Guide to Elastica2D

Brian W. Quist*, Lucie A. Huet*, Hayley M. Belli, and Mitra J. Z. Hartmann

*Authors contributed equally

Associated Publication:

Quist BW and Hartmann MJZ (2012) Mechanical signals at the base of a rat vibrissa: the effect of intrinsic vibrissa curvature and implications for tactile exploration. *J. Neurophys*. **107**:2298-312.

Table of Contents:

1. Introduction

Elastica2D is a numerical model that simulates the bending of a rodent whisker (or any other thin elastic beam) due to a point force. It outputs the shape of the deflected whisker, the mechanical signals generated at the base of the whisker, and other mechanical data.

The model relies on a few important underlying assumptions:

- *Quasi-static* – The deformations are steady-state. There are no inertial effects due to mass or damping. All bending depends only on steady-state forces and bending stiffness.

- *Frictionless* – The applied force remains perpendicular to the tangent of the whisker at the point of contact. Other versions of the model have incorporated friction (Solomon and Hartmann, 2008).

- *2D* – The model exists in two spatial dimensions. Elastica3D (in progress) will incorporate the third spatial dimension.

- *Equidistant nodes* – Given the particular way that Elastica2D handles bending, accuracy is improved if all the nodes are equally spaced along the arc length of the whisker. By default, Elastica2D contains a subroutine that ensures that the nodes are equally spaced. The subroutine can be turned off if desired.

All calculations take place in whisker-centered coordinates, where the base of the whisker is located at the origin (0,0), and the initial linear portion of the vibrissa is aligned with the x-axis (Figure 1). The user must enter the whisker coordinates consistent with these conventions, and Elastica2D outputs the deflected whisker shape and the forces at the base of the whisker in these coordinates.

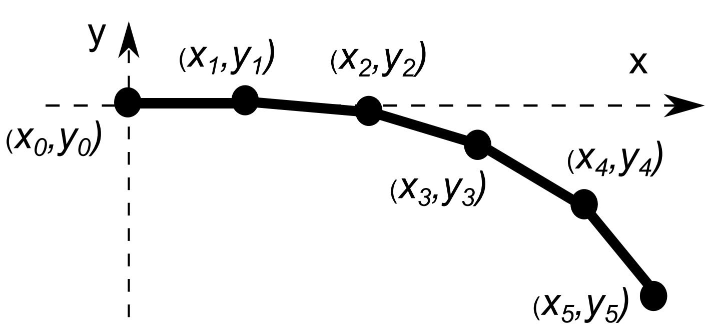

Figure 1: An example of a six-node whisker in whisker-centered coordinates. Note that the second node also lies on the x-axis so that the initial linear portion of the whisker is aligned with the x-axis.

There are two methods or modes in Elastica2D. The first and most basic mode, called the “Force” mode, simply applies a certain force at a certain arc length out along the whisker. The whisker bends, and Elastica2D returns the shape of the deflected whisker and other data (Figure 2a).

The second mode, called the “Point” mode, is likely the more useful of the two modes and remains the default mode in Elastica2D. In this mode, the user defines a contact point in space (a, b), and the model bends the whisker until it touches this point, supplying the force magnitude and arc length of application on its own (Figure 2b). It gives the same outputs as the Force mode.

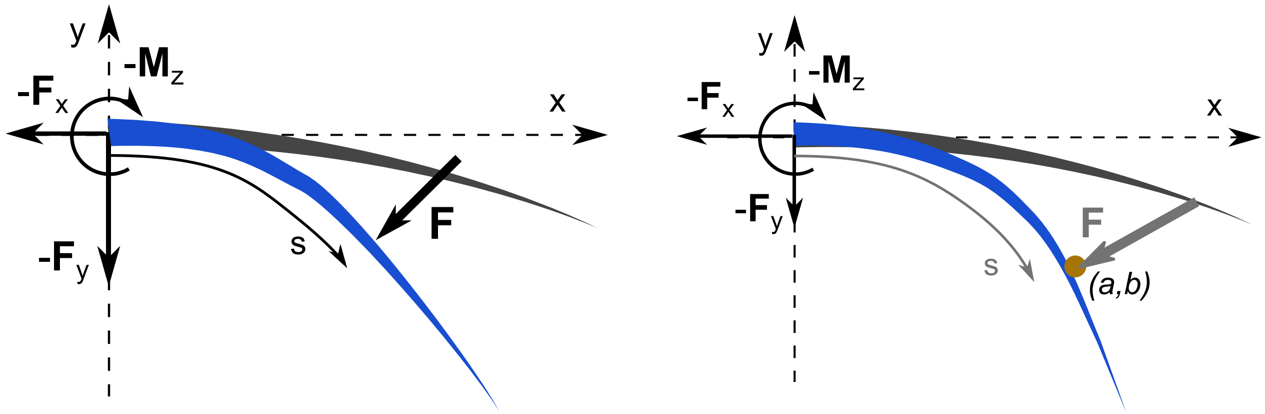

Figure 2: (a) An example of a whisker bending using force mode in elastica2D. The dark gray line is the undeflected whisker before the force is applied. The blue line is the deflected whisker. The output forces and moment at the base are also shown. (b) An example of a whisker bending using point mode in elastica2D. The thick black line is the undeflected whisker, and the blue line is the whisker that has been deflected to the contact point (a, b) input by the user. The force (F) and arc length of application (s) have been found by the algorithm.

It is important to note that the model must remain in whisker-centered coordinates. Thus, as shown in Figure 3b, in order to simulate the whisker rotating against an object, the user should rotate the contact point through an angle of θ_{push} degrees instead of rotating the whisker.

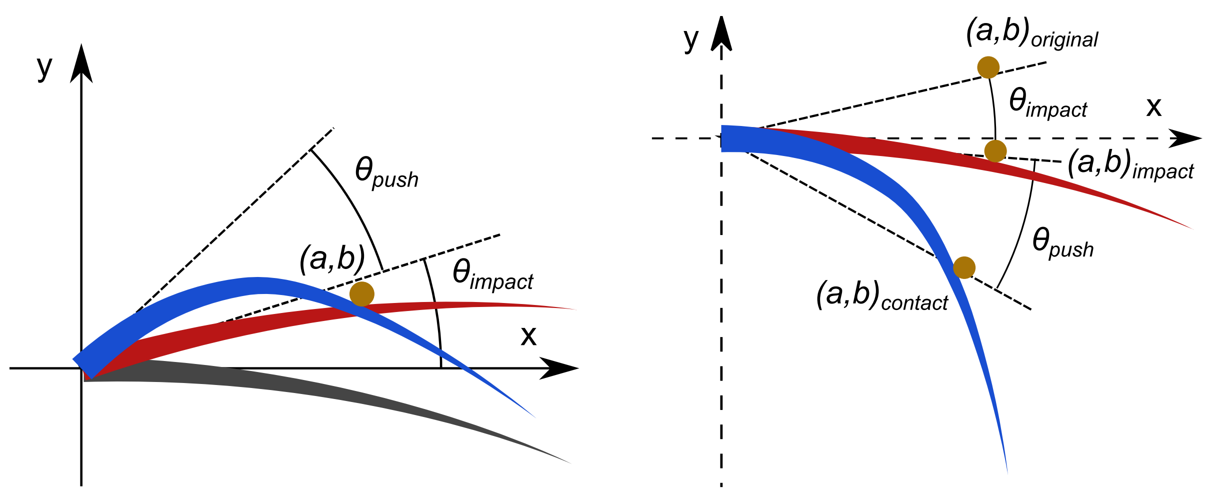

Figure 3: a) In world coordinates, the whisker is initially aligned with the x-axis. It rotates counterclockwise until it hits an object at θ_{impact} and then rotates θ_{push} more. b) The same motion, but drawn in whisker-centered coordinates. The contact point (a, b) rotates clockwise. This is how this simulation should be entered in elastica2D.

2. Mechanics Background

Implementation of elastica2D requires a basic understanding of the following mechanical principles:

1. Moment (or torque) is the tendency of an object to rotate about an axis. The moment is a vector and is mathematically represented as the cross product of the (vector) force with the (vector) radial distance measured from the point of rotation, i.e. M = r x F.
2. Area Moment of Inertia is a geometric quantity that determines the resistance of a beam to deflection about an axis. In the context of elastica2D which assumes a circular whisker cross-section, this parameter is given by the equation: $\frac{\pi}{4}r^4$, where r is the radius of the beam at each individual node along the whisker.
3. Young's Modulus: a material property of the whisker that determines its stiffness. The higher the value of the Young's modulus, the stiffer the material. Stiffness is defined as the product of Young's modulus and the area moment of inertia.

The computational algorithm for Elastica2D uses the following two relationships:

$$\mathrm{dk} = \frac{\mathrm{M}}{\mathrm{EI}} \qquad (1)$$

$$\mathrm{dk} = \frac{\mathrm{d\theta}}{\mathrm{ds}} \qquad (2)$$

In these equations, M, I, and E represent the three variables moment, area moment of inertia, and Young's modulus described above. dk represents the change in whisker curvature, ds the change in arc length, and $\mathrm{d\theta}$ the change in angle between two nodes. Rewriting equation (2) as $\mathrm{d\theta} = \mathrm{ds} * \mathrm{dk}$ and plugging into equation (1) yields:

$$\mathrm{d\theta} = \mathrm{ds}\frac{\mathrm{M}}{\mathrm{EI}} \qquad (3)$$

Equation (3) is the step that Elastica2D uses for evaluating the change in angle at each individual node in order to find the corresponding bending moment and forces at the base of the whisker.

For more details regarding the mechanical theory of elastica2D, see Birdwell, et. al. 2007.

3. Inputs and Outputs for the two modes of operation

Elastica2D has two modes of operation depending on which parameters the experimenter knows. In both modes of operation, the shape of the undeflected (resting) whisker must be known.

<u>FORCE MODE:</u>

In "Force" mode, the experimenter applies a force with known magnitude at a certain arc length (s) out along the whisker. Under the assumption of no friction, the direction of the force remains normal to the whisker at the point of application. Elastica2D then computes the shape of the deflected whisker and all 2D components of force and moment at the base.

The basic function call looks like this:

[x, y, P] = elastica2D(x1, y1, [F s], 'mode', 'force');

A breakdown of the basic inputs:

- x1 : vector containing the x-coordinates of all the nodes of the undeflected whisker [m]
- y1 : vector containing the y-coordinates of all the nodes of the undeflected whisker [m]

- F : Magnitude of the applied force [N]
- s : Arc length along the whisker at which the force (F) acts [m]

These two are combined into a vector and entered together as one input.

- 'force' (after the defining parameter name 'mode') : This indicates to elastic2D that the Force mode is being used; it is necessary since the Point mode is the default.

Outputs:

- x : vector containing all the x-coordinates of the nodes of the deflected whisker [m]
- y : vector containing all the y-coordinates of the nodes of the deflected whisker [m]
- P : struct containing mechanical signals at the base and other data in the following fields:
  - .E : A vector of the Young's modulus at each node of the whisker (If the user defined Young's modulus as an optional input, it will be the same as the original user input.) [Pa]
  - .I : A vector of the area moment of inertia at each node of the whisker (If the user defined the area moment of inertia as an optional input, it will be the same as the original user input.) [m^4]
  - .fx : The force at the base in the x-direction [N]
  - .fy : The force at the base in the y-direction [N]
  - .m : The moment at every node of the whisker [N•m]
  - .dk : A vector displaying the change in curvature at every node of the whisker [1/m]
  - .k : A vector displaying the total curvature at every node of the whisker [1/m]
  - .k_global : The global curvature of the whisker from the base to the point of contact [1/m]
  - .fs : A vector [F s] containing: F) the magnitude of the force output (which will match the F input in force mode), and s) the arc length at which the force was applied (which will match the s input in force mode)
  - .gamma : The angle of the whisker at the point of contact [rad]
  - .contact : A vector [x_{cp} y_{cp}] containing the final point in space where the force is acting on the whisker (in point mode, this vector should match the contact point input) [m]

POINT MODE:

In "point" mode, the coordinates of the contact point (a, b) are known. The code searches for an applied force (F) and arc length at which the force acts (s), and it has found the correct F and s when the final location at which the force is acting on the whisker (x_{cp}, y_{cp}) is the same as the location of the contact point (a, b). The algorithm then returns the deflected whisker shape and all the 2D components of force and moment at the base.

The basic function call looks like this:

```
[x, y, P] = elastica2D(x1, y1, [a b]);
```

OR

```
[x, y, P] = elastica2D(x1, y1, [a b], 'mode', 'point');
```

A breakdown of the basic inputs:

- x1 : vector containing the x-coordinates of all the nodes of the undeflected whisker [m]
- y1 : vector containing the y-coordinates of all the nodes of the undeflected whisker [m]

- a : the x-coordinate of the contact point [m]
- b : the y-coordinate of the contact point [m]

These two are combined into a vector and entered together as one input.

- 'point' (after the defining parameter name 'mode') : This indicates to elastic2D that the point mode is being used, but this input is not necessary since point mode is the default.

Optional inputs:

In both modes, there are a number of ways to modify the whisker you are simulating or even how the model runs. These fall under the variable input arguments in MATLAB and follow the standard format of {'parameter_name', parameter}. Defining the 'mode' parameter as 'point' or 'force' is one such example of these.

Inputs to modify the whisker:

- 'E' : Change the Young's modulus of the whisker. The default value is 3.3 GPa. The parameter entered can either be a scalar, in which the every node of the whisker has the same Young's modulus, or the parameter can be a vector the length of the number of nodes of the whisker, defining a different Young's modulus at each node. [Pa]

- 'I' : Change the area moment of inertia of the whisker. The default values are calculated assuming the whisker has a circular cross section. The default whisker has a radius of 100 micrometers at the base, and the radius at the tip is 1/15 of the radius of the base. Like the 'E' parameter, this parameter can be input as either a scalar or a vector the same length as the number of nodes. A scalar input would give the whisker the same area moment of inertia at every node and imply a cylindrical instead of tapered whisker. [m^4]

- ‘BC’ : Change the boundary condition at the base of the whisker. The default is ‘r’ for rigid, which means the node at the base does not rotate at all. Entering ‘e’ for elastic boundary condition means the base would bend using the defined ‘E’ and ‘I’ at the base. Last, a scalar value can be entered; in this case, the boundary would be elastic and use the input scalar value as the E*I value at the base. [Pa• m^4]

Other optional inputs:

- ‘plot’ : Toggle this to 0 (default) or 1 to turn the final plot on or off. This plots the undeflected whisker shape and the deflected whisker and notes the contact point (x_{cp}, y_{cp}).

- ‘fs_guess’ : In point mode, Elastica2D calculates an initial F and s guess on its own, but if the user has a better guess, she or he can input it here in the same [F s] input manner. [N m]

More optional inputs can be found described in the body of the elastica2D code.

4. Example of Elastic2D Implementation via Point Mode:

Suppose we have an intrinsically curved, discretized whisker with 100 nodes, arc length of 60.3mm, base diameter of 199 microns, tip of radius 1.5 micrometers, an average Young's Modulus of 3.75, and constant curvature coefficient of -20. We want the whisker to make contact with a peg located at (15mm, 6mm). Implementation in MATLAB would proceed as follows:

```
n = 100; %number of nodes along the whisker
L = 0.0603; %arc length in meters
k = -20; %curvature coefficient for shape of whisker
E = 3.75*10^9; %Young's modulus in Pascals
a = 0.015; %x-coordinate of contact point in meters
b = 0.006; %y-coordinate of contact point in meters
```

Next we calculate our non-default area moment of inertia:

```
rbase = 0.000199/2; %diameter divided by 2 for radius in meters
rtip = 0.0000015; %tip radius in meters
r = linspace(rbase, rtip, n);
I = pi/4*r.^4;
```

To provide the shape of the curved whisker, we must call the following external function, which uses inputs arc length, curvature coefficient, and number of nodes to find x and y-coordinates for the nodes equally distributed along the x-axis. Note this is different from the elastica2D subroutine that redistributes the nodes evenly along the arc length. This function is necessary when the whisker has intrinsic curvature and can be omitted for straight beams:

```
[x1, y1] = sa2xy(L, k, n);
```

Finally, we have all necessary inputs to both call elastica2D and plot the deflected whisker:

```
[x, y, P] = elastica2D(x1, y1, [a b], 'E', E, 'I', I, 'plot', 1);
```

For this example, the basic output would appear as follows, where the blue curve is the original shape of our whisker, the red circle is our contact point, and the black curve is our deflected whisker:

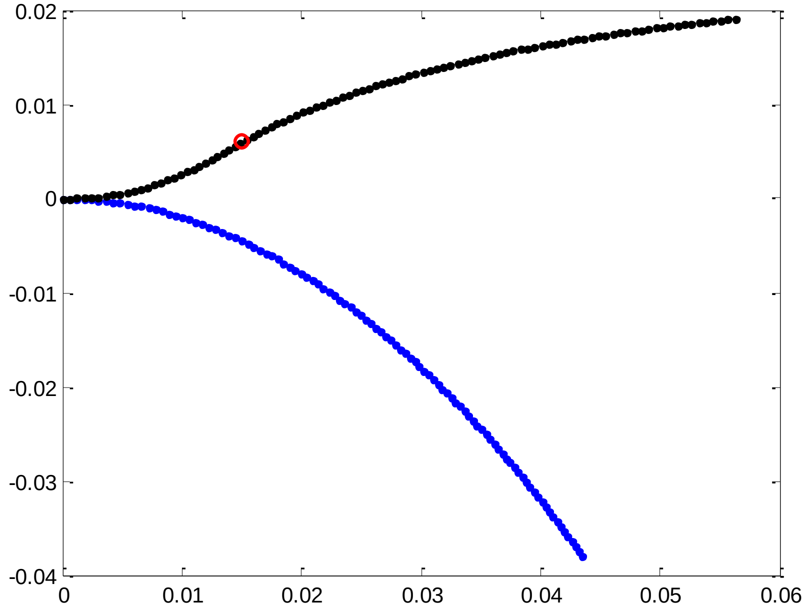

Figure 4: An example output where the blue curve is the original shape of the whisker, the red circle is the contact point, and the black curve is the deflected whisker.

5. Example with Experimental Data from a Plucked Whisker

Suppose we want to quantify the changes in forces and moment at the base of a whisker as it rotates against a peg. We demonstrate how to achieve this task using experimental data from a plucked whisker in conjunction with Elastica2D.

This example follows the general procedure used to generate Figure 13 in Quist and Hartmann (2012). A whisker from an anesthetized rat was plucked and mounted on a DC motor, and high resolution images were captured of the whisker rotating against a rigid post.

Step 1. Extract the resting whisker shape (x1, y1) and contact point (a, b) and place both in a whisker-centered coordinate system. This may require both translation and rotation depending on the image.

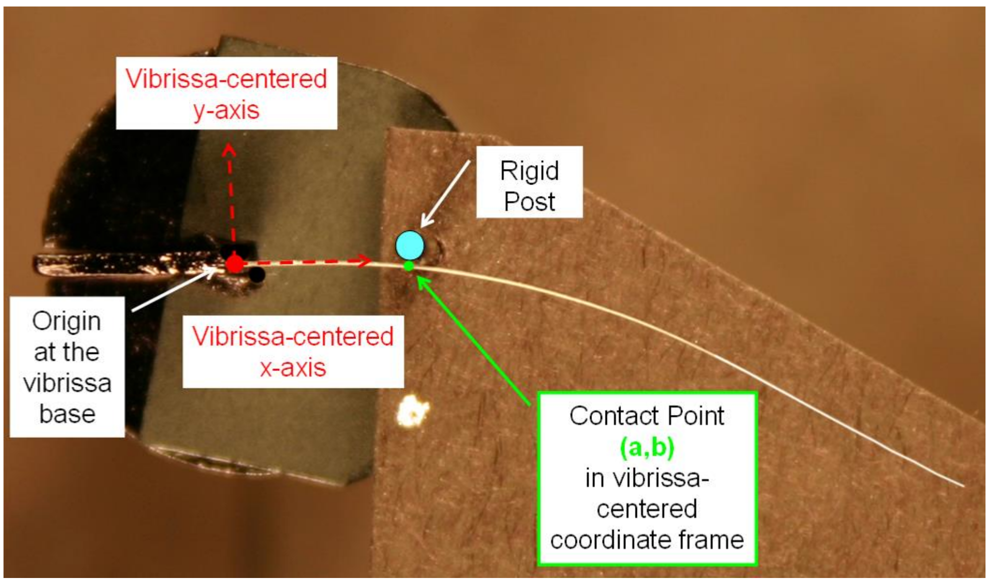

Figure 5: Raw image date/resting whisker shape

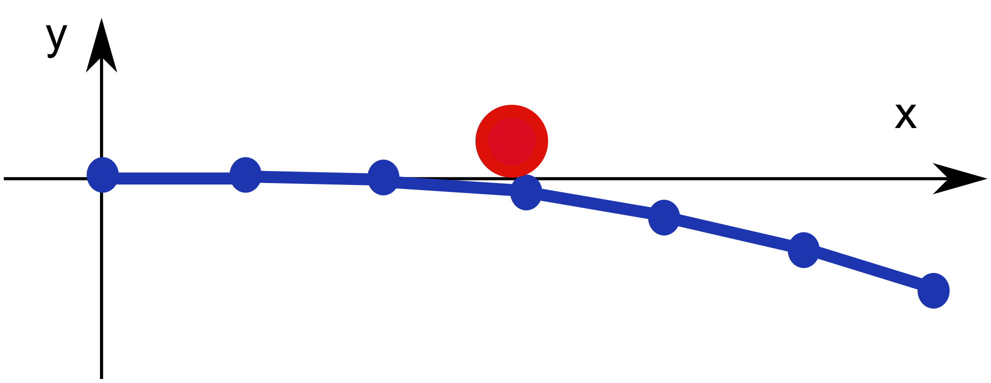

Figure 6: Extracted image data/resting whisker shape

Step 2. In the physical experiment, rotate the whisker counterclockwise until it first touches the contact point. We refer to this angle as θ_{impact}. Extract the location of the contact point and the location and orientation of the whisker base. Use the base location and orientation to position the new contact point in whisker-centered coordinates. Enter this contact point and the previously extracted whisker shape into Elastica2D to begin computing moment and forces at the base. Repeat this process by experimentally incrementing θ, extracting the whisker base and contact point, and repositioning and orientating the contact point in whisker-centered coordinates.

Following this method, Elastica2D provides experimentalists with a representation of the quasi-static changes in moment and forces at the base of the whisker throughout the whisking cycle.

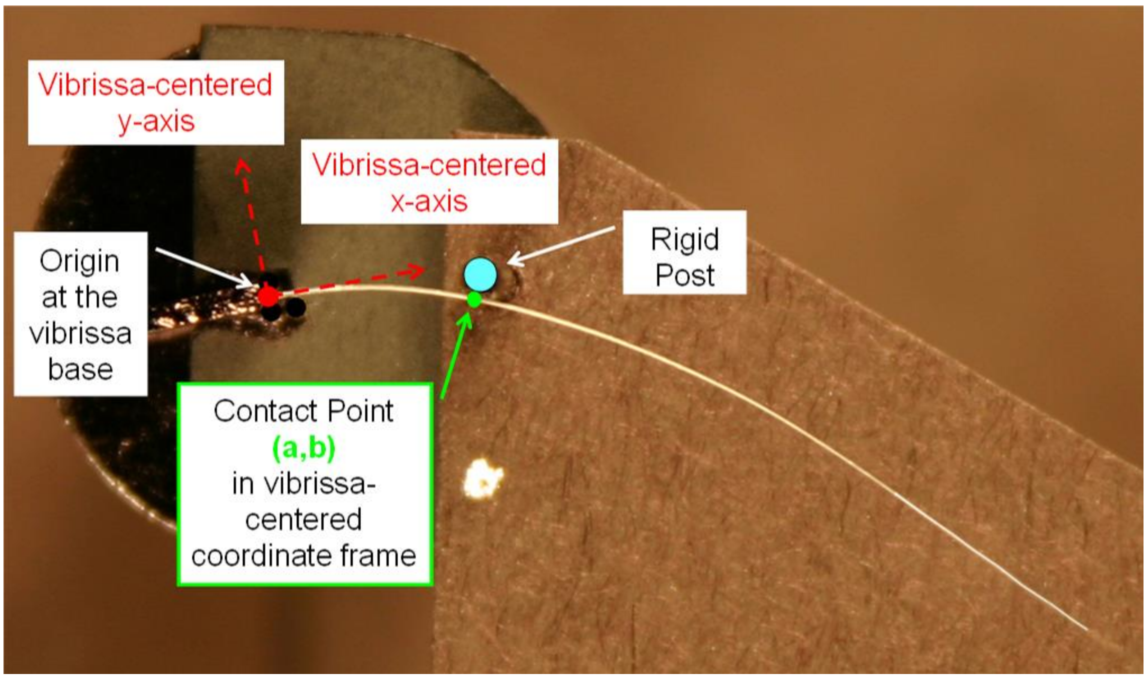

Figure 7: Raw image data/ 10º Rotation of the whisker into the object

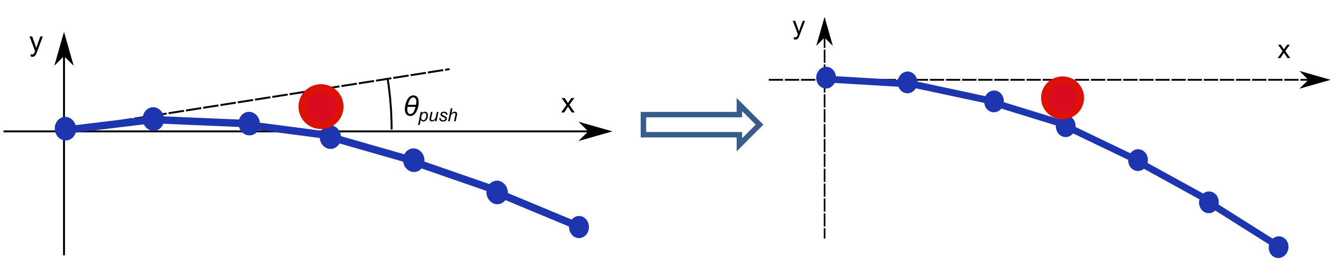

Figure 8: Extracted image data and corresponding whisker centered coordinate translation and rotation

6. References

Birdwell JA, Solomon JH, Thajchayapong M, Taylor MA, Cheely M, Towal RB, Conradt J and Hartmann MJZ. Biomechanical models for radial distance determination by the rat vibrissal system. *Journal of Neurophysiology* 98**:** 2439-2455, 2007.

Knutsen PM, Biess A, and Ahissar E. Vibrissal kinematics in 3D: Tight coupling of azimuth, elevation, and torsion across different whisking modes. *Neuron* 59: 35-42, 2008.

Quist BW, and Hartmann MJZ. A two-dimensional force sensor in the millinewton range for measuring vibrissal contacts. *J Neuroscience Methods* 172: 158-167, 2008.

Quist BW, and Hartmann MJZ. Mechanical signals at the base of a rat vibrissa: the effect of intrinsic vibrissa curvature and implications for tactile exploration. *Journal of Neurophysiology* 107: 2298-2312, 2012.

Solomon JH, and Hartmann MJZ. Artificial Whiskers Suitable for Array Implementation: Accounting for Lateral Slip and Surface Friction. *IEEE Transactions on Robotics* 24: 1157-1167, 2008.

Towal* RB, Quist* BW, Gopal V, Solomon JH, and Hartmann MJ. The morphology of the rat vibrissal array: a model for quantifying spatiotemporal patterns of whisker-object contact. *PLoS Comput Biol* 7: e1001120, 2011. * Authors contributed equally