

# A FAST RECOGNITION ALGORITHM FOR DETECTION OF FOREIGN 3D OBJECTS ON A RUNWAY

V. V. Kniaz<sup>a\*</sup>

<sup>a</sup> State Res. Institute of Aviation Systems (GosNIIAS), 125319, 7, Victorenko str., Moscow, Russia - vl.kniaz@gosniias.com

Commission III, WG III/3

**KEY WORDS:** 3D object recognition, foreign object detection systems, runway detection system

## ABSTRACT:

The systems for detection of foreign objects on a runway during the landing of an aircraft are highly demanded. Such systems could be installed in the airport or could be mounted on the board of an aircraft. This work is focused on a fast foreign object recognition algorithm for an onboard foreign object detection system.

The algorithm is based on 3D object minimal boundary extraction. The boundary is estimated through an iterative process of minimization of a difference between a pair of orthophotos. During the landing an onboard camera produces a sequence of images from which a number of stereo pair could be extracted. For each frame the runway lines are automatically detected and the external orientation of the camera relative to the runway is estimated. Using external orientation parameters the runway region is projected on an orthophoto to the runway plane. The difference of orthophotos shows the objects that doesn't coincide with the runway plane. After that the position of the foreign object relative to the runway plane and its minimal 3D boundary could be calculated. The minimal 3D boundary for each object is estimated by projection of a runway region on a modified model of the runway. The extracted boundary is used for an automatic recognition of a foreign object from the predefined bank of 3D models.

## 1. INTRODUCTION

### 1.1. Foreign object debris detection problem

The state and the quality of a runway plays the critical role for the safety of the aircraft during the landing stage. Any foreign object on a runway poses a significant threat to the safety of air travel. Such an object usually called foreign object debris (FOD) has the potential to damage aircraft during critical phases of flight, which can lead to catastrophic loss of life and airframe, and increased maintenance and operating costs (FAA, 2009). It is estimated that damage caused by FOD costs the international aviation industry US\$13 billion per year.

Examples of FOD include:

- airport operation vehicles
- fueling tracks
- aircraft parts
- rocks, broken pavement
- garbage, maintenance tools

The foreign object detection systems are highly demanded by airports. Such a system consist of sensors that check the surface of a runway for any changes or unexpected objects. In the case of detection the system alerts the airport service.

### 1.2. Existing FOD detection systems

Even prior to the development of any automatic FOD detection systems runways were constantly inspected by airport personnel. However the usage of automatic electronic FOD detection systems greatly increases FOD detection rate and reduces the potential hazard.

FOD detection systems could be divided into two categories depending on the type of sensor they use. Nowadays systems with radar and electro-optical sensors are available. Such systems could be either stationary located on the ground or mounted on top of a moving inspecting vehicle. Such systems

usually provide a great accuracy and capable to detect FOD as little as 3.0 cm high. However the usage of statically mounted FOD detection system could only provide the safety during the landing on runways equipped with such systems.

This paper presents the algorithm for detection of FOD using the mobile system mounted on top of an aircraft. Such system is capable to detect moderate size FOD on runways unequipped with stationary FOD detection systems. The algorithm uses the data from an electro-optical sensor to automatically detect the foreign object on a runway and estimate its position and size. If the object exists in the predefined database of possible FOD it is automatically recognized.

## 2. SYSTEM CONFIGURATION

### 2.1. Sensors and recognition process workflow

To alert the pilot in the case of FOD on a runway it is proposed to mount an electro-optical sensor on top of an aircraft. An infrared sensor could be used to increase the performance of the system in the poor weather conditions.

It is possible to use sensors incorporated in commercially available enhanced vision systems (EVS) to obtain the input data from for the algorithm. An example of EVS system with an infrared data sensor (1-5 Micron) manufactured by Kollsman presented on figure 1.

The data from the sensor is transferred to the main processing unit. The recognition of FOD is done in the following steps:

1. Restoration of the original data.
2. Detection of the runway.
3. Determination of camera external orientation.
4. FOD region estimation.
5. Reconstruction of FOD 3D shape.
6. Search for a similar object in the FOD database.



Figure 1. Infrared data sensor in EVS system. Credit: Kollsman

### 3. RUNWAY RECOGNITION

The essential step prior to the estimation of the external camera orientation is determination of reference points and lines on a runway. It is possible to use any visible feature points with known geodetic coordinates as reference points. The most persistent and noticeable features visible during the approach and the landing stages are runway edge markings during the day and runway edge lightning during the night. An example of typical runway edge marking is shown on figure 2.



Figure 2. An example of typical runway markings in Domodedovo airport (DME). Credit: Google Earth.

Four reference points were chosen to develop a fast algorithm for runway recognition. They coincident with the beginning and the end of left and right edge markings of the runway.

#### 3.1. Runway edge detection

To detect edge markings of the runway a convolution with a vertical edge detection kernel is used. Usually the edge between runway marking and the inner part of the runway is more prominent than the edge between runway marking and the outer part of runway. To effectively detect runway edge marking in such cases left and right runway edge markings are detected separately by two different kernels given by:

$$\begin{aligned} G_L &= [0 \ 1 \ -1] \\ G_R &= [-1 \ 1 \ 0] \end{aligned} \quad (1)$$

Result of the convolution is shown on figure 3.



Figure 3. Image of the runway after convolution with the left border marking detection kernel

After the convolution with both kernels two resulting images are binarization using Otsu's method. To detect runway edge markings, threshold marking and touchdown zone marking an image segmentation was used.

An original fast image segmentation algorithm was developed to perform an image segmentation in real time. The algorithm performs the segmentation through the single scan of the image data. To detect image regions that has complex connections a label image is created during the scan. Each pixel of the label image holds the unique index of a separated area in the original image. Such index is assigned to each non-zero pixel of an original image that yet has no associated index in the label image. After that four neighbor pixels are examined (three in the vertical positive direction of the scan and one in the horizontal positive direction of the scan). The same index is assigned to each neighbor pixel if it is non-zero and doesn't have any index in the label image yet. If the neighbor pixel already has an index in the label image the back substitution of indexes in the label image is done: all pixels with the current index of the current pixel are reindexed to the index of the neighbor pixel. The label image produced after the segmentation is shown on the figure 4.

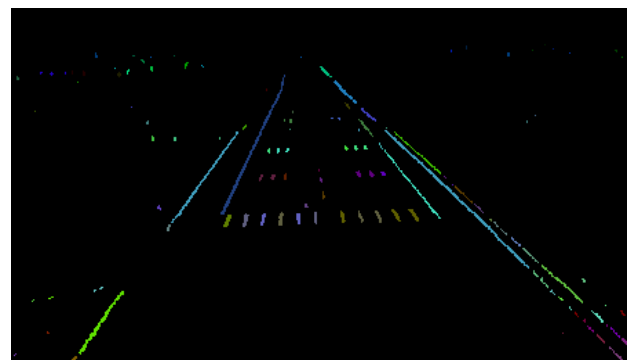


Figure 4. Label image that shows the separated pixel region image of binary image. The runway left edge marking is visible as a long inclined blue line left to the image center.

The automatic detection of runway edge markings is done under the following assumptions:

- the length of a runway edge marking region is at least 1/10 of the image height
- the borders of the region are parallel to its axis of symmetry (i.e. it is nearly a straight line)
- the left runway edge marking has a positive gradient (i.e. is inclined towards the right border of the image) and the right edge marking has a negative gradient

Given the following assumptions each region that was found in the original image is examined. Firstly all regions that have less pixels than 1/10 of the image height are excluded from the search. After that the straight line approximation of all remaining regions is performed. The approximation is done by a straight line fitting proposed by (Press, 2007).

After the line fitting the borders of each region are checked to be parallel with the approximated axis of symmetry. Orthogonal distances to edges of the region are calculated (border profiles). The standard deviation of distances to edges gives the nonlinearity criteria. Regions with low standard deviations are likely to be lines.

Finally runway edge marking regions are selected using the nonlinearity criteria and approximated gradient of the line. The region with low nonlinearity and positive gradient on the left image is selected as the left runway edge marking. The right runway edge marking is found in the similar way.

### 3.2. Runway threshold and touchdown zone detection

To detect runway threshold and touchdown zone markings the region between estimated border markings is searched. The line passing through the beginning of left and right border markings gives the position of the first scan lines. The label image (either left or right) is scanned along the lines that are parallel to the first scan line. If 12 (sometimes 16) regions with different indexes are found along the scan line they are chosen as runway threshold marking regions. If 6 regions with different indexes are found along the scan line they are chosen as runway touchdown marking regions. Figure 5 illustrates the process of runway threshold and touchdown zone markings detection.

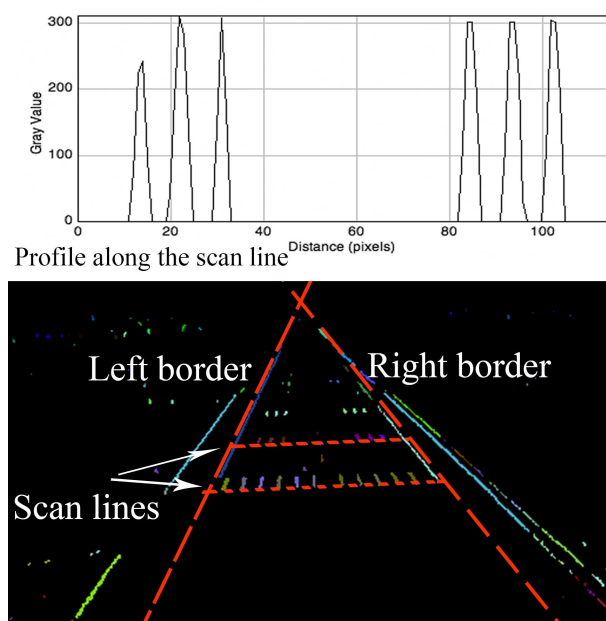


Figure 5. Process of runway threshold and touchdown zone markings detection

## 4. CAMERA EXTERNAL ORIENTATION

The process of camera external orientation is done in two steps: firstly the rotation matrix is found using the vanishing points given by runway edge markings and the Gaussian sphere method proposed by (Barnard, 1983). Secondly the position of the center of projection is found using the runway reference points found on the image.

### 4.1. Runway, camera and image coordinate systems

The runway coordinate system is defined as follows: the origin is located at the beginning of the right runway edge marking, X-axis is coincide with the right runway edge marking, Y-axis is coincide with the line passing through the beginnings of edge markings, Z-axis is normal to X- and Y- axes. The image coordinate system is defined as follows: the origin is located at the bottom left pixel, X-axis is directed to the right, Y-axis is directed upwards. The camera coordinate systems is defined as follows: the origin is located at the projection center, X-axis is parallel to the image X-axis and has the same direction, Y-axis is parallel to the image Y-axis but has the opposite direction, Z-axis is normal to X- and Y- axes. Figure 6 illustrates runway, camera and image coordinate systems.

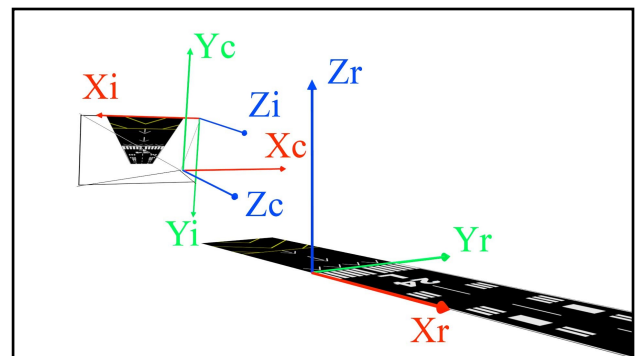


Figure 6. Runway, camera and image coordinate systems

### 4.2. Determination of rotation matrix

The rotation matrix that defines the transformation from the camera coordinate system to the runway coordinate system is found using vanishing points. The vanishing point that is given by the intersection of runway border markings provides the direction of runway coordinate system X-axis in the camera coordinate system. To find the direction of runway coordinate system Y-axis runway threshold and touchdown zone markings are used. Two lines parallel to Y-axis of runway coordinate system could be defined as the line congaing front points of runway threshold and touchdown zone markings. The intersection of this lines gives the second vanishing point. The second vanishing point gives the direction of the runway coordinate system Y-axis in the camera coordinate system. The direction of Z-axis could be found as a cross product of vectors defining X- and Y- axes.

The reference points on the runway are named as follows:

- $b_L, b_R$  – beginning points of left and right runway edge markings
- $e_L, e_R$  – end points of left and right runway edge markings
- $tr(i)$  – 12 front points of threshold marking
- $td(i)$  – 6 front points of the first touchdown zone marking

Coordinate points in the camera coordinate systems are named with lowercase letters, e.g.  $b_L$ . Coordinate points in the runway coordinate system are named with capital letters, e.g.  $B_L$ . Coordinate points in the image coordinate system are named with lowercase letters with 'i' subscript, e.g.  $b_{iL}$ . The transformation from the image coordinate system to camera coordinate systems is given by:

$$x = x_i b + p_0 \quad (2)$$

where  $x$  - point in the camera coordinate system,  $x_i$  - point in the image coordinate system,  $b$  - pixel scale,  $p_0$  – principal point position in the camera coordinate system.

Using this notation normals to interpretation planes containing left and right border markings in the image plane in the camera coordinate systems are given by:

$$n_L = e_L \times b_L; n_R = e_R \times b_R \quad (3)$$

The unit vector  $v_X$  that gives the direction of X-axis of the runway coordinate system in the camera coordinate system is given by:

$$v_X = \frac{n_R \times n_L}{|n_R \times n_L|} \quad (4)$$

The second vanishing point is found in the similar way using points  $tr(1)$ ,  $tr(12)$ ,  $td(1)$ ,  $td(6)$  located on runway threshold and and touchdown zone markings lines. This vanishing point provides the unit vector  $v_Y$  that gives the direction of Y-axis of the runway coordinate system in the camera coordinate system. The unit vector  $v_Z$  that gives the direction of X-axis of the runway coordinate system in the camera coordinate system is given as cross product of  $v_X$  and  $v_Y$ .

After that the rotation matrix that defines the transformation from the camera coordinate system to the runway coordinate system is given by:

$$R = \begin{bmatrix} v_X^T \\ v_Y^T \\ v_Z^T \end{bmatrix} \quad (5)$$

#### 4.3. Determination of the projection center

The vector  $b_R$  coincide with the center of origin of the runway coordinate system. The rotation of this vector by  $R$  matrix gives the direction vector to the center of origin of the camera coordinate system in the runway coordinate system. To find the length of this vector the scale is found using the runway reference points. Two reference points  $B_L$  and  $B_R$  that represent the width of the runway are used:

$$XC_0 = R \cdot b_r \cdot \frac{|R^{-1}(B_L - B_R)|}{|b_L - b_R|} \quad (6)$$

## 5. FOREIGN OBJECT DETECTION

To detect foreign objects on the runway the orthophoto images difference method proposed by (Zheltoy, 2000) is used. As the images in the sequence captured by an aircraft during the approach stage have different external orientation it is possible to use a pair of images to detect an object that doesn't lie in the runway plane (figure 7). To detect such an object an orthophoto to the runway plane is generated for each image and the difference of orthophoto images is calculated. Any non-zero pixel on the difference image indicates a potential foreign object located on the runway.

### 5.1. Orthophoto generation

To generate the orthophoto to the runway plane a grid of the points in the runway plane is defined. Each point corresponds to the pixel of the resulting orthophoto. For each point an intersection with the image plane is found. The intersection point is found as the intersection of a line and a plane. The line is given by:

$$p = d \cdot i + i_0 \quad (7)$$

where  $p$  – is the point of intersection,  $d$  – is the distance along the direction vector  $i$  from  $i_0$ ,  $i$  – direction vector of the line given by the vector of the current point on the runway in the camera coordinate system,  $i_0$  – is a point on the line (projection center). The plane is given by:

$$(p - p_0) \cdot n = 0 \quad (8)$$

where  $p$  – is the point of intersection,  $p_0$  – is a point on a plane (principal point position),  $n$  – is normal to the plane. Solving (7) and (8) for  $d$  gives the distance from  $i_0$  to the point of intersection  $p$ :

$$d = \frac{(p_0 - i_0) \cdot n}{i \cdot n} \quad (9)$$

Substituting  $d$  back to (7) gives the point of intersection in the camera coordinates system. Using the equation (2) the pixel coordinates in the image coordinate system are found. The value of that pixel gives the value of the current point on the orthophoto. An example of resulting orthophoto is shown on figure 8, a.

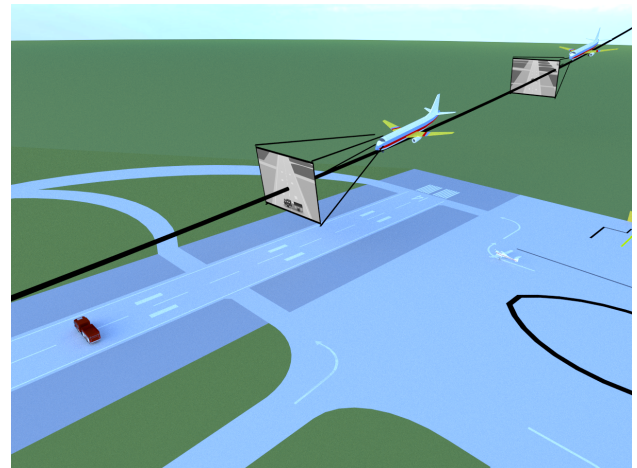


Figure 7. A pair of images used for the detection of foreign objects obtained by onboard camera during the landing



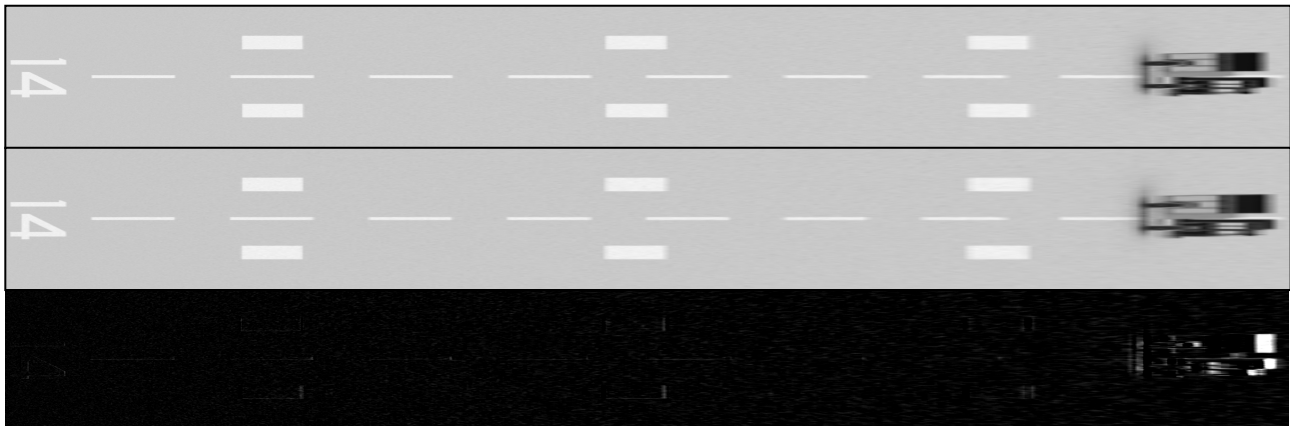


Figure 8. (a, upper) An example of an orthophoto with a foreign object (a firetruck) visible on the right side. (b, middle) Second orthophoto generated from the image captured from the different position of the camera. (c, down) Difference of orthophotos.

## 5.2. Foreign object detection

To detect a foreign object a pair of orthophotos is generated. The non-zero pixels of the difference of orthophotos indicate the location of an object that doesn't lie in the runway plane (i.e. an object with non-zero height). An example of the difference of orthophotos is shown on figure 8, c. The foreign object is clearly visible as the bright spot on the left side of the difference image.

## 6. FOREIGN OBJECT RECOGNITION

To automatically find the detected object in the predefined database of 3D objects a minimal bounding box is estimated using the orthophoto. After that an object with the most similar dimensions of the minimal bounding box is searched in the database. The 3D object with a closest minimal box dimensions found in the database gives the type of the detected object.

### 6.1. Foreign object position and minimal bounding box estimation

To estimate the position and the minimal bounding box of the foreign object binarization and image segmentation of the orthophoto is done. The bright region in the difference of the orthophotos gives the seed point for image segmentation in the orthophoto. An example of binarization of the orthophoto is shown on figure 9. The difference of the orthophotos image is then added to the binary image to exclude the bright region from the foreign object region. After the segmentation an approximation of the contour of the foreign object region with a rectangle is done. The contour of the bright region visible on the difference image is also approximated with a rectangle.

After that the parameters of the minimal bounding box are found. The height of the foreign object rectangle gives the actual width of the minimal bounding box in the runway coordinate space. The width of the bright region rectangle gives the depth of the minimal bounding box. The width of the foreign object rectangle actually represents the diagonal of the minimal bounding box in the height/depth plane. So the actual height of the foreign object is given by:

$$h = \sqrt{l^2 - d^2} \quad (10)$$

where  $h$  – is the height,  $l$  – diagonal (width of foreign object rectangle),  $d$  – depth (width of bright region rectangle)

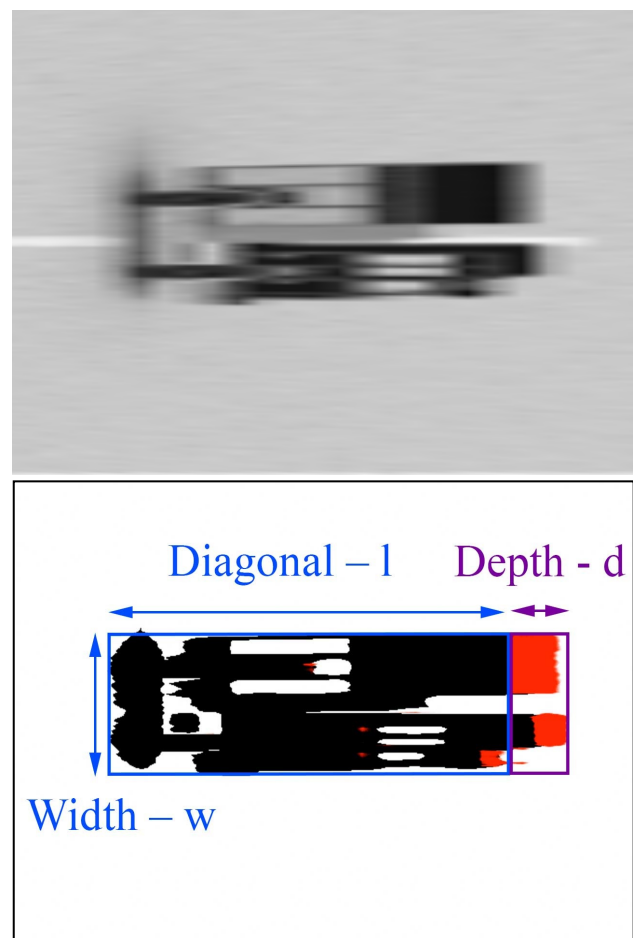


Figure 9. (a, upper) Fragment of the original orthophoto centered on the foreign object. (b, lower) Binarization of the orthophoto. Bright region visible on the difference of orthophotos is shown with red.

The position of the foreign object in the runway coordinate system is also found using the orthophoto. The bottom left point of the foreign object rectangle defines the origin point of the foreign object coordinate system. The rotation of the main axes of the foreign object rectangle gives the rotation of the foreign object coordinate system around the Z-axis.

## 6.2. Minimal bounding box dimensions refinement

To refine dimensions of the bounding box an iterative minimization of the difference of orthophotos is done. After the primary estimation of minimal bounding box position and dimensions it is appended to the runway 3D model. Runway points located under the minimal bounding box are lifted to the height of the minimal bounding box. Using the new runway 3D model a new pair of orthophoto images is generated. To get the new pixel values of the orthophoto each ray connecting the runway point and the image plane is checked for intersections with planes of the foreign object minimal bounding box.

If the ray intersects a plane of the minimal bounding box the current runway point is invisible from the camera point of view. Such point is labeled with zero value on the resulting orthophoto (black color). A new pair of orthophotos generated using the updated 3D model of runway is shown on figure 10.

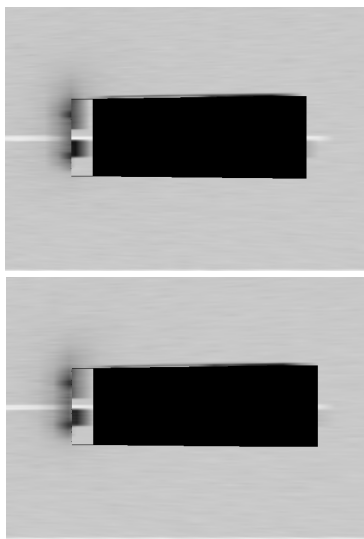


Figure 10. New pair of the orthophotos generated using the updated 3D model of the runway. The black areas represent the regions of the runway that are invisible from the camera's point of view (i.e. occluded by the foreign object). Note that the pixels representing the roof of the firetruck are now located on the place where the firetruck is positioned.

After the generation of the new pair of orthophotos their difference is found. If the primary estimation of the parameters was good enough values of the difference image should be close to zero (i.e. black image should be produced). If there were errors in the estimated dimensions of the minimal bounding box the iterative minimization of the difference of orthophotos is done. Small increments are given to the width, the height and the depth of the minimal bounding box and a new pair of orthophotos and their difference image are generated. The integral value of the new difference image is compared with the integral value of the previous image. If the new integral value is smaller than the previous one a new increment to the dimensions is done. The process continues until the desired minimal integral value is reached (figure 11).

## 7. CONCLUSIONS

A combination of a basic EVS system and the algorithm that uses image processing and photogrammetric methods provides a way to detect moderate size foreign objects located on a runway. The usage of such algorithm could increase the safety

of the landing, especially in airports that are not equipped with stationary FOD systems.

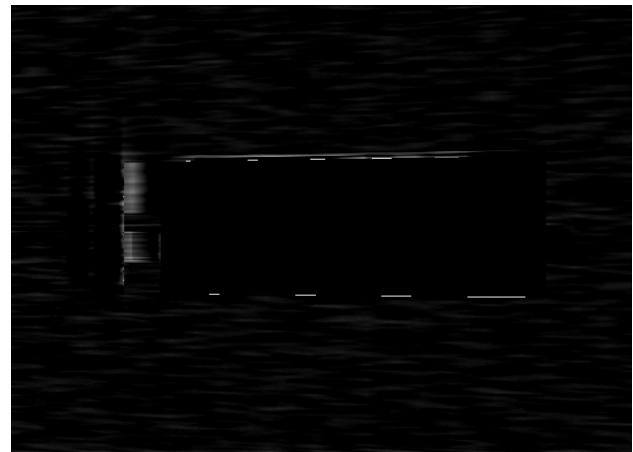


Figure 11. The resulting difference of orthophotos images after the process of iterative minimization.

An algorithm for fast detection and recognition of foreign objects on a runway was developed. The algorithm automatically detects runway markings and estimates the camera external orientation in the runway coordinate system. The algorithm doesn't require navigation information of an aircraft for estimation of the external orientation. The only additional information needed is the geodetic coordinates of reference points on the runway. The algorithm detects and estimates the dimension of the foreign object using the difference of the orthophotos.

The test input data for the algorithm was created using a 3D modeling software. A 3D model of the airport scene was created and images from a virtual camera were rendered. Errors in the estimation of the external parameters were no more than 5 meters in the runway space.

To reduce the errors it is suggested to use all available runway reference points during the process of projection center position estimation. The major step to improve the quality of object recognition can be done by presenting a 3D model of a foreign object in voxel form with variable voxel size. The recognition of such features could be done by a recursive application of the developed algorithm.

## REFERENCES

- Barnard, S. T., 1983. Interpreting perspective images. *Artificial Intelligence* 21, pp. 435-462.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., 2007. *Numerical Recipes: The Art of Scientific Computing, Third Edition*, 1235 pp. (New York: Cambridge University Press).
- U.S. Federal Aviation Administration, 2009. *Advisory circular on Airport Foreign Object Debris (FOD) Detection Equipment* (30 Sep. 2009).
- Zheltev S., Sibryakov A., 2000. Method of 3d-object detection based on orthophoto difference analysis. *The International Archives of Photogrammetry and Remote Sensing*. Vol. XXXIII, Part B3. Amsterdam.