Data Mining of Astronomical Processing Configuration Parameters by the ThresHolds Software

Sergii Khlamov[a], Tetiana Trunova[a], Iryna Tabakova[a]

[a] *Kharkiv National University of Radio Electronics, Nauki avenue 14, Kharkiv, 61166, Ukraine*

Abstract

In this paper we presented the implementation of the data mining purposes related to the astronomical processing configuration parameters during invocation the image processing pipeline. Each processing module in the pipeline has a lot of configuration parameters, which allow tuning the processing process as well to improve the accuracy and speed of calculations during processing. Because of the big amount of such configuration parameters in the complex pipeline of the scientific software, the data mining approach is very useful and productive. For our research we have selected the scientific software for detection the moving objects in a series of CCD-frames called “CoLiTec”. Such software performs a lot of different image processing tasks, like filtering, background alignment, objects detection, astrometry, photometry, motion detection, etc. The CoLiTec software consists of more than 30 mathematical and processing modules related to the different stages of the image processing, where each of them has a lot of configuration parameters to be set. So, to resolve the management issues of astronomical processing configuration parameters, the developers decided to create the new software for the data mining of such parameters called “ThresHolds”. It was implemented as a software with graphical user interface (GUI) using the Java programming language, JavaFX technology. The main goal of ThresHolds software is to classify configuration parameters, manage and validate them, visualize for the end user, and prepare the batch of required parameters for the appropriate stage in processing pipeline. The ThresHolds software in scope of the CoLiTec software was successfully installed as the main astronomical image processing pipeline in the different observatories.

Keywords

Data mining, image processing, processing pipeline, dataflow, configuration parameters, CoLiTec, XML

1. Introduction

There are different tasks for the astronomical image processing and machine vision purposes in astronomy. Some of them are as follows: filtering [1], brightness equalization [2], background alignment [3], object and motion detection [4, 5], astrometry, photometry, images cross-matching, object recognition [6], Wavelet coherence analysis [7] and others. Such astrophysical objects that can be processed and detected in the series of CCD-images [8] are as follows: galaxies, stars, robots [9, 10], drones [11], rockets, satellites [12], and even comets or asteroids [13].

The complex scientific processing pipelines are required to implement the different image processing and machine vision tasks. In common words, the processing pipeline is a set of data processing modules connected in series, where the output of one module is the input of the next one [14]. The modules in pipeline are often executed consequently one by one and rare in parallel or in time-sliced fashion. Also, there are different buffer storage can be inserted between the modules to save the intermediate results or processing data.

There are different processing pipelines in the software engineering and computer science [15]:

ICST-2023: Information Control Systems & Technologies, September 21-23, 2023, Odesa, Ukraine
EMAIL: sergii.khlamov@gmail.com (S. Khlamov); tetiana.trunova@nure.ua (T. Trunova); iryna.tabakova@nure.ua (I. Tabakova)
ORCID: 0000-0001-9434-1081 (S. Khlamov); 0000-0003-2689-2679 (T. Trunova); 0000-0001-6629-4927 (I. Tabakova)

- *instruction pipeline* is a classic pipeline that is used in the microprocessors and central processing units (CPUs) to allow overlapping execution of multiple instructions with the same circuitry. Such circuitry is often divided up into stages and each stage is responsible for the processing of a specific part of one instruction at a time and passing results to the next stage (for example, instruction for the coding/decoding/encoding, logic/arithmetic or register fetch).
- *graphics pipeline* is a pipeline in the most graphics processing units (GPUs), which contains the multiple arithmetic units for implementing the different stages of rendering operations (for example, window clipping, color and light calculation, perspective projection, rendering, etc.).
- *software pipeline* is a sequence of computing processes (program runs, commands, tasks, procedures, threads, etc.) that executed in parallel, with the output stream of one process being automatically fed as the input stream of the next one.

Along with the software processing pipeline the data pipeline, known as a dataflow, is also performed. And when the processing is performed with the big astronomical data along with the configuration parameters of each mathematical and processing module, the data mining approach is very useful [16]. The data mining is an analysis step of the "knowledge discovery in databases" (KDD) process [17]. The data mining carries out about the useful information extracting using the intelligent methods from a data set or configuration parameters set to transform it according to the required contracts and protocols and prepare for the further usage in the processing pipeline.

In this paper we presented a description of the different processing pipelines, selected one of the astronomical scientific software based on such processing pipeline, described the implementation of data mining purposes related to the astronomical processing configuration parameters during invocation the image processing pipeline and its implementation in the developed *ThresHolds* software. It is especially designed as a part of *CoLiTec* software [18] for working with a big amount of the astronomical configuration parameters that are used by the different mathematical and processing modules and components.

2. Processing pipeline in the CoLiTec software

For our research we have selected the astronomical scientific software for detection the moving objects in a series of CCD-frames called “*CoLiTec*”, which implements the image processing pipeline. Such software performs almost all astronomical image processing tasks, like filtering [1, 19], brightness equalization [20], background alignment [20], image stacking/segmentation [21], object detection [4], motion detection [5], object astrometry [22], object photometry [23], object’s image and motion parameters estimation [4, 5, 24], machine (computer) vision [25] of the reference objects to be cataloged [26], object recognition [27] time series analysis [28], Wavelet coherence analysis [29], machine learning recognition [30] and others.

CoLiTec software realizes the different knowledge discovery in databases and data mining approaches, like pre-processing, clustering, classification, identification, processing, summarization. *CoLiTec* software is a very complex astronomical system for the big data sets processing, which includes the different features, user-friendly tools for the processing management, results reviewing, integration with online astronomical catalogs [25] and a lot of computational components and modules that are based on the developed mathematical methods [5, 19, 22]. The high level processing pipeline with the developed modules and implemented methods of the *CoLiTec* software is presented in the Figure 1.

Totally, the *CoLiTec* software [13] consists of more than 30 mathematical and processing modules/components related to the different stages of the image processing that included to the common processing pipeline. Each such module/component has a lot of configuration parameters to be set for the proper image processing and tuning the processing results.

All relationships between processing modules and components in the *CoLiTec* software are based on the predefined contracts as a stable form of the input description for the module processing. Almost all contracts have a fixed structure and description based on the eXtensible Markup Language (XML) [31]. This is a file format with the main purposes to serialize, store, transmit, and reconstruct the different arbitrary data. XML defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

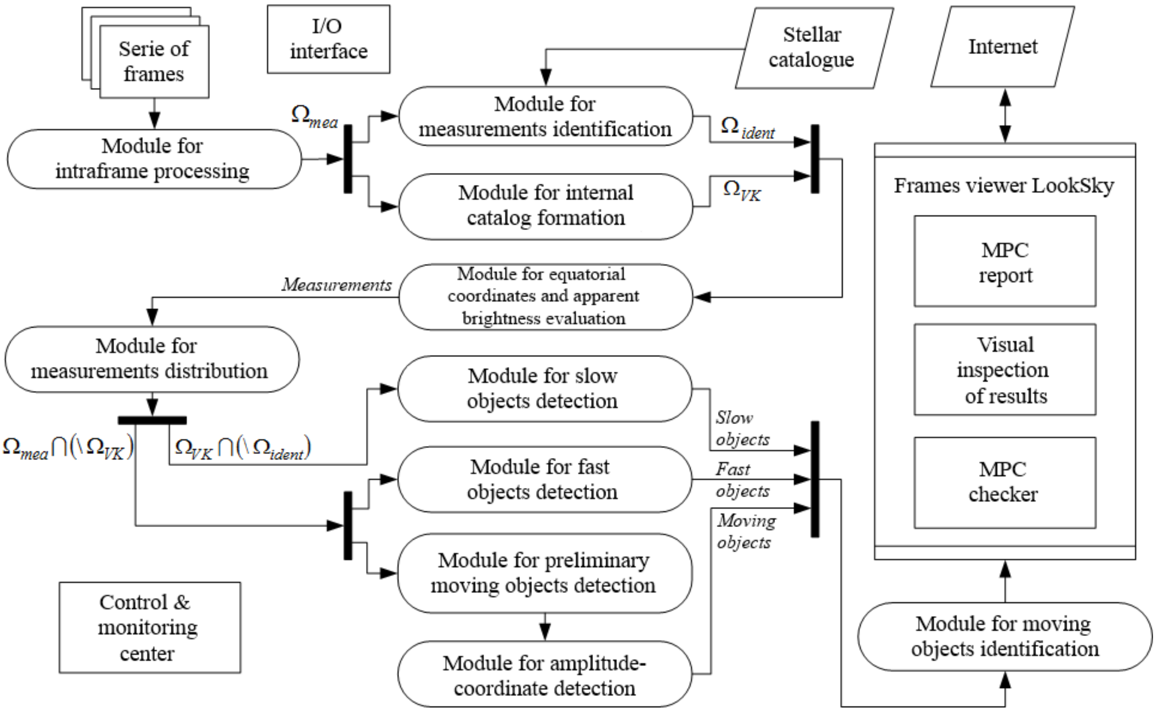

Figure 1: The high level processing pipeline of the *CoLiTec* software

The main design goals of XML are to emphasize simplicity, generality, and usability across the processing modules. For the exchanging of different kind of information between two disparate and separate processing modules, they need to agree upon a file format, and the XML standardizes this process. Generally, the XML file is a file with the textual data format with strong Unicode support and a fixed structure.

Each XML document has a hierarchical structure and is conceptually interpreted as a tree structure, called the XML tree. Such tree should contain the root element (only one parent element of all other elements), sub elements with attributes and text. The example of XML file with the astronomical configuration parameters in the *CoLiTec* software is presented in the Figure 2.

Totally, there are more than 700 astronomical configuration parameters related to the different image processing tasks performed by up to 30 processing modules in the pipeline of *CoLiTec* software.

The structure of such XML configuration file has the restricted rules:

- each element of the XML tree has the required attributes: "*default*" and "*type*";
- "*default*" and "*type*" attributes should not be empty;
- "*type*" attribute should have only fixed values according to the XML Schema Definition (XSD) [32] defined by the World Wide Web Consortium (W3C) [33]: "*xs:string*", "*xs:boolean*", "*xs:integer*", "*xs:decimal*";
- each element should contain the text;
- "*list*" attribute is allowed with coma separator only for element with type "*xs:string*";
- "*min*" and "*max*" attributes are required only for element with types "*xs:integer*" and "*xs:decimal*";
- "*min*" and "*max*" attributes should have the values according to the "*type*" attribute;
- "*node*" attribute should contain only the following values: "*checkbox*", "*textfield*", and "*radiobutton*";
- "*id*", "*label*" and "*node*" attributes are optional, but should not be empty;
- "*id*" attributes should contain the unique value;
- "*label*" attribute should contain the text value.

```xml
<CoLiTec>
  <artifactsFilter>
    <IsDetect default="true" type="xs:boolean">true</IsDetect>
    <NegativePixels default="5.0" max="10.0" min="2.0" type="xs:decimal">5.0</NegativePixels>
    <HotPixels default="5.0" max="10.0" min="2.0" type="xs:decimal">5.0</HotPixels>
  </artifactsFilter>
  <astroPhotoMetry>
    <UseOpticalPolynom default="false" type="xs:boolean">false</UseOpticalPolynom>
    <MarksFromFrame default="50" max="100" min="30" type="xs:integer">50</MarksFromFrame>
    <PercentOfMatching default="70" max="80" min="60" type="xs:integer">70</PercentOfMatching>
    <StarsResearch default="0.0" max="1.0" min="0.0" type="xs:decimal">0.0</StarsResearch>
    <LimitOfMagnitude default="25.0" max="25.0" min="15.0" type="xs:decimal">25.0</LimitOfMagnitude>
    <DifferentBetweenAngles default="3" max="10" min="1" type="xs:decimal">3</DifferentBetweenAngles>
    <DistanceTo3DVertex default="0.3" max="0.9" min="0.1" type="xs:decimal">0.3</DistanceTo3DVertex>
    <AstroReductionDirection default="XYtoRADE" list="RADEtoXY,XYtoRADE" type="xs:string">XYtoRADE</AstroReductionDirection>
    <AstrometryReductionDegree default="3" list="1,3,5" type="xs:integer">3</AstrometryReductionDegree>
    <ReductionDegree default="1" list="1,3,5" type="xs:integer">1</ReductionDegree>
    <OpticalPolynomDegree default="5" list="1,3,5" type="xs:integer">5</OpticalPolynomDegree>
    <StarsRejectionThreshold default="0.1" max="10.0" min="0.0" type="xs:decimal">0.1</StarsRejectionThreshold>
    <PairReductionCoefficient default="1.0" max="99.9" min="0.1" type="xs:decimal">1.0</PairReductionCoefficient>
    <FragmentsNumber default="4" max="1000" min="1" type="xs:integer">4</FragmentsNumber>
    <AmplitudeSort default="A2" list="A2,A3" type="xs:string">A2</AmplitudeSort>
    <UseAstrocatalogErrors default="false" type="xs:boolean">false</UseAstrocatalogErrors>
    <DoIC default="false" type="xs:boolean">false</DoIC>
    <DoNotSelectStarsOnFrame default="(0;0-0;0)(0;0-0;0)" type="xs:string">(0;0-0;0)(0;0-0;0)</DoNotSelectStarsOnFrame>
    <CriticalSNR3forMeasurements default="5.0" max="999.9" min="0.1" type="xs:decimal">5.0</CriticalSNR3forMeasurements>
    <MinPixelScale default="1.0" max="180.0" min="1.0" type="xs:decimal">1.0</MinPixelScale>
    <MaxPixelScale default="180.0" max="180.0" min="0.1" type="xs:decimal">180.0</MaxPixelScale>
  </astroPhotoMetry>
  <brightnessFramesAlignment>
    <MedianWindow default="101" max="500" min="50" type="xs:integer">101</MedianWindow>
    <BinningCoefficient default="3" max="16" min="1" type="xs:integer">3</BinningCoefficient>
    <SpecifiedMasterFrames default="false" type="xs:boolean">false</SpecifiedMasterFrames>
    <UseCommonFolder default="false" type="xs:boolean">false</UseCommonFolder>
    <ProcessingRawFolder default="" type="xs:string"></ProcessingRawFolder>
    <UseMasterBias default="false" type="xs:boolean">false</UseMasterBias>
    <ProcessingRawBias default="" type="xs:string"></ProcessingRawBias>
    <UseMasterDark default="false" type="xs:boolean">false</UseMasterDark>
    <ProcessingRawDark default="" type="xs:string"></ProcessingRawDark>
    <UseMasterDarkFlat default="false" type="xs:boolean">false</UseMasterDarkFlat>
    <ProcessingRawDarkFlat default="" type="xs:string"></ProcessingRawDarkFlat>
    <UseMasterFlat default="false" type="xs:boolean">false</UseMasterFlat>
    <ProcessingRawFlat default="" type="xs:string"></ProcessingRawFlat>
    <UseFHF default="true" type="xs:boolean">true</UseFHF>
    <IgnoreTimeFactor default="false" type="xs:boolean">false</IgnoreTimeFactor>
    <ExposureAutoDetermination default="false" type="xs:boolean">false</ExposureAutoDetermination>
  </brightnessFramesAlignment>
```


Figure 2: The example of XML file with the astronomical configuration parameters in the *CoLiTec* software

3. Data mining by the ThresHolds software

Under the research in scope of the *CoLiTec* project [18] we have developed the *Telescope* software for the following data mining and processing tasks:

- mining the astronomical processing configuration parameters;
- classification, managing and validating of the astronomical processing configuration parameters;
- visualization for the end user;
- preparing the batch of required configuration parameters for the appropriate stage in processing pipeline of the big astronomical data from the different storages and archives.

3.1. Technical implementation

The *ThresHolds* software realized the different data mining tasks, like receiving, storing, selecting, preprocessing, transforming, useful data extraction, classification, and knowledge discovery in databases (KDD) [17]. The following stack of technologies were used for the software development: Java programming language [34], JavaFX technology for graphical user interface (GUI), Maven for compilation and building, Java Development Kit (JDK) [35] and XML [31].

As a JDK the developers selected the Open JDK as an open-source and free Java platform including the Java machine, which is supported by the Java community instead of the Oracle JDK. The Open JDK plays the role as a platform for developing the client applications for desktop, laptop, and tablet PCs and is a cross-platform, so it supports the different operational systems (OSs), like Windows, Linux, MacOS.

The *ThresHolds* software has a GUI for visualization of the processing configuration parameters for the end user and the additional useful features described below.

3.2. Data mining tasks

The *ThresHolds* software is designed for the full integration with astronomical image processing pipeline of the *CoLiTec* software and performs the following data mining tasks:

- recurrency searching for the configuration XML files;
- visualization by the dynamically creating the GUI according to the data in the configuration XML file;
- managing the astronomical configuration parameters by the end user using the GUI (Figure 3 and Figure 4);

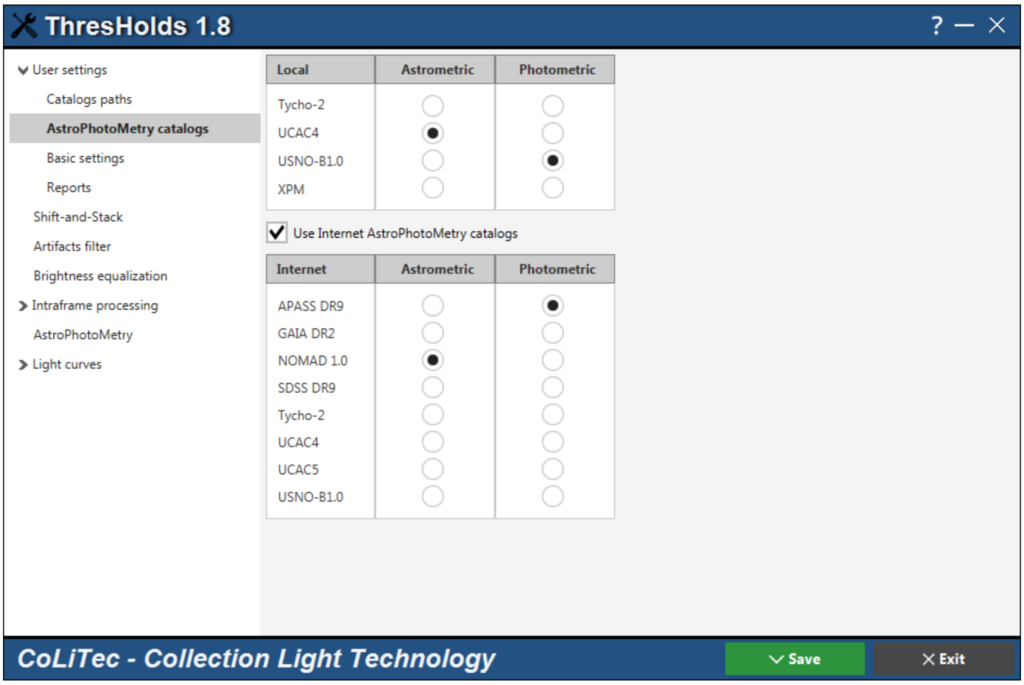

Figure 3: The "*AstroPhotoMetry catalogs*" section of the *ThresHolds* software

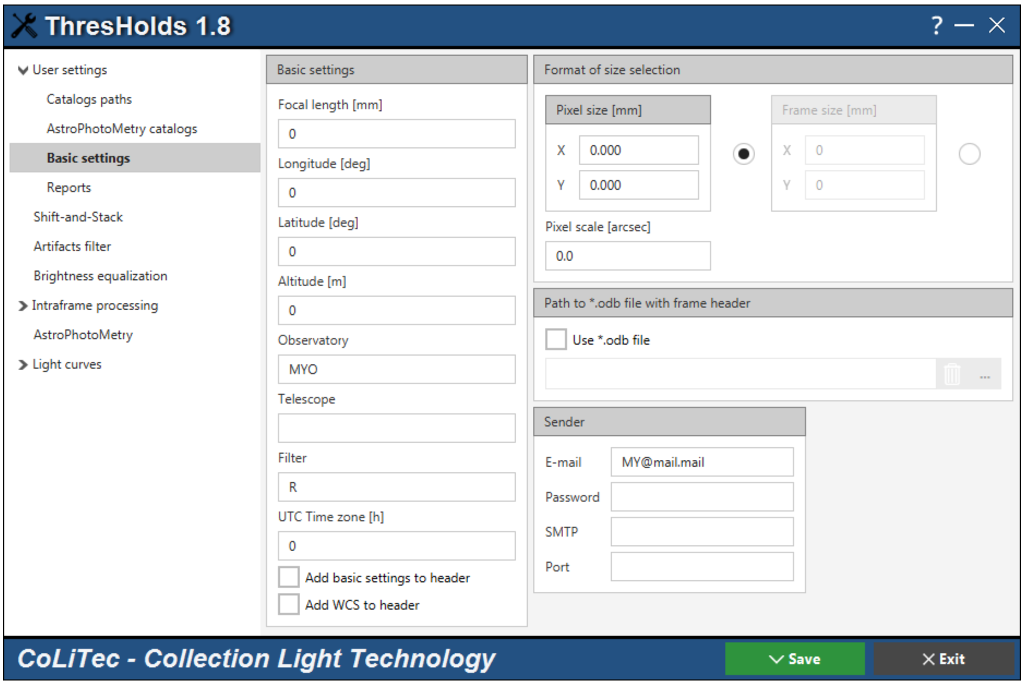

Figure 4: The "*Basic settings*" section of the *ThresHolds* software

•	mining the astronomical processing configuration parameters from the modified XML file (see Figure 2);
•	classifying the astronomical configuration parameters according to the different image processing tasks and stages in the processing pipeline (see Figure 1);
•	validating the astronomical configuration parameters according to the restricted rules described in section 2;
•	dynamically generating the batch of required parameters for the appropriate image processing tasks and stages in the processing pipeline (see Figure 5).

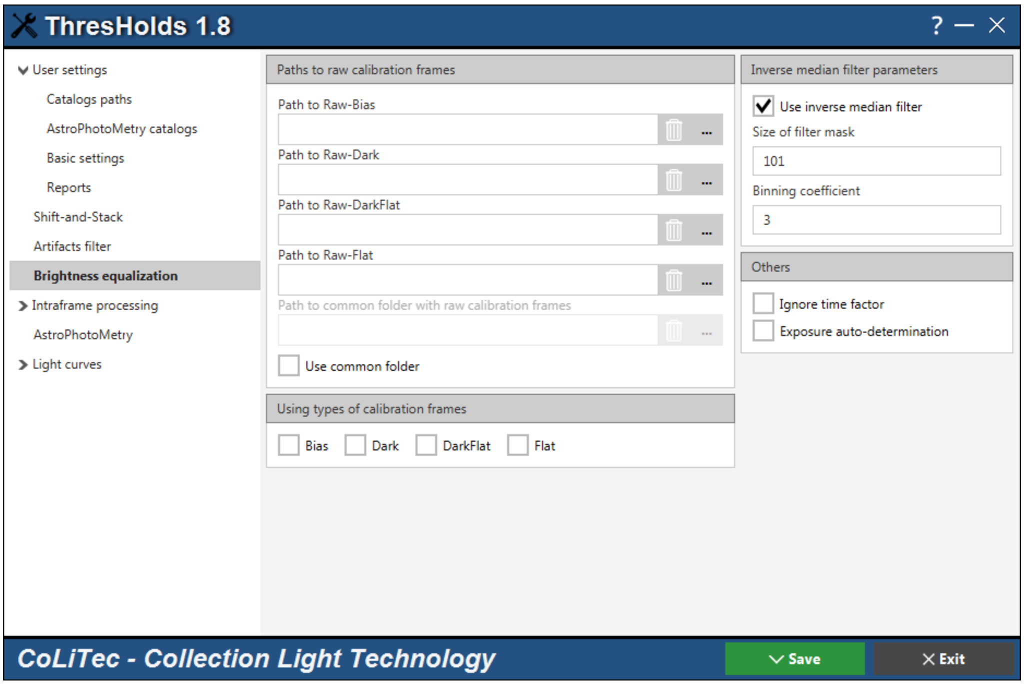

Figure 5: The "*Brightness equalization*" section of the *ThresHolds* software

For example, to perform the brightness equalization by the inverse median filter in the processing pipeline, the *ThresHolds* software prepares the batch of the following required parameters for processing:

•	"*Binning coefficient*" to perform the median filtering with binning image. Such result is de-binning and subtracted from the original image.
•	"*Size of filter mask*" as a width and height of the square median filter mask m. The recommended value of it is determined by the following equation:

$$m \geq \sqrt{3\pi}R, \qquad (1)$$

where R is an image radius of the brightest object by $2\hat{\sigma}_{noise}$ level of background; $\hat{\sigma}_{noise}$ is a root-mean-square (RMS) evaluation of the background brightness.
•	"*Do image inversion*" – activates the image inversion function during transformation.
•	"*Make crop*" – activates the crop creation function. It defines by the coordinates of the upper left vertex, the width and height.
•	Crop parameters can be set in "*X*", "*Y*", "*Width*" and "*Height*" fields.
•	"*Output mask*" – mask for file names of the processed images.
•	"*Process frames by RGB channels*" allows processing a color image by channels. It does not affect on filtering of gray images and FITS files [36];
•	"*Path to input file(s)*" – path to the raw files.
•	"*Path to output folder*" – path to the output folder of processed files.
•	Option for creating and using of the master-frames.

- Paths of the created master-frames will be inserted to the appropriated fields:
    a. For Bias – “*Path to Master-Bias*”.
    b. For Dark – “*Path to Master-Dark*”.
    c. For DarkFlat – “*Path to Master-DarkFlat*”.
    d. For Flat – “*Path to Master-Flat*”.
- “*Mask after subtraction*” – a prefix to the name of the target (Light) frames after subtraction operation;
- “*Mask after division*” – a prefix to the name of the target (Light) frames after division operation;
- “*Pixels rejection RMS*” – coefficient of the pixel’s rejection in an operation of master-frames creation.

An example of the XML configuration file for the mathematical module of the inverse median filtering formed but the *ThresHolds* software as a batch of the required parameters for processing is presented in the Figure 6.


```
<FhfSettings>
  <highpassFilter>
    <IsHighpassFilter>0</IsHighpassFilter>
    <FullNameFitInFilter>E:\Ubuntu15\refhflinux\flat-002R.fit</FullNameFitInFilter>
    <FullNameFitOutFilter>E:\Ubuntu15\refhflinux\_fhf99_flat-002R.fit</FullNameFitOutFilter>
    <DayFit>0</DayFit>
    <RadiusFilter>100</RadiusFilter>
    <NumberIter>12</NumberIter>
    <KSkoFilter>0</KSkoFilter>
  </highpassFilter>
  <crop>
    <IsCrop>0</IsCrop>
    <FullNameFitInCrop>E:\Ubuntu15\refhflinux\flat-002R.fit</FullNameFitInCrop>
    <PathFitOutCrop>E:\Ubuntu15\refhflinux\flat-002R_crop\</PathFitOutCrop>
    <SizeUp>200</SizeUp>
    <SizeDown>200</SizeDown>
    <SizeLeft>200</SizeLeft>
    <SizeRight>200</SizeRight>
    <Invert>0</Invert>
    <Crop>3</Crop>
    <CoordinateX>1000</CoordinateX>
    <CoordinateY>2000</CoordinateY>
    <WidthCrop>1000</WidthCrop>
    <HeightCrop>2000</HeightCrop>
  </crop>
  <inversMedianFilter>
    <IsInversMedianFilter>1</IsInversMedianFilter>
    <FullNameFitInInversMedianFilter>e:\Ubuntu15\refhflinux\RXJ1803 -114R.fit</FullNameFitInInversMedianFilter>
    <FullNameFitOutInversMedianFilter>e:\Ubuntu15\refhflinux\RXJ1803 -114R_w51_b2.fit</FullNameFitOutInversMedianFilter>
    <InversMedianWindow>51</InversMedianWindow>
    <ColorOrBw>0</ColorOrBw>
    <SmallOrBig>0</SmallOrBig>
    <Binning>2</Binning>
  </inversMedianFilter>
 </FhfSettings>
```


Figure 6: The formed XML configuration file for the mathematical module of the inverse median filtering

As mentioned above in the Figure 1, the *ThresHolds* software is included into the high level processing pipeline of the *CoLiTec* software in automated and autonomous mode. Each such mentioned module and submodule has its own set of configuration parameters, and the main data mining task of the *ThresHolds* software is to collect appropriate data and send it to the processing pipeline. In this case, it is a realization of the subject mediator approach.

The *ThresHolds* software operates with astronomical configuration parameters, which are equal to the scientific constants or variables of the different mathematical methods and algorithms for the astronomical image processing.

The high-level diagram of the developed algorithm for such data mining realized in the *ThresHolds* software is presented below in the Figure 6.

The developed algorithm for data mining tasks realized in the *ThresHolds* software contains the following main steps.

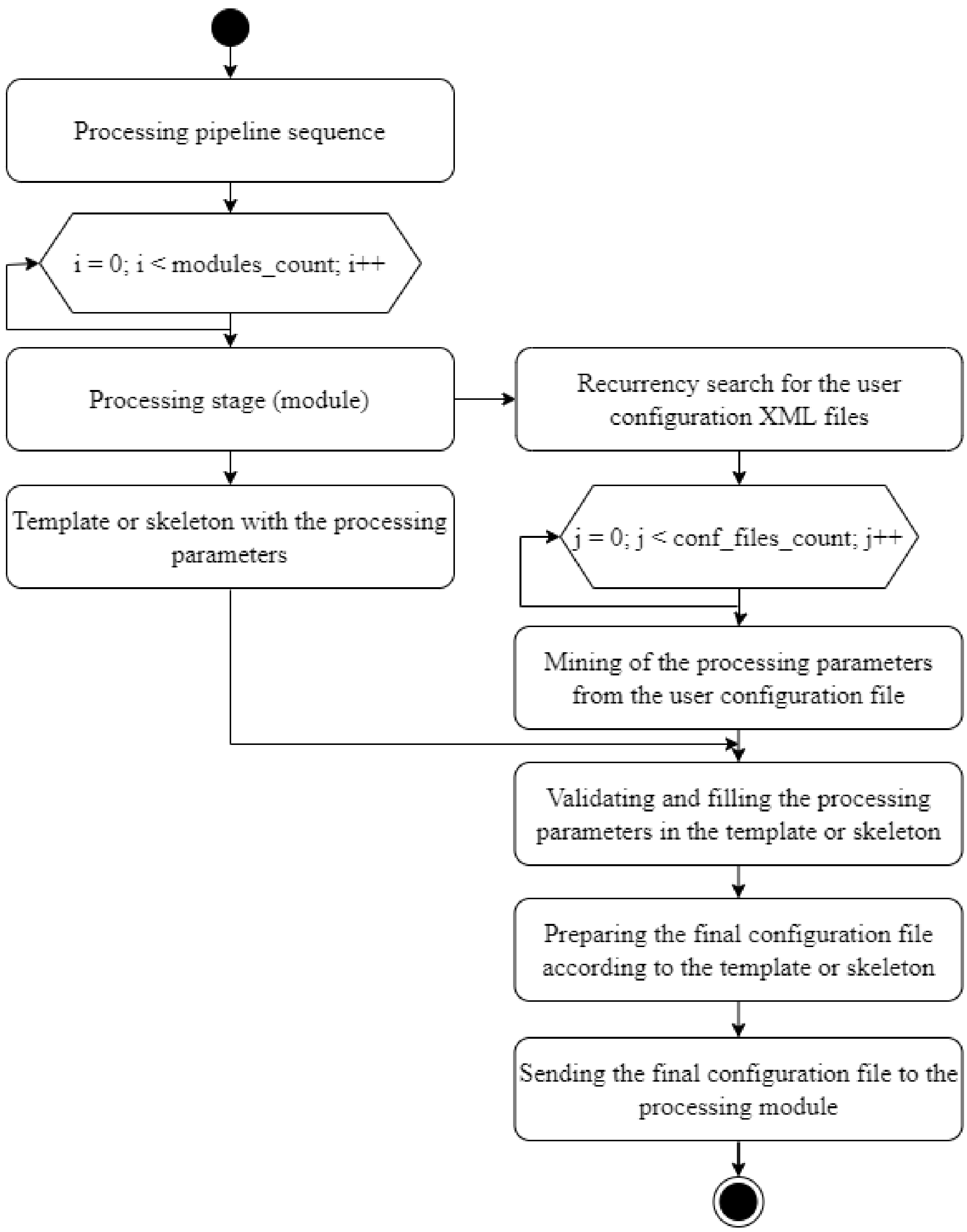

Figure 7: The high-level diagram of the developed algorithm for data mining realized in the *ThresHolds* software

1. Processing pipeline selects the next module, which is predefined in a sequence.
2. Such processing module has the appropriate template or skeleton for the configuration file with processing parameters required for it.
3. Classification of the astronomical configuration parameters according to the different image processing tasks and stages in the processing pipeline.
4. Recurrency search for the different user configuration XML files prepared by user before processing starts.
5. Processing parameters mining from the selected user configuration XML file according to the template or skeleton.
6. Validating of the astronomical configuration parameters according to the restricted rules described in section 2;
7. Dynamically generating the batch of required parameters according to the template or skeleton for the appropriate image processing tasks and stages in the processing pipeline.
8. Processing pipeline selects the next module, which is predefined in a sequence.

3.3. Practical implementation

The *ThresHolds* software in scope of the *CoLiTec* software was installed in the different observatories (Mayaki Astronomical Observatory [37], ISON-NM and ISON-Kislovodsk observatories, Vihorlat Observatory [2, 20]), astronomical archives [38], and Ukrainian Virtual Observatory (UkrVO) [39]. An example of data mining of the astronomical processing configuration parameters by the *ThresHolds* software integrated into the processing pipeline is presented in the Figure 8.

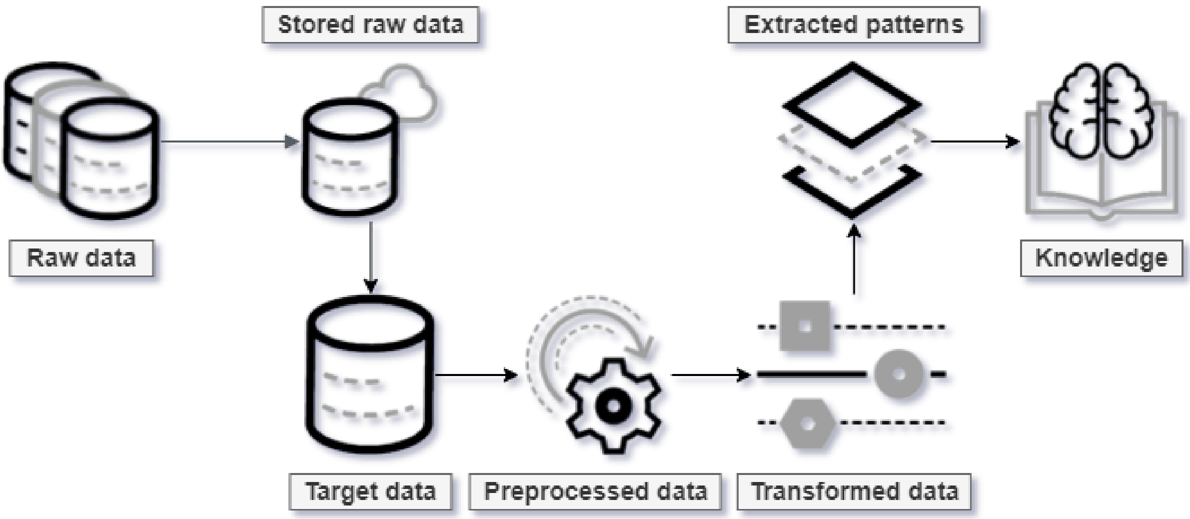

Figure 8: Example of data mining of the astronomical processing configuration parameters by the *ThresHolds* software integrated into the processing pipeline

The more detailed information about the observatories, telescopes equipped by the different CCD-cameras as well as their parameters are provided below.

The Mayaki observing station of "Astronomical Observatory" Research Institute of I. I. Mechnikov Odessa National University has the 0.48 m AZT-3 telescope – reflector with focal length 2025 mm and CCD-camera Sony ICX429ALL (resolution 795×596).

The observatory "ISON-NM observatory" has the 0.4 m SANTEL-400AN telescope with CCD-camera FLI ML09000-65 (3056×3056 pixels, 12 microns).

The observatory "ISON-Kislovodsk" has the 19.2 cm wide-field GENON (VT-78) telescope with CCD-camera FLI ML09000-65 (4008×2672 pixels, 9 microns).

The observatory "Vihorlat Observatory in Humenne" has the Vihorlat National Telescope (VNT) – Kassegren telescope with 1 m main mirror with focal length 8925 mm and CCD-camera FLI PL1001E (512×512 pixels).

The Vihorlat Observatory also has the Celestron C11 telescope – Schmidt-Cassegrain telescope with 28 cm main mirror with focal length 3060 mm and CCD-camera G2-1600 (resolution 768×512 pixels).

The data mining of astronomical processing configuration parameters by the *ThresHolds* software was performed during the processing of up to 1 million astronomical files both archived and original formed from the different telescopes.

4. Conclusions

The *ThresHolds* software with the realization of the developed algorithm for data mining purposes related to the astronomical processing configuration parameters during invocation the image processing pipeline was developed. The software is implemented using the Java programming language and JavaFX technology. The advantages of such selected programming language is an open-source libraries and cross platform approach.

The *ThresHolds* software was developed for the full integration with the astronomical image processing pipelines with a GUI for visualization of more than 700 processing configuration parameters. The main goals of the ThresHolds software are the mining of astronomical processing configuration parameters, their classification, managing and validation. Also, the visualization for end user is available as well as preparation the batch of required parameters for the appropriate stage in processing pipeline of the big astronomical data from the different storages and archives.

Research showed that using the developed algorithm for data mining purposes the *ThresHolds* software execution time speeds up in comparison to the common algorithms for the data processing without optimization and mediator subject approach. In total, based on the statistical [40] experiments result is more than 30% of speeding up of the processing time of the whole pipeline and the whole sequence of the astronomical scientific data processing.

The *ThresHolds* software was developed as a part of the *CoLiTec* software [18]. It was tested during several years after successful installation in scope of the astronomical image processing pipelines on the different observatories.

5. Acknowledgements

The authors thank all observatories for their collaboration and possibility to install the *ThresHolds* software in scope of the *CoLiTec* software as a main astronomical processing pipeline to conduct the current research and test the developed software.

6. References

[1] C. A. Politsch, et al., Trend filtering–I. A modern statistical tool for time-domain astronomy and astronomical spectroscopy, MNRAS, 492 3 (2020) 4005-4018. DOI: 10.1093/mnras/staa106.

[2] Š. Parimucha, et al., CoLiTecVS – A new tool for an automated reduction of photometric observations, Contributions of the Astronomical Observatory Skalnate Pleso, 49 2 (2019) 151-153

[3] E. Corsaro, et al., Spin alignment of stars in old open clusters, Nature Astronomy, 1 4 (2017) 1-6. doi: 10.1038/s41550-017-0064.

[4] L. Mykhailova, et al., Method of maximum likelihood estimation of compact group objects location on CCD-frame, Eastern-European Journal of Enterprise Technologies, 5 4 (2014) 16-22. doi:10.15587/1729-4061.2014.28028.

[5] S. Khlamov, et al., Development of computational method for detection of the object's near-zero apparent motion on the series of CCD–frames, Eastern-European Journal of Enterprise Technologies, 2 80 (2016) 41-48. doi: 10.15587/1729-4061.2016.65999.

[6] R. Klette, Concise computer vision, Springer, London, 2014.

[7] M. Dadkhah, et al., Methodology of wavelet analysis in research of dynamics of phishing attacks, International Journal of Advanced Intelligence Paradigms, 12 (2019) 220-238. doi: 10.1504/IJAIP.2019.098561.

[8] G. K. Adam, et al., Embedded microcontroller with a CCD camera as a digital lighting control system, Electronics, 8 1 (2019). doi: 10.3390/electronics8010033.

[9] A. Tantsiura, et al., Evaluation of the potential accuracy of correlation extreme navigation systems of low-altitude mobile robots, International Journal of Advanced Trends in Computer Science and Engineering, 8 5 (2019) 2161–2166. doi: 10.30534/ijatcse/2019/47852019.

[10] N. Yeromina, V. Tarshyn, S. Petrov, et al., Method of reference image selection to provide high-speed aircraft navigation under conditions of rapid change of flight trajectory, International Journal of Advanced Technology and Engineering Exploration, 8 85 (2021) 1621–1638. doi: 10.19101/IJATEE.2021.874814.

[11] M. Ivanov, et al., Effective informational entropy reduction in multi-robot systems based on real-time TVS, IEEE International Symposium on Industrial Electronics, 2019, pp. 1162–1167. doi: 10.1109/ISIE.2019.8781209.

[12] V. Akhmetov, et al., Cloud computing analysis of Indian ASAT test on March 27, 2019, Proceedings of the 2019 IEEE International Scientific-Practical Conference: Problems of

Infocommunications Science and Technology, PIC S and T 2019, 2019, pp. 315–318. doi: 10.1109/PICST47496.2019.9061243.
[13] S. Khlamov, et al., Data Mining of the Astronomical Images by the CoLiTec Software, CEUR Workshop Proceedings, 3171, 2022, pp. 1043–1055.
[14] A. Jangda, U. Bondhugula, An effective fusion and tile size model for optimizing image processing pipelines, ACM SIGPLAN Notices, 53 1 (2018), pp. 261-275. doi: 10.1145/3200691.3178507.
[15] J. K. Li, K.-L. Ma, P5: Portable progressive parallel processing pipelines for interactive data analysis and visualization, IEEE transactions on visualization and computer graphics, 26 1 (2019) 1151-1160. doi: 10.1109/TVCG.2019.2934537.
[16] Y. Zhang, Y. Zhao, C. Cui, Data mining and knowledge discovery in database of astronomy, Progress in Astronomy, 20 4 (2002) 312-323.
[17] Q. Luo, Advancing knowledge discovery and data mining, First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), IEEE, 2008, pp. 3-5. doi: 10.1109/WKDD.2008.153.
[18] S. Khlamov, V. Savanevych, Big astronomical datasets and discovery of new celestial bodies in the Solar System in automated mode by the CoLiTec software, Knowledge Discovery in Big Data from Astronomy and Earth Observation, Astrogeoinformatics: Elsevier, 2020, pp. 331–345. doi: 10.1016/B978-0-12-819154-5.00030-8.
[19] S. Khlamov, V. Vlasenko, V. Savanevych, O. Briukhovetskyi, T. Trunova, V. Chelombitko, I. Tabakova, Development of computational method for matched filtration with analytic profile of the blurred digital image, Eastern-European Journal of Enterprise Technologies, 5 119 (2022) 24–32. doi: 10.15587/1729-4061.2022.265309.
[20] I. Kudzej, et al., CoLiTecVS – A new tool for the automated reduction of photometric observations, Astronomische Nachrichten, 340 (2019) 68–70. doi: 10.1002/asna.201913562.
[21] V. Savanevych, et al., Formation of a typical form of an object image in a series of digital frames, Eastern-European Journal of Enterprise Technologies, 6 120 (2022) 51–59. doi: 10.15587/1729-4061.2022.266988.
[22] V. Akhmetov, et al., Astrometric reduction of the wide-field images, Advances in Intelligent Systems and Computing, 1080 (2020) 896–909. doi: 10.1007/978-3-030-33695-0_58.
[23] V. Savanevych, et al., CoLiTecVS software for the automated reduction of photometric observations in CCD-frames, Astronomy and Computing, 40 (2022) 15. doi: 10.1016/j.ascom.2022.100605.
[24] R. Gonzalez, R. Woods, Digital image processing, 4th edition, NY: Pearson, 2018.
[25] V. Akhmetov, et al., New approach for pixelization of big astronomical data for machine vision purpose, IEEE International Symposium on Industrial Electronics, 2019, pp. 1706–1710. doi: 10.1109/ISIE.2019.8781270.
[26] V. Savanevych, et al., Selection of the reference stars for astrometric reduction of CCD-frames, Advances in Intelligent Systems and Computing, 1080 (2020) 881–895. doi: 10.1007/978-3-030-33695-0_57.
[27] C. Steger, M. Ulrich, C. Wiedemann, Machine vision algorithms and applications, John Wiley & Sons, 2018.
[28] L. Kirichenko, A.S.A. Alghawli, T. Radivilova, Generalized approach to analysis of multifractal properties from short time series, International Journal of Advanced Computer Science and Applications, 11 5 (2020) 183–198. doi: 10.14569/IJACSA.2020.0110527.
[29] V. Baranova, et al., Wavelet Coherence as a Tool for Studying of Economic Dynamics in Infocommunication Systems, IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 336-340. doi: 10.1109/PICST47496.2019.9061301.
[30] L. Kirichenko, P. Zinchenko, T. Radivilova, Classification of time realizations using machine learning recognition of recurrence plots, Advances in Intelligent Systems and Computing, 1246 (2021) 687–696. doi: 10.1007/978-3-030-54215-3_44.
[31] M. Wang, J. Wang, K. Guo, Extensible markup language keywords search based on security access control, International Journal of Grid and Utility Computing, 9 1 (2018) 43-50. doi: 10.1504/IJGUC.2018.090227.

[32] S. Guo, H. Xia, G. Xiang, Research on the translation from XSD to JSON schema, IEEE 9th International Conference on Communication Software and Networks (ICCSN), 2017, pp. 1393-1396. doi: 10.1109/ICCSN.2017.8230338.
[33] T. Brooks, World Wide Web Consortium (W3C), Encyclopedia of Library and Information Sciences, CRC Press, 2017.
[34] G. O'Regan, Java programming language, The Innovation in Computing Companion. Springer, Cham, 2018. doi: 10.1007/978-3-030-02619-6_35.
[35] D. Winnie, Setting Up the Java JDK and IntelliJ, Essential Java for AP CompSci. Apress, Berkeley, CA, 2021. doi: 10.1007/978-1-4842-6183-5_2.
[36] W. D. Pence, L. Chiappetti, C. G. Page, et al., Definition of the flexible image transport system (fits), 2010. doi: 10.1051/0004-6361/201015362.
[37] S. Kashuba, et al., The Simeiz plate collection of the ODESSA astronomical observatory, Proceedings of the 11th Bulgarian-Serbian Astronomical Conference, 2018, pp. 207-216.
[38] I. B. Vavilova, et al., Surveys, catalogues, databases, and archives of astronomical data, Knowledge Discovery in Big Data from Astronomy and Earth Observation, Astrogeoinformatics: Elsevier, 2020, pp. 57-102. doi: 10.1016/B978-0-12-819154-5.00015-1.
[39] I. B. Vavilova, et al., Astrometry and photometry of asteroids from the UkrVO database of astroplates, Proceedings of the IAU, vol. 15, issue S364, 2021, pp. 239-245. doi: 10.1017/S1743921322000047.
[40] V. Shvedun, S. Khlamov, Statistical modelling for determination of perspective number of advertising legislation violations, Actual Problems of Economics, 184 10 (2016) 389-396.