Satellite Tracks Removal in Astronomical Images

Haider Ali and Christoph H. Lampert and Thomas M. Breuel

Image Understanding and Pattern Recognition (IUPR) Research Group
German Research Center for Artificial Intelligence (DFKI)
and Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
{ali, chl, tmb}@iupr.net

Abstract. This paper describes a new system for "Finding Satellite Tracks" in astronomical images based on the modern geometric approach. There is an increasing need of using methods with solid mathematical and statistical foundation in astronomical image processing. Where the computational methods are serving in all disciplines of science, they are becoming popular in the field of astronomy as well. Currently different computational systems are required to be numerically optimized before to get applied on astronomical images. So at present there is no single system which solves the problems of astronomers using computational methods based on modern approaches. The system "Finding Satellite Tracks" is based on geometric matching method *"Recognition by Adaptive Subdivision of Transformation Space (RAST)"*.

Keywords: Satellite Tracks, Geometric Matching, Astronomical Images, RAST

1 Introduction

1.1 Motivation

Astronomical images play an important role in man's effort to understand the universe. These images are taken by space observatories spread across the globe. Acute weather conditions are required for image capturing. It takes several minutes to take a single astronomical image. Therefore it is important to extract maximum information from the astronomical images. These images contain important information about stars and galaxies and irrelevant information (like satellite tracks) which appear quite often. The target of our application is "weak lensing", which is a statistical method to estimate mass distributions in the universe by measuring galaxy ellipticities. Therefore, astronomers want to measure the ellipticity of very many small objects in an image which has to be done automatically. The presence of satellite tracks would disturb those measurement. Since data collection is time consuming, the images containing additional objects can not be simply discarded. Instead, irrelevant objects have to be removed from these images [5].

Currently astronomers remove these satellite tracks manually from images according to [4]. We replace this traditional approach by using a computational system. This system is based on a *Geometric Matching* technique called *"Recognition by Adaptive Subdivision of Transformation Space (RAST)"*. A working system will help in saving a lot of human effort to remove these tracks manually.

1.2 Problem Formulation

The objective of this research work is to find satellite tracks in the images taken by astronomers. The images taken from the sky are contaminated by a lot of noise (Cosmic rays, CCD defects, ghost images and satellite tracks) which disturbs the detection or identification of important objects in the images like stars and galaxies. These tracks can be characterized as straight lines in the images. The major aim of this part of work is to develop a system to find and remove satellite tracks in astronomical images. In this system a well known algorithm RAST is applied based on Geometric Matching techniques.

1.3 Related Work

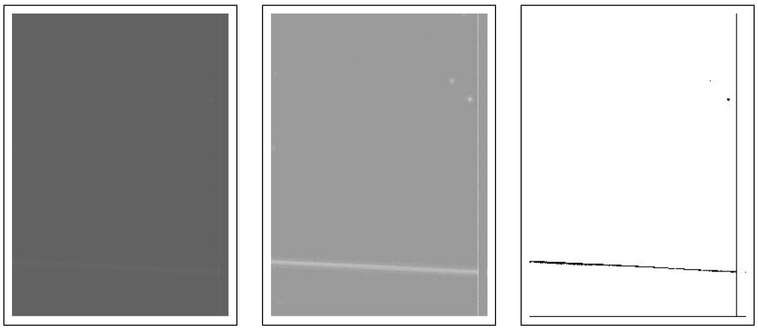
There is no computational method to reliably identify and remove satellite tracks at present. Currently astronomers do this job by hand after selecting and masking the images having satellite track. After masking satellite tracks they ignore the masked area while doing further processing of the objects. A general adaptive method *Cleaning Sky Survey databases using Hough Transform and Renewal String Approaches* was introduced by A.J. Storkey et al. (2004) to remove unusual objects in the astronomical images. This method was developed to remove four types of objects - satellite or aeroplane tracks, scratches, fibres and other linear phenomena introduced to the plate, circular halos around bright stars due to internal reflections within the telescope and diffraction spikes near to the bright stars in the sky survey data using catalogue of objects as described in [8]. The system we present is different from the adaptive method. As our system uses pixel based representation of images instead of using catalogue of objects. This system is based on geometric matching techniques a modern approach. This paper is organized as follows: Section 2 gives a brief overview of image pre-processing and related work including data preparation for satellite tracks system. Then the use of RAST algorithm in the system is described for finding straight lines in the image in Section 3 and Section 4. Section 5 presents the evaluation of the tasks and Section 6 leads to the conclusion, discussion of unsolved problems and future work.

2 Finding Satellite Tracks

2.1 Image Pre-Processing

The available astronomical images are generally in Flexible Image Transport System (FITS) format. This format is widely used in astronomy for convenient exchange of astronomical data. FITS images are composed of a sequence of header data units (HDUs). A header data unit contains keywords (value statements) that describe the organization of the data in HDUs and the format of the data contents. Based on the special structure of these header data units, FITS images may provide additional information like used instrument, status and history of the data etc. [6]. The given images were binarized and converted from FITS grayscale images to Portable BitMap (PBM) images. In this work the binarization is done by using global thresholding. Global thresholding sets all pixels above a defined value to white and the rest of the pixels to black in the image. It is very important to decide the appropriate threshold value to binarize the image, though it is difficult to decide a global value which is suitable for all images [7]. In this case the intensity range does not vary much from image to image. After looking at grayscale values in different images on the available data set, 102 were founded as a suitable value (40 %) threshold of total intensity range $0 - 255$. `Convert`, a utility in the ImageMagick software package was used to binarize the grayscale images. An example image before and after binarization is shown in Figure 1.

After binarization the morphological operation *Opening* was applied to remove noise and CCD defects in the images. Opening is based on the morphological operations Erosion and Dilation. Opening smoothes the inside of the object contour, breaks narrow strips and eliminates thin portions of the image. It is done by first applying erosion and then dilation operations on the image. Erosion shrinks the foreground objects in the image by certain amount. The amount of growing and shrinking the size of objects depends on the structuring element. CCD defects and noise have to be eliminated. These defects have a typical size of one to three pixels. So the opening is performed by using 3 × 3 structuring element. `Pgmmorphconv` is used for the opening operation. It is a utility in the



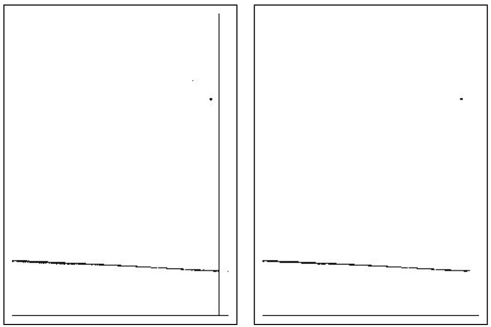
(a) original grayscale image

(b) with contrast stretching

(c) after binarization

Fig. 1. Example of binarized image (foreground objects become visible), the binarized image is inverted to white background and black foreground

Netpbm software package, a graphics programming library. The opening effect is demonstrated in Figure 2.



(a) binary image (b) after opening

Fig. 2. Example of opening effect (CCD defect and bad pixels are removed), Both the images are inverted to white background and black foreground

3 Geometric Matching

3.1 General Description

There are different algorithms to perform geometric matching. Many geometric matching problems in computer vision are based on image models. A model of an image can have geometric features like points, lines, arcs etc. Different algorithms are developed to match these geometric features in the model image [3]. As satellite tracks have geometric resemblance to straight lines, we look for line features in the image. In this work the RAST algorithm is applied for finding satellite tracks in the image.

3.2 Data Preparation

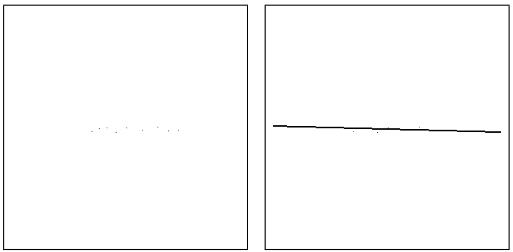
RAST takes sample points as input features. The computation of these sample points is described in this section. In binarized images these input features consist of the pixel positions of the foreground objects. The binarized images contains white foreground objects on a black background. A routine is written in C++, which computes the (x, y) positions of all the white pixels in the image. These pixels constitute the sample points.

3.3 Recognition by Adaptive Subdivision of Transformation Space (RAST) Algorithm

The RAST algorithm was developed by Breuel [1]. RAST consists of a family of geometric matching algorithms, one of these are for finding lines.

3.4 Geometric Transformation

The RAST algorithm is applied on prepared sample data points. RAST is based on forward transformation (model to image) and an error model [2]. In this case, the algorithm takes a collection of sample data points and tries to fit an optimal line on these sample points in the image.
The algorithm is implemented using hierarchical and adaptive subdivision of the space of line parameters. RAST implements a "best first" search by a binary tree based on recursive subdivision of the parameter space [2].
The RAST algorithm considers a set of all possible parameters as region of interest before execution. Then it recursively divides the parameter space into sub regions. For each region in the parameter space there is a set of consistent image features under bounded error. During the execution RAST eliminates the regions of disinterest. These are the regions which do not contain a solution [2].
An illustration of sample points and RAST line fitting is presented in Figure 3.



(a) points (b) RAST line fitting

Fig. 3. A hand made image with sample data points and reconstructed copy of the hand made image with RAST line fitting

3.5 Parametrization

The RAST algorithm for finding lines based on the parametric model, described by line parameters (r, θ). In the xy-plane, r is the distance of the line from the origin and θ is the angle between the perpendicular and the x-axis. The angle θ is always measured anticlockwise from the positive side of the x-axis. As shown in 1.

$$x\cos\theta + y\sin\theta = r \tag{1}$$

An illustration of line parameters in xy-plane is presented in Figure 4. There are different parameters to the algorithm like the *epsilon*, *quality*, *tolerance* and *angle tolerance.*

- The epsilon (eps) parameter defines the distance up to which a point can contribute to the line.

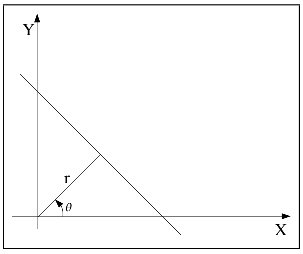


Fig. 4. Representation of r and θ in xy-plane

- The quality (minquality) parameter specifies the minimum acceptable quality of a line as given by:

$$q(\vartheta, P) = \sum_{k=1}^{N} max(0, 1 - \frac{d_k^2}{\varepsilon^2}) \tag{2}$$

  where ϑ is the set of parameters (r,θ), P is set of points, N is total number of points and d_k is the distance of k^{th} point from the line.
- The tolerance (tol) and angle tolerance (atol) parameters specifies the allowed deviation of the parameters r and θ respectively from their optimal values.

The RAST algorithm is used with default values of these parameters. The default values of tolerance is set to 0.1 and the value of angle-tolerance is set to 0.001 in the implementation by Breuel [2]. To find lines in the astronomical images the the epsilon (eps) parameter value is set to 2 and the weight (minweight) of the line is set to 1000.

3.6 The Algorithm

The algorithm is used with the interest of finding the best match of all possible lines in the image. The execution steps of RAST algorithm are explained to calculate the quality function and error tolerance function. [2]. Following are the steps involved in the execution of RAST for finding lines.

- Step 1. Choose an initial region T in parameter space containing all specified parameter values.
- Step 2. Define a priority queue Q, where the priority of the queue Q is based on the upper bound on the quality of possible best match in that region as defined by the equation 2. The initial region is inserted into the priority queue Q.
- Step 3. Extract the element of the highest priority from the queue Q.
- Step 4. If the upper bound on the quality of the match in the extracted region T is less than the minimum quality threshold, go to step 7.

- Step 5. If the size of the region T is less than the tolerance size of the region dimensions r and θ then the region T is reported as a solution and continue to step 3.
- Step 6. If the specified size of the region T is greater than the minimum size of the interval for a solution then it is subdivided into two sub regions $T1$ and $T2$ and these sub regions are pushed into the priority queue Q and the algorithm continues to step 3.
- Step 7. Terminate

3.7 Output

The algorithm returns a list of parameter values r and θ. The RAST algorithm provides an output parameter *maximum result* to find more than one lines in the images.

4 Image Reconstruction

4.1 General Description

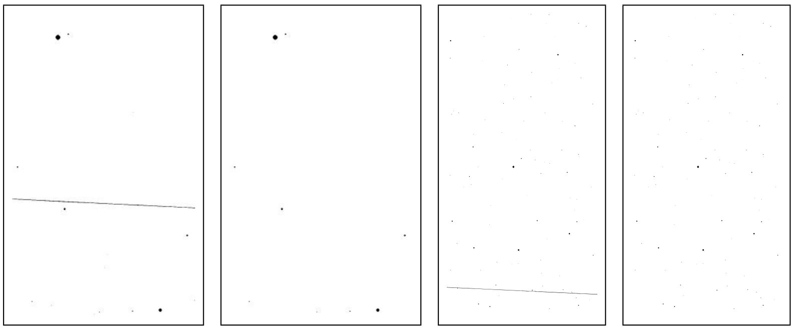
To remove lines from the image the position of the lines are computed by the RAST algorithm. To visualize the removal of the satellite tracks having geometric resemblance to lines, a new image is reconstructed from the binarized copy of the image. It is important to make a decision whether a line founded by RAST algorithm is a satellite tracks or not. This decision is based on the size of the line, that is how many pixels contribute to this line. An assumption is made that if the weight of the line is 1000 or greater, then the line found by the algorithm is considered as satellite track.

4.2 Painting Image

This algorithm reads the original image (PGM file) and data (line file) of pixel positions computed by the RAST algorithm. The number of lines returned by the RAST algorithm depends on the value of maxresult parameter. By setting the value of the maxresult parameter to more than one, the algorithm returns the optimal quality lines in decreasing order. The line found by the algorithm is painted black in the reconstructed image with a width of 20 pixels. The painted width of the line is decided on the basis of observed thickness of the satellite track. Examples of binarized image before and after the removal of a satellite track are shown in the following 5.

5 Evaluation

The evaluation of the satellite tracks removal system is done using a large database of astronomical images. This database consists of 1000 images. A total of 102 random images are chosen to evaluate this method. To generate ground



(a) binarized (b) reconstructed (c) binarized (d) reconstructed

Fig. 5. Example: Binarized image containing satellite track : Reconstructed image after removal of satellite track, in these images few objects are removed because they were very close to the satellite track

Table 1. Different types of errors made by satellite tracks detection application. Each column represents the different type of error. The column labels are: total true positive (T_p), total true negative (T_n), total false positive (F_p), total false negative (F_n)

	Total Images	T_p	T_n	F_p	F_n
Satellite Track Detection	102	2	87	5	8

truth information, the images are visually examined and annotated whether they contain satellite tracks or not. The system described in Section 2 is applied to find satellite tracks in the given dataset of images.

5.1 Results

The results of the system are presented in table 1. The table gives *true positive* (T_p), *true negative* (T_n), *false positive* (F_p) and *false negative* (F_n) number of satellite tracks detection.

True positive (T_p) are the satellite tracks which actually exists and are reported as tracks as well. True negative (T_n) are the images without satellite tracks, which are correctly separated as free of tracks. False positive (F_p) are the images which do not contain satellite tracks but are detected as images containing satellite tracks. False negative (F_n) are the images containing satellite tracks but are not reported as images containing satellite tracks. The system has detected 2 true positive (T_p) images containing satellite tracks which are processed by the system as well. There are 87 true negative (T_n) images identified by the system where there is no satellite track exists. Therefore the accuracy of system for finding satellite tracks is 87.3%, which is the total number of correct detections. The system has detected 5 false positive (F_p) images. The false positive (F_p) images are analyzed and it turned out to some expanded stars and galaxies in the image due to the brightness and make a *Light Spike* in the

Table 2. Different types of results made by RAST algorithm while finding satellite tracks in the images. Each column represents different type of error. The column labels are: true positive (T_p), true negative (T_n), false positive (F_p), false negative (F_n)

Algorithm	Total Images	T_p	T_n	F_p	F_n
RAST	94	2	87	5	0

available database of images. These light spikes look like vertical satellite tracks. While considering these light spikes as satellite tracks the algorithm detects and removes them as well. These light spikes are removed completely except the circular shaped bright stars or galaxies due to their thickness. An illustration of the images having stars and galaxies affected with the telescope light spikes before and after removal of these light spikes are presented in the Figure 6.

The system has detected 8 False negative (F_n) images. These False negative (F_n) images are analyzed and the images were found to contain satellite tracks but in very low brightness. The system was not able to detect satellite tracks in them. After analyzing False negative (F_n) images deeply, it is found that the original FITS images do not contain *DATAMIN* and *DATAMAX* fields. As the header data units (HDUs) of FITS images have DATAMIN and DATAMAX containing the range of grayscale values in the images. DATAMIN describes the minimum and DATAMAX describes the maximum gray value in the image. The foreground objects are lost already in the conversion step, so there is no chance of detection by RAST algorithm. This failure was not because of the RAST algorithm but was due to the missing information about DATAMIN and DATAMAX in FITS header data units (HDUs) which are the essential information required for the conversion process. Therefore a new evaluation scheme was decided where these black images were removed from the test set. The results of this new evaluation scheme is presented in a separate table 2. As the table shows that the RAST algorithm did not find any false negative (F_n) results. The accuracy of our algorithm is increased to 94.7% with this evaluation scheme.

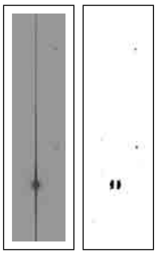


Fig. 6. The original image containing light spikes, there after the removal of these light spikes a few points of the galaxies still remains visible in the output image

6 Conclusion and Future Work

In this paper a computation method for identification and removal of satellite tracks in astronomical images is presented as an alternative of manual approach. This computation method is based on geometric matching techniques for finding lines in the images. There are different methods available for finding geometric shapes in images. As satellite tracks has geometric resemblance to lines, so the focus of this part of thesis was to find lines in the astronomical images. The RAST algorithm is applied to find these satellite tracks. The problem in this method was to find the right size of a satellite track. The problem of measurement of the size of satellite track is solved by visualizing satellite tracks of different sizes. While identifying the satellite tracks in astronomical images, it is important to know the actual size of the satellite tracks. Currently, the system is implemented based on the assumption that if 1000 or more pixels contribute to make a line then this line is considered as a satellite track. This assumption could be refined after knowing the minimum size of the satellite track. Results of this method are visualized by reconstructing the image after removal of satellite tracks. The provided solution is quite fast. It can find and remove satellite tracks from an image of size $2000X4000$ pixels on a pentium 4 processor running at 2 GHz with a typical computational time less than 30 seconds. This computation time could be decreased with high performance hardware. The system did not have any false negatives, which means that all satellite tracks were removed, which is the main priority of the system when it should be applied for weak lensing. This solution will provide ease to astronomers for their further processing of objects in the images.

References

1. T.M. Breuel. *Geometric Aspects of Visual Object Recognition.* PhD thesis, Massachusetts Institute of Technology, 1992.
2. T.M. Breuel. Finding lines under bounded error. *Pattern Recognition Letters*, 29(1):167–78, 1996.
3. T.M. Breuel. A practical, globally optimal algorithm for geometric matching under uncertainty. In *International Workshop on Combinatorial Image Analysis (IWCIA August 2001), Philadelphia, CA*, 2001.
4. T. Erben. Personal communication, 2005.
5. T. Erben and M. Schirmer et. al. Gabods: The garching-bonn deep survey; iv. methods for the image reduction of multi-chip cameras. arXiv.org:astro-ph/0501144, 2005.
6. William D. Pence. Fits overview. http://fits.gsfc.nasa.gov, 2004.
7. R. E. Woods R. C. Gonzalez. *Digital Image Processing, 2nd Edition.* Prentice-Hall, Inc., 2002.
8. A.J Storkey, Hambly N.C., C.K.I. Williams, and R.G. Mann. Cleaning sky survey databases using Hough Transform and renewal string approaches. *Monthly Notices of the Royal Astronomical Society*, 347(1):36–51, 2004.