

# Clustering Techniques Versus Binary Thresholding for Detection of Signal Tracks in Ionograms

Artem M. Grachev and Andrey Shiriy

National Research University Higher School of Economics, Moscow, Russia  
amgrachev@hse.ru, andreyschiriy@gmail.com

**Abstract.** An ionogram is a display of the data produced by an ionosonde. It is a graph of the virtual height of the ionosphere plotted against frequency. In addition to “useful signal”, an ionogram almost always contains noise of different nature, a so called background noise. That is why the signal filtering task becomes so important. There are two groups of methods to this end. The first group features methods of computer vision for image processing, namely, different filters and image binarization. The second group includes adapted clustering methods. In this paper, we show how several methods work for filtering “useful signal” from noise and emissions.

**Keywords:** ionograms, image filtering, image processing, similarity measures

## 1 Introduction

The data of radio sounding is necessary for enhancement of over-the-horizon radar systems, systems of shortwave communication, as well as for solution or many problems in radiophysics and geophysics [1].

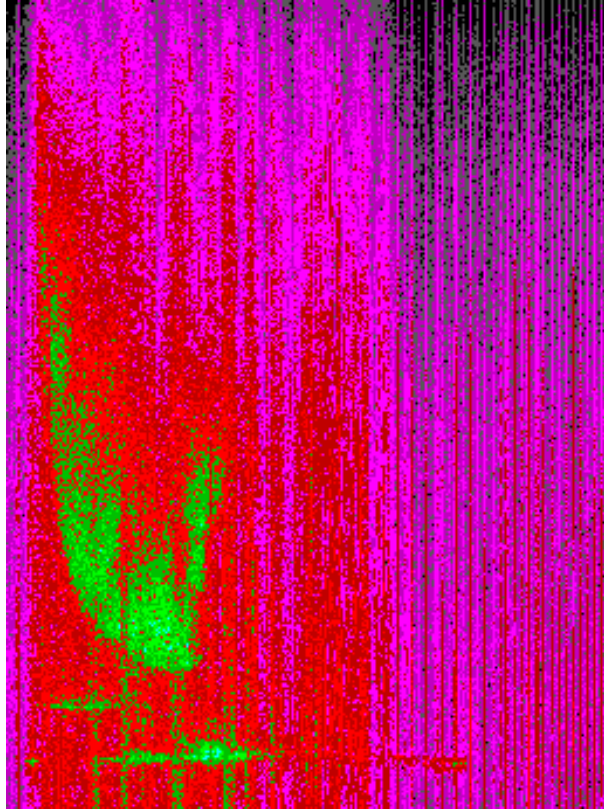
Usually, the results obtained by an ionosonde are represented by means of ionograms[2]. An ionogram of oblique radio sounding of the ionosphere shows a dependence of the amplitude of the received signal from the frequency  $f$  of sounding and the group delay time  $\tau$ [3].

Due to multipath shortwave propagation in the ionosphere, an ionogram contains tracks of different signal modes. In addition to the useful signal, there is a noise of different nature in ionogram images. In Fig. 1, one can see the mode of the signal’s track (a sloped body in the bottom left part of the ionogram), background noise, and concentrated noise, i.e. vertical stripes<sup>1</sup>.

When we work with ionograms one of the most important problem is to filter the useful signal from the noise. There are several types of useful signals. In fact,

---

<sup>1</sup> The data of ionograms shown in the paper are available at <https://drive.google.com/open?id=0Bxdt09RRxaqMY2pCYUI4eWR0T1U>. More comprehensive datasets are available from the second co-author by request.



**Fig. 1.** Ionogram example

we have a problem similar to automatic classification or clusterization depending on the availability of training (labeled) data.

The rest of the paper is organised as follows. In Section 2, we consider signal segmentation using image processing methods. In Section 3, we use machine learning methods for the same purposes. We treat an input image as a dataset with each pixel as a separate element and then cluster it. In Section 4, we try to exploit the best of these methods to create our final algorithm. In the conclusion, we discuss shortly relevant techniques and problems for future work.

We should note that when we tested our methods, we tried several configurations for our models (sometimes enumerating parameters' values by grid search). Of course, there may be better configurations of parameters in a particular case.

## 2 Detection of signal tracks by image processing methods

In this approach, we consider an ionogram as an image. We need to filter out the noise and isolate the signal track of an input ionogram. We have tested two filters for image filtering: the median filter and the filter given by the matrix below.

$$Ker = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

In the next example, we show the original image and the results of application of two filters to the image and its binarization by thresholding.

Image binarization is the way to define the class of each pixel as signal/background by thresholding. That is we set the threshold value of brightness and apply it to all the pixels; the pixels with brightness higher than this threshold belong to the first class, and the remaining ones belong to the second. In Fig. 2, the images of the original ionogram are shown in three color model. And, in the remaining figures, for illustration we use only one color model.

It is clear that filtering with  $Ker$  matrix is able to better keep signal's shape and eliminate the noise in comparison with the median filter.

## 3 Detection of signal tracks by machine learning methods

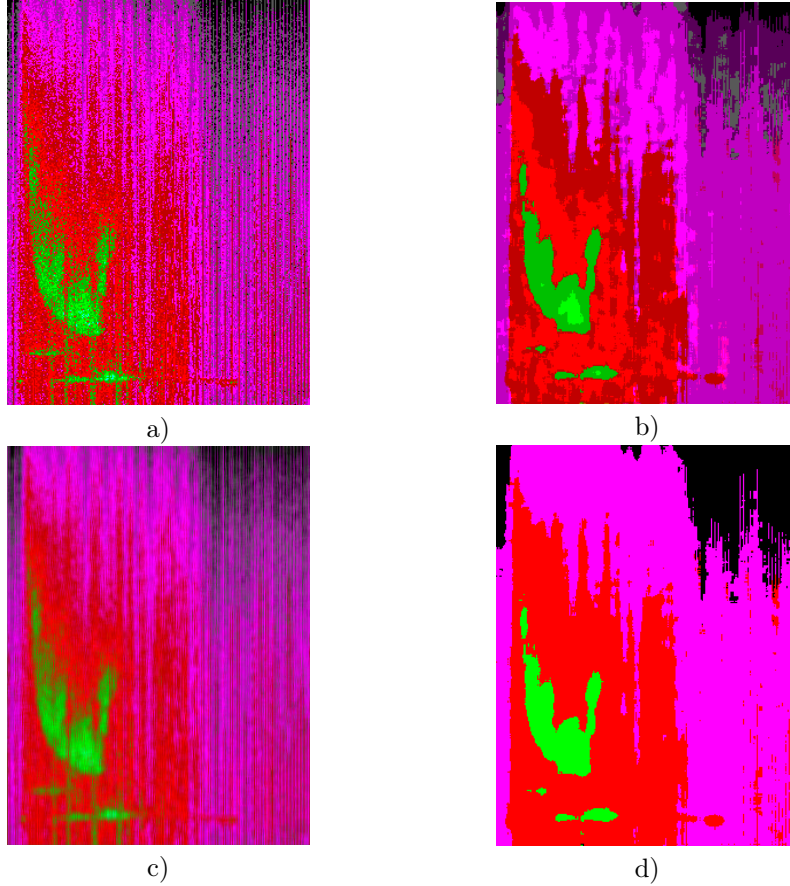
Another approach is based on the ionogram representation in form of triples  $\langle x, y, V \rangle$  for each original pixel, where  $x$  and  $y$  are pixel's coordinates and  $V$  is the value of the pixel brightness. After such transformaiton we try to do clusterization. We hypothesise that signal's pixels should belong to a separate cluster. This approach is similar to the well-known image segmentation methods that one can find, for example, in this book [4].

After clustering we again represent the results as an image. We replace the value of brightness of each input pixel by its cluster label. These three methods from scikit-learn machine learning environment [5] have been applied:

1. K-MEANS
2. DBSCAN [6]
3. MEAN SHIFT [7]

The last two methods have been chosen since they do not need to know the number of clusters in advance; moreover, according to locality hypotheis they can capture both similarity in signal/noise values and spatial closeness in axes  $x$ - $y$  (in fact,  $f$ - $\tau$ ).

DBSCAN have worked rather good visually. Main disadvantage of this method is a necessity to configure its parameters separately for each image. In Fig. 4,



**Fig. 2.** Ionograms: a) the original image, b) preprocessing by the median filter, c) filtering with matrix  $Kers$ , d) binarization

you can find the results of processing of the original ionogram given in Fig. 3 by DBscan with  $\varepsilon = 4$  (the neighbourhood size),  $N = 100$  (the number of points within the neighbourhood).

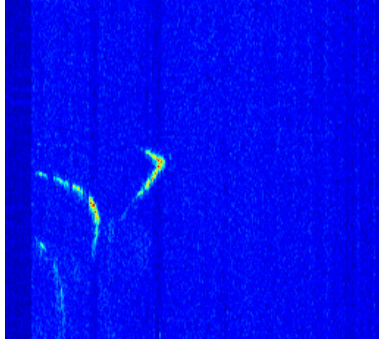
Coordinates are scaled in the way below:

$$x_{new} = \frac{x_{old}}{\max(x_{old})}, y_{new} = \frac{y_{old}}{\max(y_{old})} \quad (1)$$

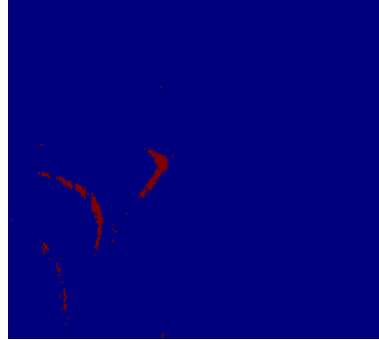
Next example launched with  $\varepsilon = 1$ ,  $N = 50$  and with following coordinate transformation:

$$x_{new} = \frac{x_{old}}{\max(x_{old})} \cdot 10, y_{new} = \frac{y_{old}}{\max(y_{old})} \cdot 10 \quad (2)$$

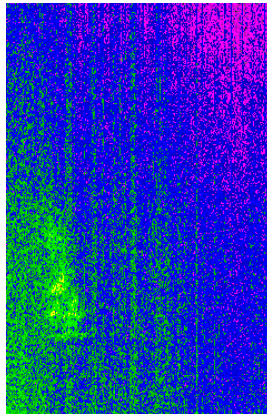
In the figures above, machine learning methods have been applied to the original image. However, we should note that we get better results if we first applied filtering and then clustering.



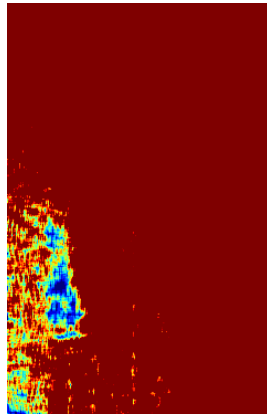
**Fig. 3.** Original image



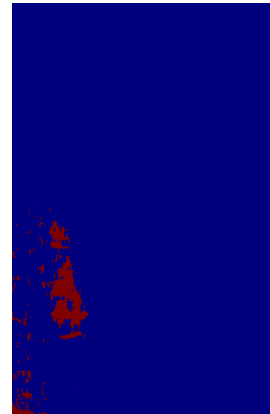
**Fig. 4.** DBscan results



**Fig. 5.** Original image



**Fig. 6.** Filtered image



**Fig. 7.** Mean shift results

It turns out that the most appropriate method for this task is MEAN SHIFT, applied after image filtering. The Python implementation of MEAN SHIFT allows us to choose the Parzen's window size automatically for each image. It depends on distance between objects; we have used 70th percentile of all pairwise distances. This property of MEAN SHIFT is much more suitable in comparison to DBSCAN since DBSCAN needs individual options for each image. Another advantage of MEAN SHIFT is its speed. Here we have also used coordinates transformation from Eq. 2.

## 4 Conclusion

This paper presents the first steps of comparison of image processing and machine learning techniques for signal detection in ionograms. Both groups of methods are suitable for noise filtering and isolation of the original (important) signal. We have compared several methods of computer vision and machine learning for this problem. It seems that MEAN SHIFT works better than its two competitors in the conducted comparison. In the future we plan to apply deep learning methods for better signal detection based on a large set of ionograms. The usage of autoencoder for automatic clustering of signal types is an attractive opportunity as well. Other image segmentation techniques that are widely used in computer vision community are highly relevant as well.

## References

1. Shiriy, A.: Development and modeling of algorithms for automatic measurement of the parameters of ionospheric shortwave radiolines. PhD thesis, Saint Petersburg State University of Telecommunications after M.A. Bonch-Bruевич (2007) (In Russian).
2. Kolchev, A., Shumaev, V., Shiriy, A.: Equipment for Research of HF Ionospheric Multipath Propagation Effect. *Journal of Instrument Engineering* **51**(12) (2008) 73–78
3. Williams, G.: Interpreting digital ionograms. *RadCom (RSGB)* **85**(05) (2009) 44–46
4. Forsyth, D.A., Ponce, J.: *Computer Vision - A Modern Approach*, Second Edition. Pitman (2012)
5. Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., Mueller, A.: Scikit-learn: Machine learning without learning the machinery. *GetMobile* **19**(1) (2015) 29–33
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, AAAI Press (1996) 226–231
7. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5) (May 2002) 603–619