29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

FAST ANALYSIS OF THE NATURAL EDGE LANDMARKS FOR AUTONOMOUS NAVIGATION

Tikhomirova T.A., Fedorenko G.T.
State Research Institute of Aviation Systems

Keywords: *edge tracing, visual landmarks selection, pattern recognition*

Abstract

We present a noval approach to the analysis of images for navigation purposes. It is based on reading and processing of all information provided by visible and hidden edges. The problems involved data redundancy and ambiguity are solved by means of finite elements of various sizes, which are created sequentially by integration of useful data and suppression the influence of confounding factors. Close analogy with mRNA-seq is demonstrated.

1 Introduction

The vast majority of modern unmanned aerial vehicles (UAVs) are controlled by means of global positioning system. Navigation using visual landmarks is limited to certain weather conditions and types of the underlying surface.

The landmarks selection and flight routing are usually performed by human operator and require special knowledge and a priori data – maps, photos etc. These circumstances hinder the expansion of the scope of the UAV, one of which, of course, is to survey inaccessible territories, including the disaster zones.

No doubt about the fact that the bulk of useful information in the image is represented by edges and lines of different nature. Such contour objects are optimal for navigation, as they combine high positioning accuracy, detection reliability, compact storage of reference data, as well as capability of affine invariant recognition. However, full realization of these advantages requires effective methods of reading and analysing of contour information. Currently, the choice of these methods is limited, so the typical usage of visual landmarks is based on 2D correlation with reference images. Known shortcomings of the correlation method, such as high sensitivity to affine transforms and other global distortions, greatly restrict autonomous navigation of highly maneuverable vessels in poor weather conditions

This paper presents a new approach to the analysis of images for navigation purposes. It is based on reading and processing of all information provided by visible and hidden edges. The problems involved data redundancy and ambiguity are solved by means of finite elements of various sizes, which are created sequentially by integration of useful data and suppression the influence of confounding factors.

The main stages of image processing are presented in Figure 1.

Preprocessing by special filter "Snowflake" [1] provides a visualization of natural edge objects, transforming them into bright lines. At this stage we create point-size finite elements - pixel of output image.

At the second stage the mean brightness along the line segments of fixed length is computed, and the field of alternative tangents is created. Such tangents become small-scale finite elements.

At the third stage we create middle-size finite elements – short tangent polygons. Such polygons we call *villi*, and full set of them - *fleece*. Than we combine villi pairwise so that the length of fleece is doubled at each cycle and large-scale finite elements – *tracks* – are created.

It is shown that fleece structure greatly simplifies the analysis of contour information, including related detection, landmarks selection and reference object recognition and alignment. Close analogy with mRNA-seq is demonstrated.

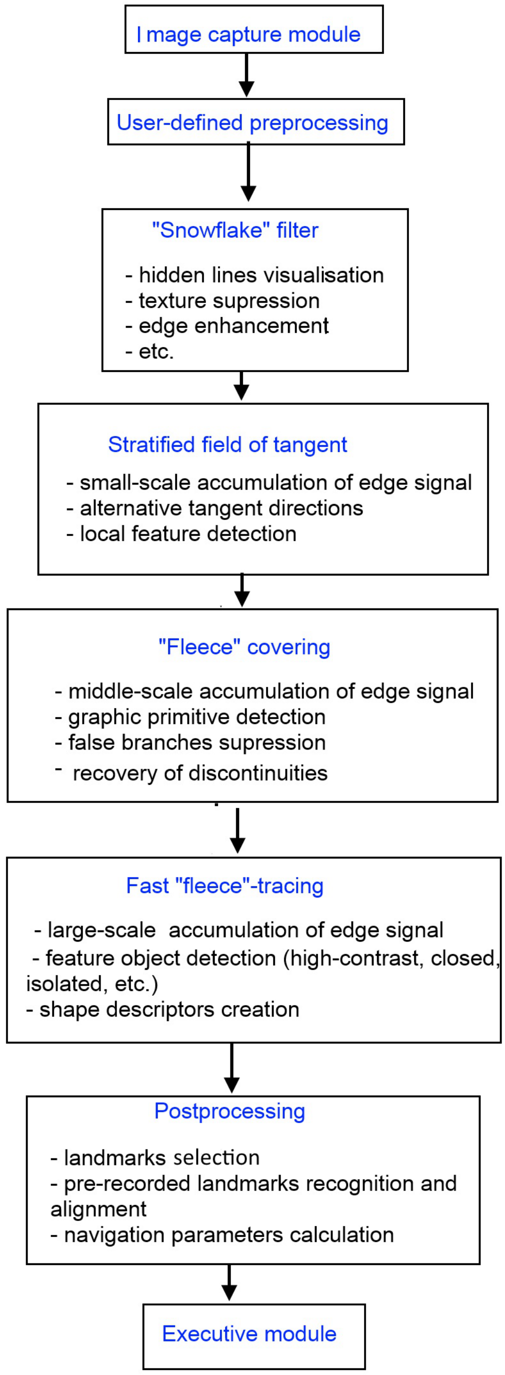


Fig.1 Stages of edge-based image analysis.

2 Collection of the edge information.

Fast and precise reading of edge information from the images is a corner stone of reliable navigation using visual landmarks. But it should be clear that the visible boundaries and lines represent only a fraction of this information. The other part consists of hidden contours - "skeletons" of the objects with fuzzy boundaries, as well as level lines of trend. Visualization of such hidden objects extends the choice of optical landmarks for navigation in difficult conditions.

Usually it is difficult to trace the large-scale edge objects on the texture background, since the latter provokes track deviation from the guide edge. Therefore, removing the small-scale high-contrast texture while maintaining boundaries increases reliability of tracing.

For visualization the hidden edge objects, texture removal and other helpful morphological operations, we have designed a "Snowflake" filter [1].

2.1 "Snowflake" filter.

Originally "Snowflake" filter was created as a lite version of the bilateral filter, which is essentially a Gaussian filter in three-dimensional space of coordinates and brightness. Despite of effectiveness in edge – preserved texture removing, usage of bilateral filter is limited by high resource demand. Although the capabilities of modern graphics processors allow to disguise this fact, it can not be ignored when designing miniature computer vision systems, especially for small UAVs.

"Snowflake" filter is based on the direct implementation of the principle "*suppress texture along the border or far away from it*". To estimate the position of the nearest border we use three linear detectors of boundaries oriented along the rays of snowflake. Simultaneously we calculate the signals of linear texture detectors oriented similarly (Fig. 2). The output signal of correction in each of three directions is proportional to the signal of texture detector, if no boundary is detected, and it fades in the vicinity of boundary. Example of texture suppression using "Snowflake" filter is shown in Figure 3, which as well presents output signal of texture detector after 4 cycles of smoothing.

The detectors of texture and boundary use no weighting coefficients that provides a minimum complexity of computations. Furthermore, the linear geometry of the detectors ensures a high resolution of the signals along the boundary with the low resolution in the transverse direction. This property gives priority to the borders of large objects, which is important for their reliable tracing.

Importantly, the signals of border and texture detectors are not independent, since the texture detector "sees" the border. This fact is used to highlight the "skeletons" of two-dimensional objects. Formally, in this mode filter suppresses "texture", which is actually a boundary. As a result, the edges of objects become broad and smooth, and their central region become sharp and take the form of dendrites (Fig. 4). This procedure provides visualization "skeletons" of the objects.

Visualization of the level lines of the trend is not difficult, so we will not discuss this procedure here.

Depending on the type of objects - borders, lines, "skeleton" or trend - we can use output signal of filter, or one of the signals of detectors for further processing. Additional enhancement of lines can be achieved using Laplace filter.

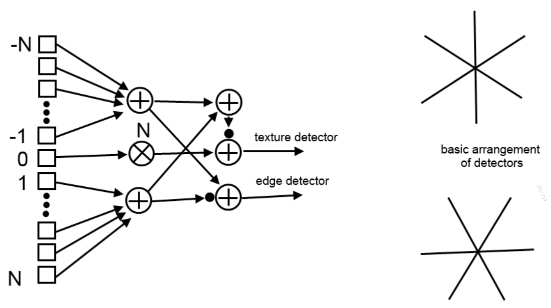


Fig.2 Structure of "Snowflake" filter.

2.2 Field of alternative tangents (FAT)

For tracing of complex edges one needs to compare multiple alternatives of track continuity. In our approach such analysis is conducted within the framework of a hidden Markov model (HMM), which allows to overcome gaps and false branching of edge induced by textured background. Alternative edge directions in HMM are described by multilayer field of alternative tangents (FAT), where each layer describes the most likely direction of the tangent in one of four angular sector. Probability for each direction is estimated by mean brightness of direct segment of fixed length *d*, which acts as a small-scale finite element.

The exact calculation of the tangent field components is very laborious, even using fast algorithms. We prefer the approximate calculation, using the fact that the integral over the tangent segment is weakly dependent on the exact position of the point of tangency. Therefore, an exact calculation is carried out on a square grid with cell size equals to half the length of the interval of integration *d*. Next, we interpolate FAT to the entire image area.

Tangential field can also be used to detect features of the edges and lines. For this purpose, the field should be rearranged so that the mean brightness of the tangent decreases monotonically from the first layer to the last. After that first layer will contain mainly information of the direct elements of contour, the second layer - the information of the elements with significant curvature, and the third layer - the information of branching, sharp corners and other features.

An example of feature detection by using FAT is shown in Figure 5. The main advantage of this method compared to differential-based methods is that extracted features are "embedded" into real edges and lines, and so exhibit high reliability.

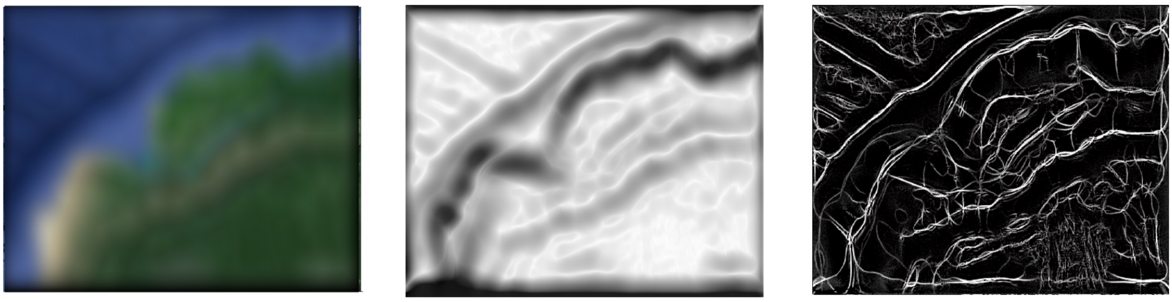
2.3 Fleece and tangent covering [3].

Fleece consists of short open polygons all segments of which belong to the field of tangents described above. These polygons are called *villi*. Each pixel in the image is the root of unique villus.

Fleece is created using a hidden Markov model for which villi are Viterbi paths

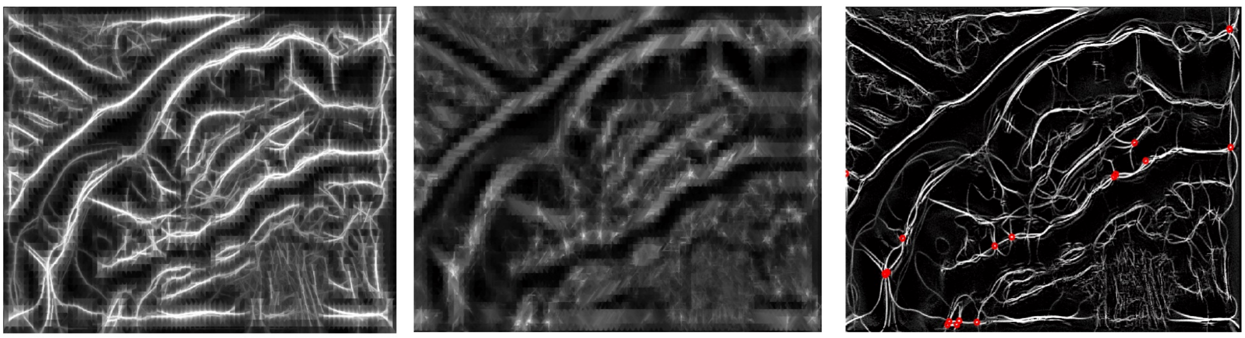
a) b) c)

Fig.3 Texture suppression and large-scale edges visualization using “Snowflake” filter: a) input frame; б) output image after 4 iterations; c) output signal of texture detector.



a) b) c)

Fig.4 Hidden edges visualization with “Snowflake” filter: a) result of edge suppression for input frame Fig.3a; b) inverted signal of edge detector; c) same signal processed by Laplace filter.



a) b) c)

Fig.5 Uisng FAT for edge features detection: a) brightness of first-rank tangets; b) brightness of third-rank tangets; c) detected features.

and the target function is average brightness of each villus.

Restriction is imposed on the angle between adjacent segments to avoid looping. First segment of the villus becomes its tip, and villus is prolonged through root segment (Fig.6).

Villi are middle-size finite elements. Their main function — middle-scale representation of contour connectivity, which is used in the next step of the tracing.

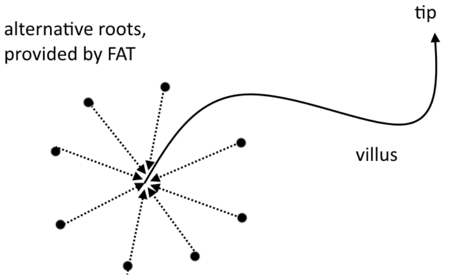
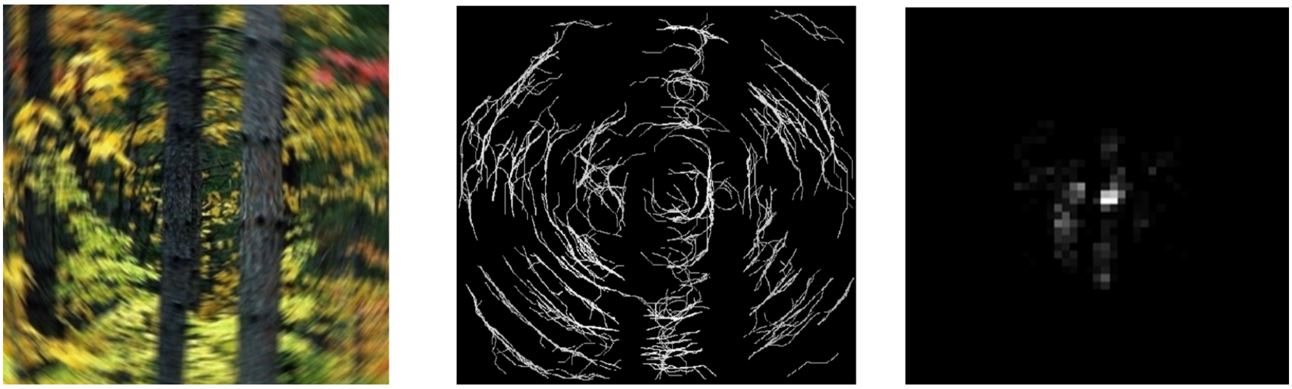


Fig.6 Alternative roots of growing villus.

It should be noted that if the root segment of villus belong to some edge, the entire villus likely belong to the same edge. Therefore, to select villi covering some edges it is enough to know their average brightness and brightness of their root segments. The sample of such villi we call *tangent covering* of edges.

Fleece can also be used to detect the graphics primitives, for example - concentric arcs typical for circular blurring. Since all parts of the villi are of equal length, analysis of villus shape is trivial. Figure 7 shows an example of the localization of the center of circular blurring by means of the tangent covering. In this example, we just select nearly-arc villi and use them to estimate position of the center of the corresponding circles. The accumulation of such signals leads to detection of rotation center.



a) b) c)

Fig.7 Localization of the center of circular blurring by means of the tangent covering: a) input frame; b) tangent covering; c) probability distribution of blur center.

2.4 Fleece-tracing.

Villi define the connectivity of the edge image in the sense that if the root of villi lies on the contour, its tip indicates quite remote element of the same edge. This creates conditions for fast tracing of extended edges.

Tracks are the large-scale finite elements. All tracks are composed only of villi provided by fleece. So villi are the shortest tracks. At each iteration tracks are combined pairwise, so the length of all tracks increases exponentially, producing more 'long-hair' fleece.

It is likely that some of the tracks are included in the fleece twice while being read in opposite directions. In this case, simple pairing of tracks leads to the formation of short cycles. To avoid such situation we allowed a small gaps between tracks. This gaps are optimized to create brightest and non-reverse tracks.

Nevertheless, the formation of cycles is not completely forbidden because it allows a relatively small object 'captures' long tracks. So we can use same-size tracks for detecting objects of various sizes.

For more effective tracing of branched lines and small-size outlines the final stage of the tracing is performed using additional options for track connection - "dendrite" and "socket" (Fig.8). These types of connections provide partial *self-repetition* of tracks, so long tracks can '*absorb*' complex shapes more accurate.

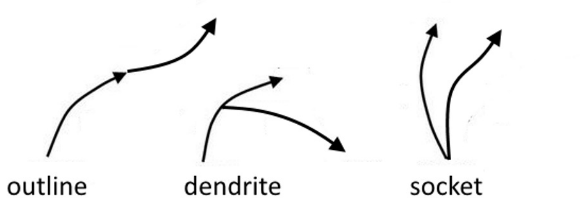
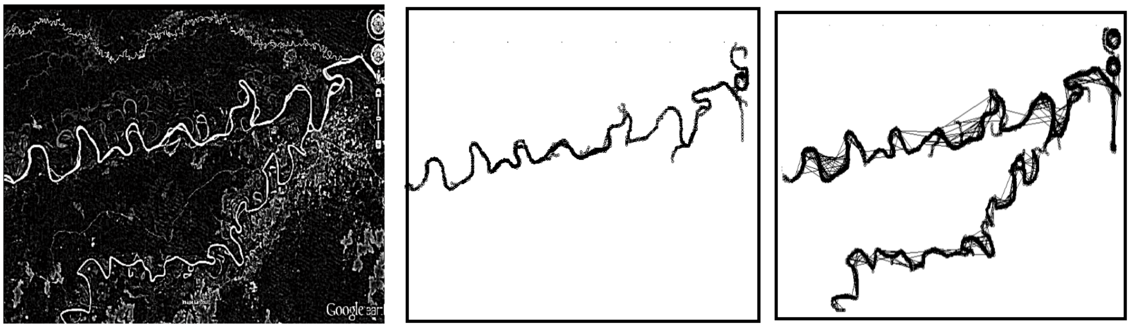


Fig.8 Types of track connections.

Examples of fleece-tracing (Fig.9) demonstrate the effectiveness of such extension.

All tracks have unique roots, while their tips are distributed between the relevant edges presented on the image. The longer the track, the smaller number of edges can capture them. At last all the tracks are captured by small number of cycles.



a) b) c)

Fig.9 Preprocessed input frame (a) and fleece-tracks selected according to 80% brightness threshold: b) only smooth edges are allowed; c) "dendrites" and "sockets" are also allowed. Thick lines with round markers represent tangents, thin lines (c) show the structure of "dendrites" and "sockets".

An important difference between initial villi and tracks is that villi composed of a predetermined number of tangent segments, while each track is a combination of two smaller - halfsize - tracks. This fact provides some advantages in detection and recognition of edge objects.

3 Analysis of tracks.

Instead of treating edges as image singularities we present them as 1D sequences of finite elements.

Fleece-tracing produce tree-like graph of tracks, where each piece of edge appears in a number of tracks being surrounded by different neighbors.

This approach approves the reliability of landmarks detection, but essentially increases the amount of information to be processed. Fortunately, a close analogy with isoform quantification from RNA sequence reads and recent advances in this area provide cause for optimism. Furthermore, the structure of fleece tracks described in previous section greatly simplifies their analysis.

3.1 RNA-sequencing.

Historically, the problem of RNA data collection was solved much earlier than the problem of their fast analysis. On the contrary, in the edge-based image analysis still there are no proper database, so it is difficult to estimate the effectiveness of proposed techniques of data processing. This makes the analogy between the track analysis and RNA-sequencing so important.

Track analysis includes detection of unknown objects, recognition of reference objects, and determination of their exact position relative to boundaries of the frame. RNA analysis includes detection of unknown transcripts, recognition of reference transcripts and read alignment in transcripts.

In both cases we are dealing with a huge amount of data, much of which is irrelevant for the problem, has an ambiguous representation, contains errors, etc. In both cases, we should detect and/or recognize the informative elements, taking into account their isoforms.

The analogy with RNA makes possible to implement methods have been successfully applied to the analysis of RNA for detection and recognition of visual landmarks. Before this analogy will be considered in details, let us explain some terms:

Sequencing - a description of the amino-acid sequence of matrix RNA.

Splicing - RNA modification including removal of non-coding sequences before transmission.

Transcript – RNA fragment described a full protein or gene. Mostly obtained after splicing, so contains mostly coding sequences.

Read - pattern obtained by "reading" DNA fragments. Contains coding and non-coding sequences.

K-mer — part of transcript by length of k nitrogenous bases or other finite units.

3.2 Sailfish [8].

The rapid development of RNA sequencing hardware allowed to create a comprehensive database of genomes. However, its application meet significant difficulties due to the high complexity of the analysis of RNA samples.

The breakthrough came when a new method of analyzing the results of RNA sequencing was presented [8]: now the time of sample processing reduces from 10-15 hours to a few minutes. The main feature of new method is the exclusion of the stage of read mapping to the reference transcripts, which occupied most of the processing time. Instead, the authors of method [8] have used the quick hashing of middle-size descriptive portions of transcripts (*k*-mers) [9] and subsequent recognition (or reconstruction) of transcripts using maximum likelihood method with a "quadratic" convergence [10].

Let us explain the idea of the method. Structurally RNA is a sequence of amino-acids. Acids have relatively few isoforms, so can be easily identified. However, their relative abundance in the input reads varies slightly and can not serve as a reliable criterion for mapping reads to the reference transcripts. Therefore, transcript detection was performed by mapping every read into some portion of reference transcript. To be detected the transcript should be fully covered by reads. This procedure was a stumbling block in the RNA analysis.

The transition from individual amino acids to the middle-size finite elements (*k*-mers) makes their expansion in the reads more individual, so we can use the *k*-mers distribution in reads and transcripts to transcript detection instead of direct read mapping. In practice, the length of *k*-mers is determined by compromise between its uniqueness and number of its isoforms, which should be small enough to provide fast and reliable hashing. The resulting speedup compared with earlier methods is approximately 20 times.

It is quite easy to see the analogy between RNA-sequencing and track analysis. In both cases we are dealing with a large sample of linearly ordered finite elements. And in both cases, the direct search of known objects is too time consuming. Also relevant is the problem of detection of unknown objects.

The results of [8] show that acceleration with preserved reliability can be achieved by creating of new classes of finite elements and increasing the usage of statistical shape description.

The importance of the presented analogy is due to the fact that many problems which only start to appear in computer vision are already identified and solved in RNA-sequencing.

3.3 Villi description and hashing.

In track analysis villi play the same role as *k*-mers in Sailfish[8]. We identify four types of villus according to their brightness and curvature (Fig.10).

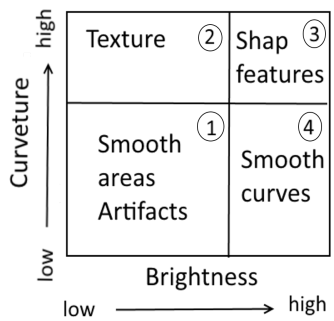


Fig.10 Types of villus.

Type 1 – "noise" - represent villi located in smooth areas, where edges are formed by random noise or artifacts, such as interlace. Those villi have low brightness and low curvature.
Type 2 - "texture" - represent villi located in areas with high contract but small-scale texture. This villi have low brightness, but high curvature.
Type 3 - "corner" - represent villi guided by sharp corners of large-scale edges *and* fitted this corners well. This villi have high brightness and high curvature. If villus is guided by sharp corner, but *not* fitted well, it has low brightness and is considered as *"texture"*.
Type 4 - "curve" - represent villi guided by smooth edge or line.

3.4 Structure of visual landmarks.

Primitive visual landmarks should contain of at least two *corner* villi to provide precise localization. *Corners* should be linked by few *curve* villus to provide precise estimation of scale and orientation. To provide reliable detection the landmarks should appear in essential sample of tracks. Such objects we call small-size landmarks (SSL).

One should note that *corner* villi have no particular scale or orientation, so they can be detected invariant to affine transforms. *Curve* villi are also mostly scale-invariant due to their small size, but provide preferred direction.

In order to detect SSL we should scan tracks and select all sequences of *curve* villi with *corner* villus on both ends.

We define the *reliability of the detection* of SSL as a number of tracks containing it. This criteria is easy to estimate because, due to fleece-tracks properties, all tracks which contain the start villus of SSL should contain whole SSL. We just need to take into account both directions of "reading".

Finally we create large-scale landmark (LSL) by mapping all reliable SSL as well as reliable links between them.

3.5 LSL description and recognition.

LSL are full analog of transcripts in RNA-sequencing. They are highly informative, but it is hard to align individual track on such landmark. Therefore, to ensure high maneuverability of UAV, we should provide alignment-free recognition of LSL.

To do so, we define a set of middle-size landmarks (MSL) as a reliable sequence of SSL and links extracted from single LSL. Links can be both solid (i.e. consist only of *curve* villi) or mixed type, though solid links are preferred. Such MSL are then grouped into classes of isoforms and are used to produce statistical description of each LSL in database. Thus we achieve instant recognition of LSL.

Final validation and alignment can be perform as described in [8].

4. Conclusion.

Images provided by means of computer vision systems, are two-dimensional arrays of data, in which the bulk of useful information is represented by edges of various nature. However, modern edge-based image analysis does not provide processing techniques suitable for most applications, including navigation using visual landmarks.

In this paper we present a solution to this problem based on the fast tracking and multiscale analysis of all significant - both visual and hidden - edges in the image. It is shown that a close analogy between edge tracing and RNA-sequencing allows us to apply methods and algorithms of proven effectiveness on an extensive base of genetic data for detection and recognition of visual landmarks.

References

[1] Tikhomirova T.A. *"Snowflake": the fastest bilateral filter,* Vestnik komp'iuternykh I informatsionnykh tekhnologii, *Vol.9, pp.18-21, 2012*

[2] Tikhomirova T.A., Fedorenko G.T., Kirillova L.N., *Tangent covering of contours: synthesis and application,* Vestnik komp'iuternykh I informatsionnykh tekhnologii, Vol. 7, pp.17 – 21, 2011

[3] Tikhomirova T.A., Tuzova T.E., Fedorenko G.T., *Features detection using tangent covering,* Vestnik komp'iuternykh I informatsionnykh tekhnologii, Vol. 10, pp. 11 – 15, 2011

[4] Tikhomirova T.A., Fedorenko G.T., Kirillova L.N., *Blur estimation using tangent covering,* Vestnik komp'iuternykh I informatsionnykh tekhnologii, Vol. 5, pp. 8 – 12, 2012

[5] Tikhomirova T.A., Fedorenko G.T., Kirillova L.N., *Fast tracing of contour images,* Vestnik komp'iuternykh I informatsionnykh tekhnologii, Vol. 12, pp. 22 – 29, 2011

[6] Tikhomirova T.A., Fedorenko G.T., Kirillova L.N., *Fleece-tracing of closed and branched curves* Vestnik komp'iuternykh I informatsionnykh tekhnologii, Vol.2, pp.16-21, 2014.

[7] Tikhomirova T.A., Fedorenko G.T., Kirillova L.N., *Fleece-tracing of contour images using "flex" links* Vestnik komp'iuternykh I informatsionnykh tekhnologii. Vol. 4, pp.10 – 12, 2013

[8] Rob Patro, Stephen M. Mount, Carl Ringford. *Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms*. Nature biotechnology, Vol.32, pp 462-464, 2014.

[9] R. Varadhan, C. Ronald *Simple and globally convergent methods for accelerating the convergence of any EM algorithm*. Scandinavian Journal of Statistics, Vol. 35, No. 2, pp 335-353, 2008.

[10] W. Zheng, L. M. Chung, H. Zhao *Bias detection and correction in RNA-Sequencing data*. BMC Bioinformatics, Vol.12, No.1, pp 290, 2011

mailto:wat902h@hotmail.com

Copyright Statement