Whisker detection as a shortest path problem

Igor Perkon[1,2,3], Andrej Košir[3], Jurij Tasič[3] and Mathew Diamond[1,2]
[1] Cognitive Neuroscience Sector, International School for Advanced Studies, Trieste, Italy
[2] Italian Institute of Technology – SISSA Unit, Trieste, Italy
[3] LDOS, Faculty Of Electrical Engineering, University Of Ljubljana, Slovenia
igor.perkon@gmail.com

Abstract

In an attempt to understand how the brain encodes tactile stimuli, neuroscientists commonly investigate how rodents translate whisker motion into a sensory percept. Analysis of whisker dynamics is a challenging task. Rats, have about 30 whiskers per side and in single-viewing angle experiments, occlusion and intersection occur frequently.

In this paper we present an unsupervised method for whisker detection. Based on the shortest path optimisation criterion, the method is designed for detecting paths that are smooth, continuous, and connected to the head contour.

1. Introduction

Research aimed at understanding how the brain encodes tactile stimuli is often performed on rats and mice [1]. Rodents, in their whiskers, have one of the most advanced tactile sensory systems. Unlike many other whiskered mammals, they use their vibrissae actively. By contracting the muscles in the whisker follicle, they produce an oscillatory motion to sweep through the air and to palpate objects. In order to characterize their tactile system, neuroscientists search for correlates between inputs and outputs. Outputs are obtained by recording neuronal signals from specific brain areas, while the inputs are extracted from the whisker dynamics.

Whisker dynamics can be captured using high-speed cameras with a frame rate typically set to 500 or 1000 fps. Since every experiment trial lasts 2 seconds and several hundreds of trials are collected per animal per day, there is an urgent need for automated video analysis tools.

Several tools have been developed to address this problem. In [2] the authors proposed a semi-automated method where the user selects manually the initial whisker position. The selected points are interpolated with a spline serving as the whisker model. Every subsequent frame is filtered with oriented filters tuned by the model position and orientation. The model then generates a subset of splines capturing its possible deformations and translations. The spline that best covers the filtered pixels is selected for updating the model. A completely automated method was presented in [3], where in each frame individual whiskers are detected by joining paths arising from grouping of pixels with similar orientations and with a high intensity gradient. Once the individual paths are fitted into a spline, the tracking is performed computing a similarity measure between splines in contiguous frames and thus grouping them into streaks. In [4] whiskers are detected by iteratively overlapping short segments using the criteria of minimum image intensity in the overlapped section. Then whiskers are tracked using the nearest neighbour rule. A similar method was presented in [5], where the authors published only their detection algorithm. After the detection of starting sites based on local line orientation and intensity variance, the authors detect whiskers with an iterative line following approach.

These tools address in general two problems: the detection of whiskers and their tracking. The extensive literature of contour tracking methods (for a survey see [6]) shows that accurate contour initialization is crucial for multiple object tracking algorithms. For this reason we present an unsupervised method for whisker detection designed for detecting paths that are smooth, continuous, and connected to the head contour.

2. Problem statement

Unlike many natural objects that exhibit significant differences in texture or colour, vibrissae can be characterized only by their shape and position. The whisking motion is contained mainly in a plane parallel to the axis from eye to the tip of the snout, so the top view is the preferred camera position in single view experiments. Whiskers grow out of the snout pad organized in 5 rows with about 6 whiskers per row, making a total of about 30 whiskers per side. The whisking cycles between separate rows are sometimes, but not always, in phase so the detection method must address intersections and partial occlusions (Figure 1) in experiments where whiskers are left full-length.

The expected output of an automated video analysis tool varies according to the investigated hypothesis. Typically it can be the whisking angle, which requires only the detection of a short part of the whisker near the base, or the entire shape, if more complex features like curvature variations along the shaft are of interest.

The method we present is suitable for detecting the whole shape. It is designed to provide a spline as output function in order to facilitate the computation of relevant geometric parameters.

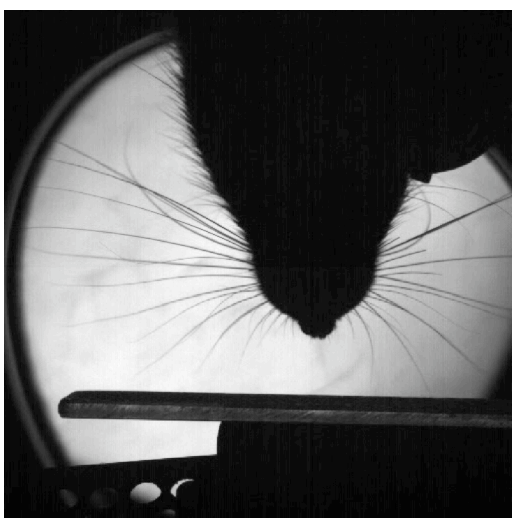

Figure 1. Input image of an exploring rat

3. Whisker detection

Whisker detection is essentially a contour extraction problem, where in order to reduce false positives, three vibrissal characteristics must be considered. Vibrissae extend radially from the head, so if we express their shape in polar head-centered coordinates, they can be correctly modeled as 1-D curves with radius acting as curve parameter; vibrissae are smooth, so low order continuous 1-D functions can be used to interpolate them correctly, and they grow out of the snout, thus among the detected curves, only those connected to the snout profile are detected as whiskers.

Therefore we formulate the whisker extraction as a shortest path problem on a graph [7], where vertices and edges represent detected points and their connectivity, points on the snout contour constitute the sink nodes and the detected line tips are the source nodes. In this formulation, valid whiskers are detected as minimum cost paths between the source nodes and the sink nodes. In the following sections we will explain how we preprocess the image, build the graphs and extract whiskers.

3.1 Image preprocessing

In our approach, whisker detection starts with background subtraction. On average the rodent is present during the entire duration of the movie, so instead of assuming the first frame of the sequence as the background image, we need to extract it automatically. Since our lighting and camera are set up to film the silhouette of the rodent, we extract the background image by selecting for every pixel its brightest value in the whole movie. After the background subtraction, we apply histogram equalization in order to normalize the intensity range of image differences in a standard interval. On the equalized image we apply tensor voting [8] for de-noising and line enhancement. In essence the tensor voting method converts every pixel to a matrix and, by applying a convolution with a bell-shaped kernel, determines the local features orientations and saliency.

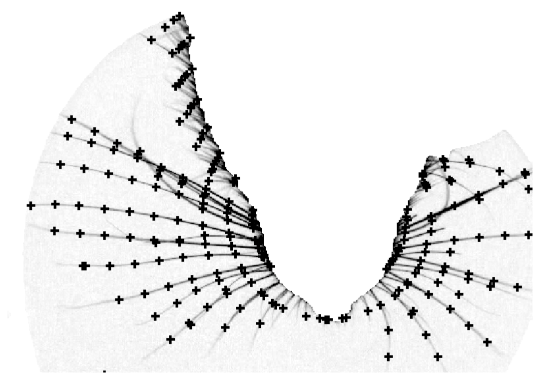

Figure 2. Preprocessed image with a subset of intensity peaks sampled on circular paths and marked with '+'

In a second step the method filters the image with an oriented Gaussian-shaped kernel and enhances the line features. The resulting image, pictured in Figure 2, shows the whiskers enhanced by the preprocessing step. The image serves as starting point for whisker detection.

3.2 Graph building

In the preprocessed image, whiskers appear as dark lines on a light-grey background, with their thickness tapering with distance from the snout. In our test movies the width of whiskers ranges between 1 and 4 pixels, so we mark the shaft position by applying intensity peak detection on circular paths centered in the head. Circular paths provide a good approximation to shaft perpendicular paths and they can be efficiently computed using constant radius paths in polar coordinates. Considering that vibrissae are smooth curves, we do not need to mark all intensity peaks in the image. We sample the image using equidistant circular paths, where the graph vertices represent detected peaks, while their spatial connectivity is encoded using graph edges. In this way we drastically reduce the number of vertices in the graph.

Every vertex is characterized by its position and its local line orientation (derived in the tensor-voting step). We connect two vertices with a unitary weight edge if their distance is under a predefined threshold and if the two absolute differences between the candidate edge orientation and the line orientation of its vertices are under a predefined threshold, which was usually set to 15°. This double condition avoids the creation of edges between whiskers that run parallel to each other (Figure 3).

Since the radius of the vertices increases with the distance from the snout, we can define ingoing and outgoing edges. If an edge connects two points with different radius, it is defined as outgoing for the vertex with greater radius and incoming for the vertex with the smaller radius. In the rare event that two vertices have the same radius, they are connected with a bi-directional edge.

3.3 Whiskers as paths in the graph

The resulting graph edge matrix shows that most of vertices have more than one ingoing and one outgoing connection. Our oriented graph paths stretch from farthest nodes to closest nodes, but unless we specify that we want paths routed into the snout, we can obtain only a generic line detector. For this reason we add to the graph an additional set of vertices located on the snout contour. These vertices are marked as sinks and they are connected with an incoming edge to every non-sink vertex in a predefined neighborhood.

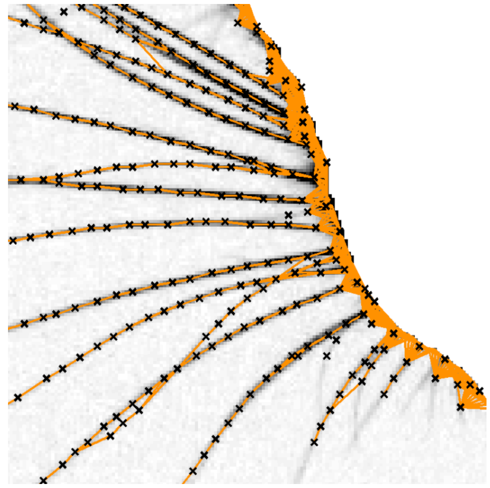

Figure 3. A zoomed picture showing graph vertices and edges

After the definition of sinks, we need to provide weights to edges. Since we defined them with unitary weights, a shortest path optimization will select the path with the minimum number of vertices regardless any whisker geometry criteria. The ideal weights should enforce smooth paths, however it is not possible to assign a static weight to an edge because the same edge may approximate perfectly one whisker shaft and poorly another with different orientation. For this reason we propose a dynamic weighting method for the graph edges, where the weight is proportional to the orientation difference between the edge and a smooth whisker approximation called stroke.

We construct strokes in the following way: since most vertices have more than one input and output edge, we select only one input and one output edge for every vertex based on shortest distance criteria. This approach decomposes the graph into sets of connected vertices, defined strokes, that smoothly approximate parts of whiskers.

For every stroke we set its farthest vertex as source node. Using a standard least squares method we fit its vertices (expressed in polar coordinates) with a 1-D third order polynomial. This smooth curve, once extrapolated toward the snout, provides in most situations a good approximation of the whisker shape and position. Based on this extrapolation we select a subset of vertices and sink nodes within a predefined distance of the extrapolated curve. For every edge, we set its weights equal to the absolute difference between the real edge end nodes coordinates and the fitted end node coordinates.

With this formulation, we penalize edges having different orientations of the polynomial template.

Starting from the source vertex, we calculate the shortest paths to the selected subset of sink nodes. For every path we fit the vertices with a cubic spline using arc-length parameter. For these splines we calculate a fitness measure based on the linear combination of smoothness measure and approximation error. The smoothness measure is calculated as the absolute sum of the local curvature calculated in M equidistant points (usually M = 30). The approximation error is defined as the mean value of the angle difference between vertices coordinates and their spline interpolation in polar coordinates. We use this error to penalize non smooth paths because they produce large approximation errors. The spline with the smallest fitness measure exhibits smooth shape and good approximation of detected peaks.

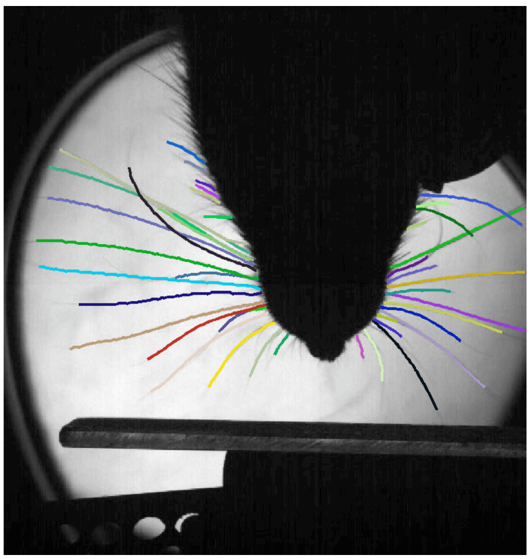

Figure 4. Detected whiskers

Some whiskers are approximated with separated strokes, so it often turns out that the optimal path derived for the farthest stroke also covers the resulting paths of the lower strokes. For this reason, we intersect the obtained paths and eliminate those completely contained in other paths. The resulting set of paths, displayed in Figure 4, is fitted with a spline and serves as the output of the method.

4. Results

We evaluated the method on movies with different number of whisker per side. Figure 4 shows a medium to high complexity scene and allows a qualitative evaluation of the method's strengths and weaknesses. Most of the whiskers have been correctly detected. There are no anomalous, jerky curves. However where whiskers stick together the method provides the optimal minimal curvature approximation, which in a small number of frames is not the correct one. The method also fails to trace whiskers upon their entire length whenever the tip is slightly out of focus as in lower right-head side of Figure 4. For this reason in the quantitative evaluation that compares our method to manually traced whiskers, we considered a true positive case if a detected whisker was overlapping at least 80% of pixels of a whisker in the gold standard, the manual tracing. In an 18-frame sequence, with about 9 manually labeled whiskers per side, the true positive rate is 250, the missed detection is 36 and the false detection is 20, mainly due to hairs or very short whiskers. True negative rate was considered zero.

The average running time for a 512x512 frame varies between 4 sec. for a single row to 7 sec. for a full-pad on a MacBook Pro© 2.0 Ghz with Matlab© R2009A. The code has not been optimized for speed.

5. Conclusion

We present a contour detection algorithm formulated for detecting whiskers. The method embeds smoothness information in a shortest path optimization framework. An extension of the algorithm is currently being tested for whisker tracking.

References

[1] M.E. Diamond, M. von Heimendahl, P.M. Knutsen, D. Kleinfeld, and E. Ahissar, *Where' and 'what' in the whisker sensorimotor system*, Nature Reviews Neuroscience, vol. 9, 2008, pp. 601-612.

[2] P.M. Knutsen, D. Derdikman, and E. Ahissar, *Tracking Whisker and Head Movements in Unrestrained Behaving Rodents*, J Neurophysiol, vol. 93, Apr. 2005, pp. 2294-2301.

[3] J. Voigts, B. Sakmann, and T. Celikel, *Unsupervised Whisker Tracking in Unrestrained Behaving Animals*, J Neurophysiol, vol. 100, Jul. 2008, pp. 504-515.

[4] J.T. Ritt, M.L. Andermann, and C.I. Moore, *Embodied Information Processing: Vibrissa Mechanics and Texture Features Shape Micromotions in Actively Sensing Rats*, Neuron, vol. 57, 2008, pp. 599-613.

[5] D.H. O'Connor, N.G. Clack, D. Huber, T. Komiyama, E.W. Myers, and K. Svoboda, *Vibrissa-Based Object Localization in Head-Fixed Mice*, J. Neurosci., vol. 30, Feb. 2010, pp. 1947-1967.

[6] A. Yilmaz, O. Javed, and M. Shah, *Object tracking: A survey*, ACM Comput. Surv., vol. 38, 2006, p. 13.

[7] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network flows*, Prentice Hall, 1993.

[8] P. Mordohai and G. Medioni, *Tensor voting*, Morgan & Claypool Publishers, 2007.