Automatic analysis of video content in the process of monitoring of industrial facilities

M Jura
KOMAG Institute of Mining Technology, Division of Mechatronic Systems,
Pszczyńska 37, 44-101 Gliwice, Poland

E-mail: mjura@komag.eu

Abstract. In order to automate the monitoring process, a review of methods of object detection in pictures or video content has been carried out. The methods have been analyzed in terms of their application in the software design, which will be implemented for monitoring with the use of UAVs (unmanned aerial vehicles). The selected methods have been chosen to create prototype software for finding video frames depicting a specific object and archiving the results in relation to this object.

1. Introduction

An adoption time of new technologies in the mining industry is longer than in other branches of industry as the requirements of this industry are strict due to harsh environment of mines. But recently the mining industry increased use of new technologies. Especially a fast adaption can be seen in information technologies as they can often be based on existing hardware. One of the most prominent technologies that will increase efficiency of many industrial processes is artificial intelligence [1, 2, 3, 4]. Another new technology, recently used in mining, is the increasingly popular technique of identifying objects through image analysis. It is used especially in the remote inventory of large industrial installations and stored resources.

Video surveillance is an important part of the monitoring of industrial facilities, processes or machines. It allows to monitor the status of supervised objects without the need to stay close to them. Monitoring systems allow for archiving data and access to it by the user, taking into account the time of registration. Monitoring is conducted continuously or on cycles (e.g. with use of unmanned aerial vehicle), depending on the specificity of monitored objects. A significant disadvantage of video surveillance systems is the large amount of data they generate and the need for analysis by the user. The next step in the monitoring development could be partial or complete automation of the analysis of the collected video content.

In the case of using unmanned aerial vehicles (UAV) in the monitoring process, the obtained video images usually present many monitored industrial facilities, installations or machines. Unambiguous detection and identification of specific devices or phenomena will be possible due to automatic processing of video content by means of a proposed computer program.

Available methods of recognition of objects in images from video files have been reviewed. On the basis of the review and specific requirements for processing video files, the computer application concept has been developed. The individual frames concept also includes an analysis of the evaluating possibility a video file in terms of their use in monitoring.

The prototype software has been developed to detect images showing a specific object and archive the results in relation to this object. The software allows to:
- recognize selected objects,
- save images (object and time of shooting),
- access saved images by the user.

The prototype software uses programming libraries to process the data received. The OpenCV library is used for processing raster graphics and the ZBar library for decoding barcodes.

2. Review of available methods

2.1. Recognition of keypoints

Keypoint recognition is a computational process designed to determine the abstraction presenting image information and to determine for each point of the image whether that point can be classified as keypoint. Detection of keypoints and comparison of images is one of the important problems in machine vision and robotics. These methods are used in many fields of science and industry. The ideal technique for a detection of keypoints should be resistant to such image transformations as rotation, scaling, brightening, noise and affine transformations. Additionally, the keypoints must differentiate from each other in order to increase the probability of the matches [5, 6].

SIFT (Scale Invariant Feature Transform) is an algorithm for the detection of keypoints created by David Lowe in 2004 [7]. SIFT has proved to be a very efficient for object recognition tasks. However, it requires high computational complexity, which is problematic especially in real time applications [8, 9]. However, there are many modifications to this algorithm that reduce its computational complexity [9, 10, 11]. SIFT allows to solve problems with rotation, brightness, affine transformations and observation point location [7].

SURF (Speed up Robust Feature) is an algorithm very similar to SIFT, but faster and does not decrease the quality of searched points [12]. The DoG (Difference of Gaussians) algorithm used in SIFT has been replaced by a kernel. This is a rough approximation, however, it allows for much faster calculations. SURF uses BLOB detection, which is based on Hessian matrix, to detect keypoints [11].

ORB (Oriented FAST and Rotated BRIEF) is another alternative to SIFT and SURF proposed by Ethan Rublee [13]. ORB is a combination of FAST detector and Binary Robust Independent Elementary Features (BRIEF) algorithm [14]. Initially FAST is used to determine the position of keypoints. Then the Harris Corner Detector is used to limit the number of the points. FAST does not allow calculation of orientation and rotation. It calculates centroids based on the brightness of area with a corner in the middle. The direction from the corner to the centroid gives the orientation. Indicators are calculated to ensure independence from rotation. The BRIEF algorithm does not give good results if there is a rotation in the image plane. In ORB the matrix rotation is calculated according to the orientation of the previously used area and the BRIEF is set according to this orientation.

When considering the detection of markers, first should be considered what kind of markers are to be recognized. It is also possible to approach the problem from the other side and design a marker specifically for a particular task, which could have a positive effect on the performance of that task.

Markers that can be easily recognized with image processing are mainly barcodes. One-dimensional codes have been designed for laser readers in mind, which allowed for the registration of the image in line. Use of matrix barcodes allows to save more data because codes made in two dimensions, increase their bit volume with a square of side length. An additional advantage of some of the matrix barcodes are special tags, which make it easier to find the code and sometimes even its orientation in 3D space. Examples of matrix barcodes are presented in Figure 1. The advantage of using these codes is the availability of libraries enabling decoding them.
2.2. Artificial neural networks

Popularity of artificial neural networks has been growing as they can be need for an increasing number of problems to be solved. They most often used for problems that are solved easily by people but do not have satisfactory solutions using classical algorithms. For kind of problems artificial neural networks go through a process of learning with supervision. When the problem is complicated or the human supervision leads to unsatisfactory results, it is recommended to use unsupervised learning. The learning network must be structured in such a way that it can see higher-order abstractions in the input data. Because the networks, then learned in this way do not aspire to a given result, they can detect irregularities and dependencies that were not known before. This makes them useful for analyzing phenomena whose model is incomplete or inaccurate [15, 16].
Google Inception V3 is a neural network used by Google to recognize objects in images. Figure 2 shows the structure of this network. For the purpose of using it in monitoring, a similar network can be applied but based on a learning set mainly containing monitored objects. A big disadvantage of such a solution would be the need to teach the network every time when adding new objects. However it will be beneficial to use the output of the network, as it will allow to compare several obtained images and to select one that is closest to the one used in the learning set.

2.3. Analysis of a point cloud
Point clouds are sets of points represented in 3D space. Typically each point in a set is assigned three values corresponding to three coordinates X, Y and Z. Depending on the method used, it is also possible to save the color of such a point. Point clouds can be obtained with laser scans or by processing photographs. In case of using UAVs, only the method with the use of photos is available. There is a large selection of software available on the market that calculates point clouds. Examples of such software include Autodesk ReCap (Figure 3), 3DF Zephyr and Selva3D. The software is based on searching for similarities on at least three images and a triangulation to obtain their position in space.
If point clouds are used, a comparison of scanned objects seems to be the main problem. Some of the programs used to create 3D objects, can also allow to combine these models by finding common features on both models. However, they do not allow to search for smaller objects on the model using this method.

There is also a dedicated software for comparing point clouds. Cloud Compare allows you to compare two point clouds together with their visualization (Figure 4). This software calculates the distances between the two models and presents them as a color model with a legend describing the distances assigned to the colors. Then mean distance and standard deviation are also given. This computer however, does not allow for a fully automatic approach. The approach, used in this application, could also be used in an automatic process.

**Figure 3.** Autodesk ReCap [autodesk.com].
Figure 4. The result of the comparison in Cloud Compare [cloudcompare.org].

It seems possible to use point clouds to search for objects in a scanned model but it requires the creation of your own automated methods for comparing point clouds. An additional limitation is also need for scanning in closed spaces and high computational complexity of the required calculations.

3. Software

3.1. Requirements

A review of available methods of recognizing objects in images serves the purpose of aiding the process of designing software that will be used to monitor the objects selected by the user applying UAVs with video recorders.

The software should allow for an introduction of recorded material in a form of photos or video files. These files can be used entirely by a desktop application or in the case of an application using client-server architecture they can be sent to the server. Subsequent frames of video or photos should be automatically analyzed in search of monitored objects. Due to this, each image will be matched with the information about a detected object and optionally with the information such as perspective, which will allow to choose the best image for monitoring purposes. It is also possible to process output images to match the purposes in a better way. An example of processing can be perspective correction or contrast change.

To speed up development the software should use programming libraries to process the received data. Depending on the selected method of object recognition, it can be OpenCV library for processing raster images or PDAL library for processing point clouds.

When saving images, additional information should also be assigned to the image. The information should contain the name of the object or its subpart that has been chosen for monitoring. It is also necessary to save date and time of registration. This information can be acquired using metadata such as exif or other available methods.
The computer application user should be able to display saved images by location and time. The images to be viewed should be selected or processed so that they are easier for the operator to compare. This effect can be achieved by selecting the saved images and then processing them.

The user should have a possibility to mark selected areas on the images so that he knows what to pay attention to during the next inventory.

3.2. Solutions

The most important choice in developing a monitoring software is to determine the method of recognizing of monitored objects. It will influence the tools and solutions used.

Image comparison methods using SIFT, SURF, BRIEF, and ORB algorithms are prone to errors related to a rotation of searched objects and their uneven illumination. Used one of these algorithms could involve additional training for the UAV operator to avoid adverse conditions and promote monitoring from a similar perspective for subsequent UAV flights. The software would also require a preparation of images which would allow to identify the monitored objects. Due to the described complications, resulting from the limitations of the analyzed methods, it was decided to use a different approach.

Currently UAVs are often used to create 3D models of physical objects and terrain. This method could also be used to identify and monitor selected objects. However, the available software is a limitation. In the often used available software to create 3D models, it is necessary to fly over the object in a specific way, which can make it difficult to monitor objects in closed spaces or with elements covering the monitored objects. Available methods of comparing point clouds obtained from UAV’s overflights, do not ensure an automatic recognition of objects. The use of this method for object recognition would require a development of new solutions and would have a high computational complexity. Use of point cloud comparisons was discarded due to the large amount of work required to obtain working software and predicted cost effectiveness of this solution.

The use of neural networks to detect objects requires a creation of learning sets, which in the case of the detection of objects would be associated with the collection of many images of monitored objects including those that are not subject to monitoring. The learning process of a neural network could be automated, but it would be difficult to predict the effectiveness of such a network. It would be necessary for such a software to be empirically evaluated or for the learning process to be carried out by a specialist. Such a system would require a considerable amount of work with each new application or some changes in the monitored facilities.

Another possible solution is to use markers that would facilitate a recognition of selected objects. The advantage of such a solution would be the ability of quick adding and removing monitored objects by adding and removing markers. Properly designed markers could also enable a perspective recognition of photographs or videos. If barcodes or other markers are used, only the readability of the marker itself is required, which can be easily assessed by the UAV operator. Due to a high flexibility of the marker-based software, it was decided to use this method in the prototype software.

It was decided to use markers as one of the types of barcodes. It is a proven method of marking objects used in many commercial applications. It was best to use the one that would allow for the most accurate determination of perspective. Some matrix barcodes such as Aztec, QR, Trillcode and Quickmark (Figure 1) have elements designed to facilitate a to determinate of the barcode position. This also allows to determine the perspective and correct it if necessary. Because of the popularity it was decided to use QR codes. This code allows to save the identification number of the monitored object and additional information if necessary.

To read QR codes it was decided to use the ZBar library, which allows to read different types of barcodes. It does not allow to find and correct a perspective. However it is possible to detect and remove the rotation of the code. Due to this limitation decision was made to use OpenCV to detect and correct the perspective. The perspective will also be used to evaluate images from the point of view of their usefulness for monitoring. The position of the code on the image and its scale will also be taken into consideration which will allow to select one image to be saved. Selected images will be placed in a directory named with the identification number of the QR code. The names of image files will contain the time of image registration.
3.3. Prototype

The prototype is designed to accept a video file recorded during a UAV’s overflight as input. During processing input files, it displays a video with additional visualization of QR code detection and data used in the process of saving images. Figure 5 shows one of the moments, when the prototype detected one of markers.

![QR Code Detection Example](image)

Figure 5. Prototype software during processing video file.

The prototype is able to find markers by detecting contours and nesting information about these contours. Basing on three clearly marked corners of QR code which are shown in purple in Figure 5, the fourth corner at the intersection of the orange lines is calculated. The code detected in this way, is ready for further processing.

The detected code is transformed to remove distortions associated with the perspective. A threshold is used for turning a color image into monochrome graphic to make it easier to read the code. A margin is also added to allow the code to be read by ZBar programming library. The image prepared in this way is shown in the left upper corner of Figure 5.

During an operation, the prototype provides real-time information about detected QR codes. It is displayed in the upper left corner and contains such information as:
- identification number,
- parameter describing yaw and pitch of the code,
- distance of the code from the centre of the image,
- scale of the code compared to the whole image.
4. Summary

A review of the methods of object detection in images or video content has been carried out. The methods were analyzed in terms of their use in the design of a computer application that was to be used for monitoring with use of UAVs. A method using QR codes to mark monitored objects, was selected. The choice of this method was influenced by its features such as:

- convenience in marking subsequent objects and their removal,
- fast-paced operation of the algorithm,
- possibility of evaluating the angles of yaw and pitch of the monitored object,
- possibility of using the programming library decoding QR codes,
- possibility of ensuring object recognition by ensuring code visibility.

The prototype software has been developed to enable the recognition of previously marked objects in a video file. The prototype allows to evaluate images with detected QR code in terms of their usefulness for monitoring. It allows to reduce the number of recorded data significantly. In the future, the way, in which the images are assessed, may change especially due to the user's opinions. The software introduces the selected images to directories with the identification number on the QR code. File names on the other hand file names contain the timestamp of image capture. The prototype can be used in the future as a part of monitoring system.

The software should be most useful in monitoring of larger objects in large open spaces. Basing on the software capabilities it could be used in big yards storing machinery that are often found around mines or in long industrial installations like powerlines and pipelines.

It is planned to create a user’s interface that will facilitate and speed up the work with the software. There will also be a part designed to allow for a legible browsing of saved images with a possibility of selecting elements of monitored objects and adding comments. There are plans to add additional information to the QR code which could improve the software. An example may be an inclusion of the object name in the code so there is no need to enter it in the software manually.

References

[1] Stankiewicz K 3/2016 A method for the self-organization of a sensor network in belt conveyor exploitation. Maintenance Problems pp 145-154
[2] Prostański D and Vargová, M 2018 Installation optimization of air-and-water sprinklers at belt conveyor transfer points in the aspect of ventilation air dust reduction efficiency. Acta Montanistica Slovaca, 23 (4).
[3] Jasiulek D, Stankiewicz K and Świder J 2011 An adaptive control system of roadheader with intelligent modelling of mechanical features of mined rock Journal of KONES Vol. 18 No. 2 pp 197-203
[4] Stankiewicz K, Jasiulek D, Rogala-Rojek J and Bartoszek S 2013 Selected, state-of-the art mechatronic systems in polish underground mining industry ISARC 2013 - 30th International Symposium on Automation and Robotics in Construction and Mining, Held in Conjunction with the 23rd World Mining Congress pp 119-129
[5] Moghaddam B, Nastar C and Pentland A 2001 A Bayesian similarity measure for deformable image matching (Image and Vision Computing) vol 19 nr 5 pp 235-244
[6] Shan B A Novel Image Correlation Matching Approach, JMM, vol 5, nr 3, 2010.
[7] Lowe D G 2004 Distinctive Image Features from Scale-Invariant Keypoints (International Journal of Computer Vision) vol 50 nr 2 pp 91-110
[8] Karami E, Shehata M and Smith A 2015 Image Identification Using SIFT Algorithm: Performance Analysis Against Different Image Deformations (Canada: St. John’s).
[9] Güzel M 2015 A Hybrid Feature Extractor using Fast Hessian Detector and SIFT (Technologies) vol 3 nr 2 pp 103-110
[10] Chiu L, Chang T, Chen J and Chang N 2013 Fast SIFT Design for Real-Time Visual Feature Extraction (IEEE Trans. on Image Process) vol 22 nr 8 pp 3158-3167
[11] Ke Y and Sukthankar R 2004 PCA-SIFT: A more distinctive representation for local image descriptors (CVPR) vol 2 pp 506–513
[12] Bay H, Tuytelaars T and Van Gool L 2008 Speeded-up robust features (SURF) (Computer vision and image understanding) vol 110 nr 3 pp 346-359
[13] Rublee E, Rabaud V, Konolige K and Bradski G 2011 ORB: and efficient alternative to SIFT
or SURF (IEEE International Conference on Computer Vision)

[14] Calonder M, Lepetit V, Strecha C and Fua P 2010 BRIEF: Binary robust independent elementary features (European Conference on Computer Vision)

[15] Dobrosielski W 2010 Zastosowanie sztucznych sieci neuronowych do rozpoznawania obrazów (Studia i Materiały Informatyki Stosowanej), vol 2 nr 3 pp 45-56

[16] Tadeusiewicz R 1993 Sieci neuronowe (Warszawa: Akademicka Oficyna Wydawnicza RM) pp 13-14