Development of a Machine Vision System for Image Recognition of Design Estimates

Article *in* Nonlinear Phenomena in Complex Systems · June 2022

DOI: 10.33581/1561-4085-2022-25-2-159-167

2 authors:

Oleg Demidenko
Francisk Skorina Gomel State University

Natalja Aksionova
Francisk Skorina Gomel State University

Nonlinear Phenomena in Complex Systems, vol. **25**, no. 2 (2022), pp. 159 - 167

Development of a Machine Vision System for Image Recognition of Design Estimates

O. M. Demidenko* and N. A. Aksionova†
Gomel State University, Gomel, BELARUS
(Received 14 January, 2022)

This paper discusses implementation of a machine vision system for pattern recognition of design estimates. The main problem in the development of these systems is the choice of unique features that remain invariant to various kinds of transformations. Angular descriptors were chosen as a dominant feature. The paper presents a comparative analysis of the corner detection methods of the Moravec algorithm, the Harris algorithm, and the Shi–Tomasi algorithm. The authors have developed software in Python language that implements the operation of the Harris detector and the Shih-Tomasi detector. The recognition system is being tested for building 3D models in Blender.

AMS Subject Classification: 68T10, 68T45, 03E15
Keywords: computer vision, machine vision, pattern recognition, singular points, detector, descriptor, OpenCV, Python
DOI: https://doi.org/10.33581/1561-4085-2022-25-2-159-167

1. Introduction

Recognition is the identification of the transformation of input information, which is appropriately defined by some features of recognizable objects in the output information, representing the conclusion about which class the recognizable object is found.

A *key point* of m-th image is the point on an image, the neighborhood of which $\Omega(m)$ is distinguishable from the neighborhood of any other point on the image $\Omega(n)$ in some other neighborhood of a special point $\Omega_2(m)$. A key point is a point in the image that more likely to be in another image of the same object. As the neighborhood of an image point for most algorithms one uses a window of 5×5 pixels, more rarely 3×3 or 9×9 pixels.

A *detector* is a method for extracting a set of special points from an image. The detector provides invariance of finding the same singular points with respect to image pre-transformations.

A *descriptor* is an identifier of a singular point that distinguishes it from the rest of the set of singular points. In its turn, descriptors must provide invariance of finding the correspondence between singular points with respect to image transformations [7].

Haralick and Shapiro [3] singled out the following as the main criteria for identifying singular points:

1. Distinguishability: The point must be unique in its neighborhood and stands out from the background;
2. Invariance: The set of singular points does not depend on affine transformations;
3. Stability: the set of singular points should be stable to noise and errors;
4. Uniqueness: A singular point must have global uniqueness to improve the distinguishability of repeated patterns;
5. Interpretability: definition of a set of singular points in such a way that they can be used for subsequent correspondence analysis and for improving image inter-preservation.

Corners are special points formed by two or more faces, which usually define the boundary between different objects or parts of the same object. In other words, an angle is a point in the

*E-mail: demidenko@gsu.by
†E-mail: govorushkina@gsu.by

vicinity of which the intensity of pixels changes in all directions relative to the center (x, y).

Most of the angle detection methods use the second order finite differences and are sensitive to noise.

2. Statement of the problem

To recognize images of project documentation, this paper proposes to develop a machine vision system consisting of a camera and software. The camera creates a digital equivalent of the architectural plan. The resulting image is placed on a computer and recognized using the developed software modules. The key points found in this way are used to build 3D model in the 3D graphics editor.

The input information of the task of design documentation image recognition is a full-color raster image. The search for a basic set of singular points is based on the idea of using only those pixels that define angles [1]. The input image is replaced by a set of such singular points and this set will be the basis for the subsequent 3D modeling process. The definition of special points for the design room plan will accelerate the modeling process, allow to manage 3D model parameters more effectively and to make adjustments without changing the input image.

Basic requirements for the project documentation recognition system:

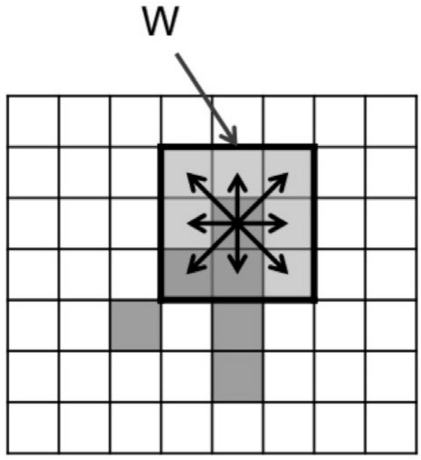
1 *Quantity* of detected singular points should be large enough. The optimal number of feature points depends on the subject area. The specificity of images of architectural plans allows you to select sets of certain pairs of singular points for constructing spaces.

2 *Accuracy* – the detected singular points must be accurately localized, both in the original image and taken on a different scale.

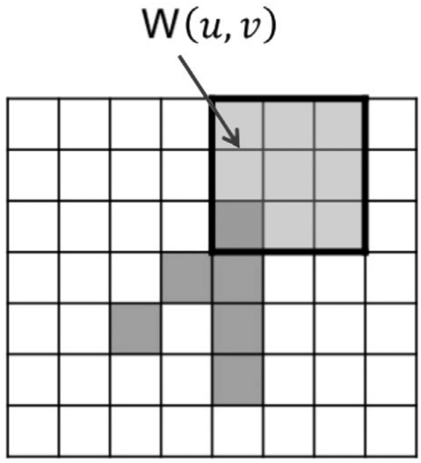
3 *Efficiency* – the time of detection of feature points in the image must be acceptable in time-critical applications.

3. Mathematical foundations

Moravec detector. The Moravec [6] method for analyzing a digital image proposes to measure the change in pixel (x, y) brightness by shifting a square window W centered at (x, y) (Figure1) by one pixel in each of eight directions (up, down, right, left, and four directions diagonally) (u, v):



a)



b)

FIG. 1. a) window shifting directions W. b) example of window shifting W at $(u, v) = (1, -1)$. Displacement of the 1×1 pixel window W centered at (x, y) in the Moravec algorithm.

Moravec detector algorithm.

1. For each direction (u, v) find the difference of pixel brightness change (x, y):

$$V_{u,\nu}(x,y) = \sum_{\forall a,b} (I(x+u+a, y+\nu+b) - I(x+a, y+b))^2,$$

where $I(x, y)$ is the brightness of the pixel (x, y) in the original image.

2. Build a probability map of angles by calculating the value $C(x, y)$ for each pixel (x, y):

$$C(x,y) = \min_{(u,v)}\{V_{u,v}(x,y)\}.$$

3. The threshold value T of the probability map is determined. Pixels with $C(x, y)$ values lower than the threshold value T are cut off, and $C(x, y)$ values are zeroed.

$$C(x,y) = \begin{cases} C(x,y), & \text{if } C(x,y) > T \\ 0, & \text{else.} \end{cases}$$

4. Find local maxima in $C(x, y)$ to detect angles. All obtained non-zero elements of map C correspond to angles in the image. The recurring angles are removed by applying the procedure to find local maxima of the response function.

The result of the algorithm is a probability map C in which the special points (corners) are nonzero values.

The Moravec method has the following disadvantages:

1. It is not invariant with respect to the "rotation" transformation;

2. A large number of errors due to the presence of noise in the image: the algorithm falsely determines angles on edges due to noise;

3. Anisotropy in only eight directions.

Harris detector. The Harris [4] method is based on the Moravec detector and is an improved algorithm, since it considers the W window shifts in all directions.

For the input image I, the window W (usually its size is chosen to be 5×5 pixels, but can be different depending on the image size) with the center at (x, y), and the shift of this window at *(u,v)* is considered [4].

Harris detector algorithm.

1. We find the difference in the intensity of the shift (u, v) in all directions. The weighted sum of the square of the differences between the window W and the window W shifted by *(u,v)* (change in the neighborhood of the point (x, y) when shifted by (u, v)) is equal:

$$E(u,\nu) = \sum_{(x,y)\in W} w(x,y)\,(I(x+u, y+\nu) -$$

$$-I(x,y))^2 \approx \sum_{(x,y)\in W} w(x,y)\,(I_x(x,y)\,u -$$

$$-I_y(x,y)\,v)^2,$$

where $W(x, y)$ is the weight function (usually a Gaussian function or a binary window), I_x and I_y are derivatives in the x and y directions respectively.

2. To find the angles it is necessary to maximize the function *E(u,v)*. To find the derivatives, we apply Taylor's theorem to the above equation and, using some mathematical steps, we obtain:

$$E(u,v) \approx [\, u \;\; v \,] M \begin{bmatrix} u \\ v \end{bmatrix}.$$

In the above formula, M is the autocorrelation matrix:

$$M = \sum_{(u,v)\in W} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix},$$

where I_x and I_y are derivatives in the x and y directions, respectively.

3. The response measure is defined

$$R_{Harris} = \det(M) - k(TrM)^2 > k,$$

where k is an empirical constant. The parameter k is the Harris parameter, its value is usually in the range 0.04–0.15 and is determined empirically.

The values of $R_{Har} >= 0$ define the angular singular points. Further, the points are filtered by R_{Har}, those points with R_{Har} values less than a certain threshold are excluded from consideration.

The angle is characterized by large changes in the function $E(x, y)$ along the set of all directions (x, y), which is equivalent to large modulo eigenvalues of the matrix M. The location of the eigenvalues is shown in Figure 2, where λ_1 and λ_2 are the eigenvalues of the matrix M.

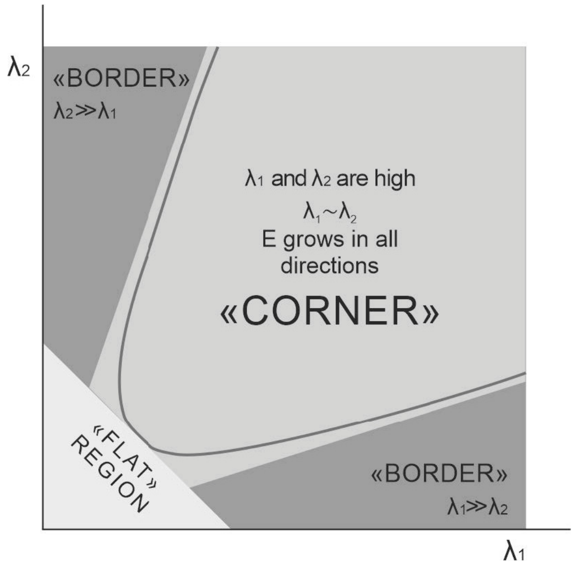


FIG. 2. Changing values of the function $E(x, y)$ when recognizing an angle.

Figure 3 shows that only when λ_1 and λ_2 are above the minimum value λ_{min} is considered an angle.

Shi–Tomasi Detector. The Shi–Tomasi angle detector (Shi–Tomasi or Kanade–Tomasi, 1994) is based on the Harris algorithm, but uses a different response measure R:

$$R_{ShiTomasi} = \min(\lambda_1, \lambda_2).$$

1) when $|R_{ShiTomasi}|$ is small, which occurs, when λ_1 and λ_2 are small, the region is a plane;

2) when $R_{ShiTomasi} < 0$ $R < 0$ at $\lambda_1 >> \lambda_2$ or vice versa, the region is a border;

3) when $R_{ShiTomasi}$ is high at high λ_1 and λ_2 and $\lambda_1 \sim \lambda_2$, the region is a corner [5].

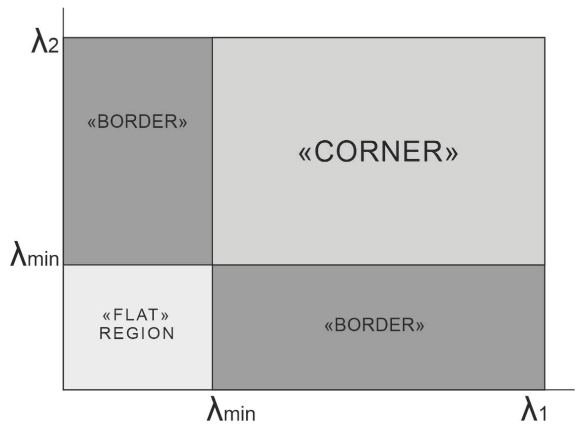


FIG. 3. Diagram for determining the values of λ_1 and λ_2 in angle recognition.

It is assumed that with this choice of response measure, the angle search will be more stable.

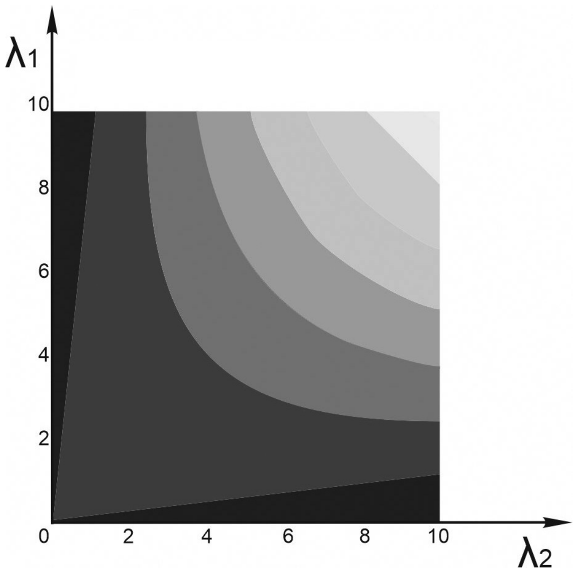
In its simplest form, the Harris detector detects the difference between λ_1 and λ_2 by subtracting the square of the sum of these values from the product of these values. In the graph shown in Figure 4a, the darkest areas are negative values, which represent edges; the lighter colors are values close to zero, which represent flat areas; the brighter areas are large numbers, which represent corners. This algorithm gives a highly detailed search area, and the mathematical formula allows you to distinguish between corners, edges, and flat areas.

However, two disadvantages are evident in this detailed representation:

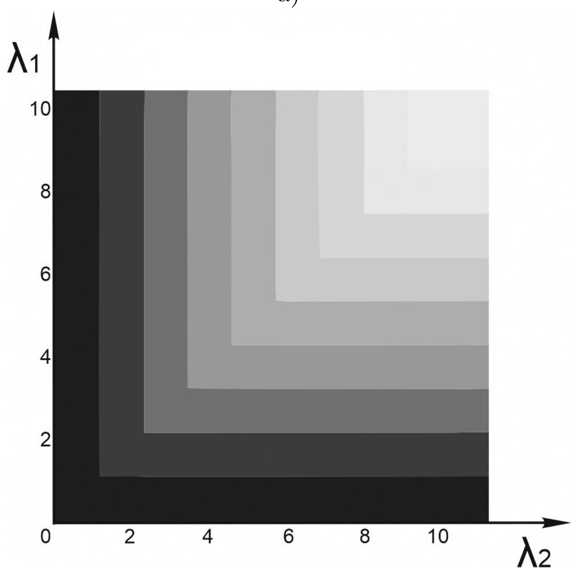
— a free parameter k is required, which must be chosen manually;

— multiplication and gain functions are good for mathematical analysis, but ineffective for computer calculations.

The Shi–Tomasi algorithm offers another estimation function that is not as accurate, but is a good approximation and computes much faster on computer pro-cessors. The speed is important because this function has to be calculated for every pic-section of the image. The function itself is quite simple: it takes the smaller of the two values λ_1 and λ_2. The result of the calculations



a)



b)

FIG. 4. a) graph for the Harris detector angle detection function $R_{Harris} = \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)$, $k = 0.09$ b) graph for the Harris detector angle detection function $R_{ShiTomasi} = \min(\lambda_1, \lambda_2)$.

is shown in Figure 5b.

Significant disadvantages of the Harris detector:

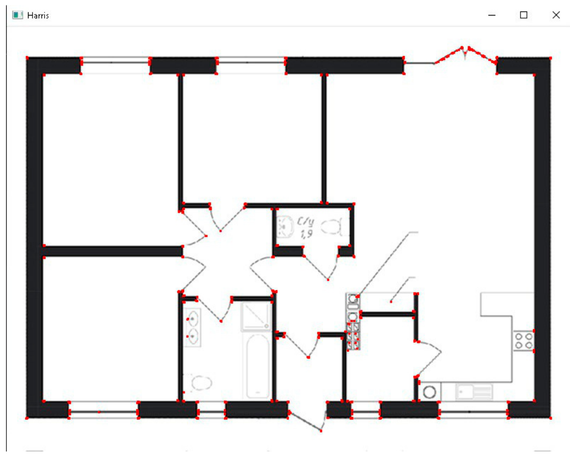
— greater computational complexity (compared to the Moravec detector);

— sensitivity to noise;

— the dependence of the detection results on the scale of the image (to eliminate this drawback, a multiscale Harris detector is used).

However, the Harris detector is invariant to the rotation transformation and partially invariant to affine intensity changes.

A practical comparison of angle detectors. The authors of this work has developed software in Python language, which implements the search of angles of special points for images of room plans of design documentation. For comparative analysis of Harris detector and Shi–Tomazi detector the images of project room plans with resolution of 600×800 pixels in png format were considered.



a)

b)

FIG. 5. (color online) Recognizing of Harris and Shi-Tomasi detectors estimated number of angles: the Harris (a) and Shi–Tomazi (b) detector test images.

Figure 5a shows the test images with marked special points found by the Harris detector. In the image 5a, 65 special points were found in 0.103 seconds. The Harris detector actually did not find any special points on the circumference and edges of the objects. However, it can be noted that in the vicinity of some corners more than one singular point was found, which can be considered a disadvantage of this algorithm. Also, the detector ignored the corners whose brightness is not very brightness (rectangles of different brightness on the left side of the image and the corners inside of some shapes). Due to the property of anisotropy in all possible directions, the Harris detector detected angles of different sizes (not just 45 and 90 degrees).

The results of the Shi–Tomasi detector (Figures 5b) are very similar to those of the Harris detector. The Shi–Tomasi algorithm found and missed the same angles, but the advantage is absence of duplicate points in the vicinity of the corners. The Shi–Tomasi method found 57 special points in 1.980 seconds.

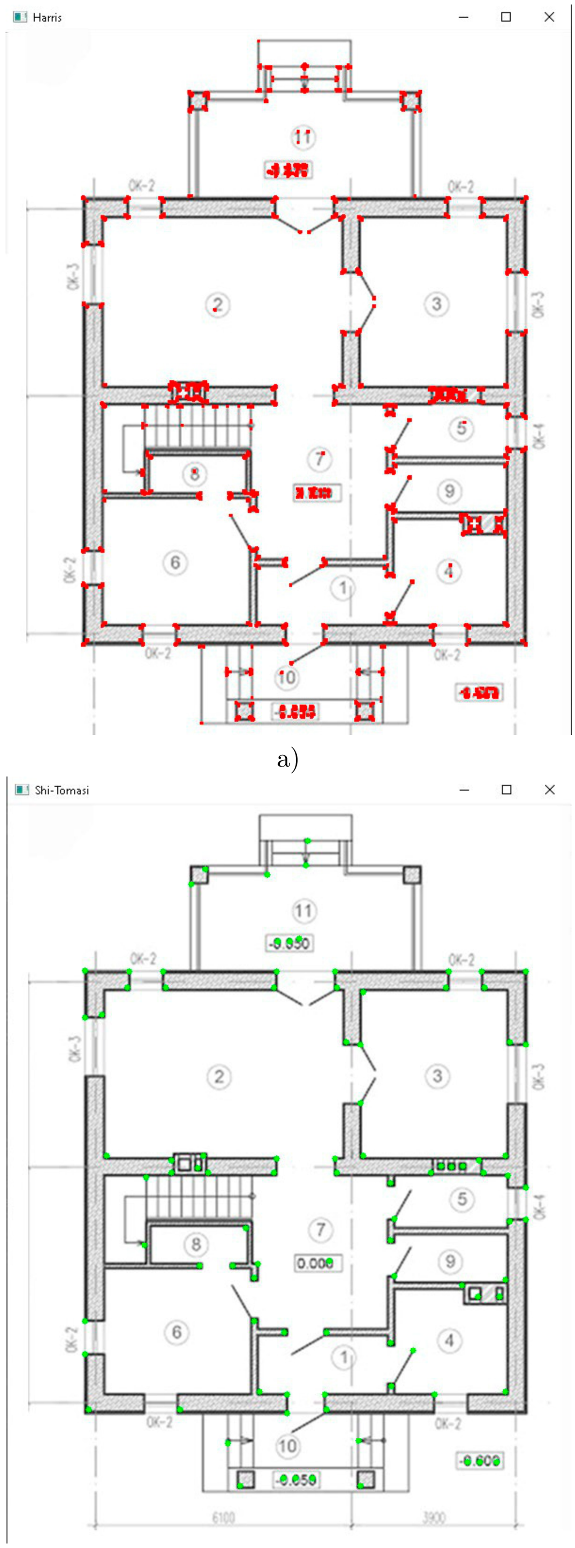
In this work we also investigated the performance of angle detectors for non-contrast images with noise. Figure 6 shows the results of the detectors for a 600 × 800 pixel image.

On the noisy image, the Harris detector made a small several special points were found where they were detected by the Harris detector. The algorithms did not show perfect, though good enough results.

We can conclude from the results that the Harris method works well for images with bright corners that stand out in the background. The method is invariant to rotation and has a fairly high running speed. However, it does not find special points on images of different scales, in the presence of noise and differences in brightness.

4. Approbation for creating 3D models

Program module should be imported into Blender 3D. By clicking the button, user opens file browser in which a reference image file can be



a)

b)

FIG. 6. (color online) Examples of Harris (a) and Shi–Tomazi (b) detector test non-contrast noisy images.

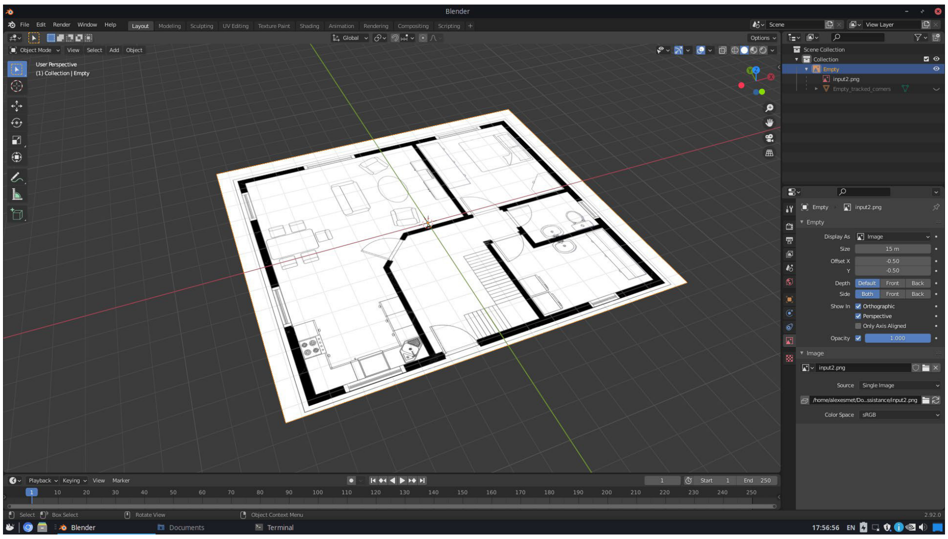


FIG. 7: (color online) Importing reference image.

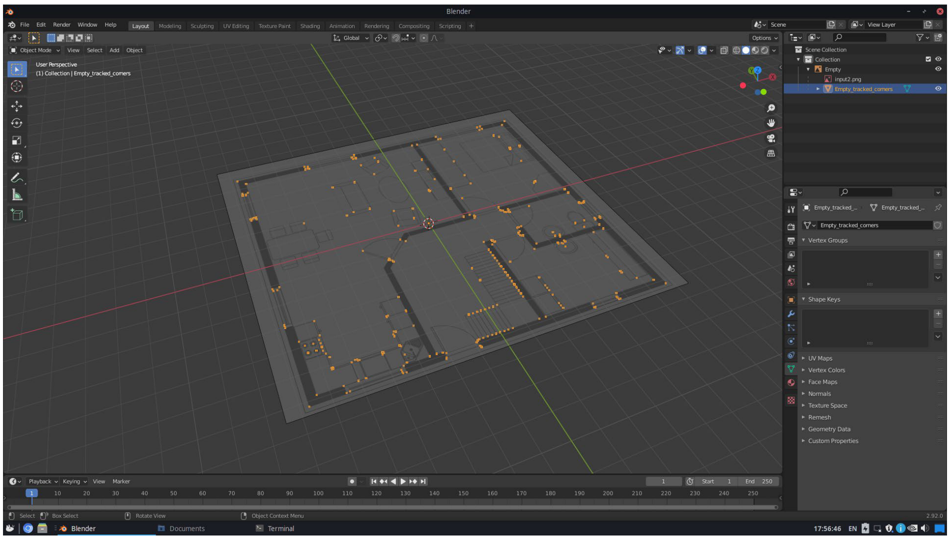


FIG. 8: (color online) Corners detected by the program module.

located. By default, imported image will be automatically oriented to the current view (Figure 7).

When image is imported and selected, user is able to track corners on it. If plug-in import has gone correctly, "Track Corners" item should appear at the end of the Object menu. Corners should appear as points on top of the selected image, as well as the program module configuration dialog in the bottom of the screen. By default, configuration dialog is contracted, user should locate it end expand by clicking an arrow icon in its corner.

By changing values in the appeared dialog, user is able to select how much corners should be detected. Essentially, user should increase amount of detected corners until all visible corners are selected (Figure 8). User is also able to change quality of the detected corners, applying a threshold to the detected corners.

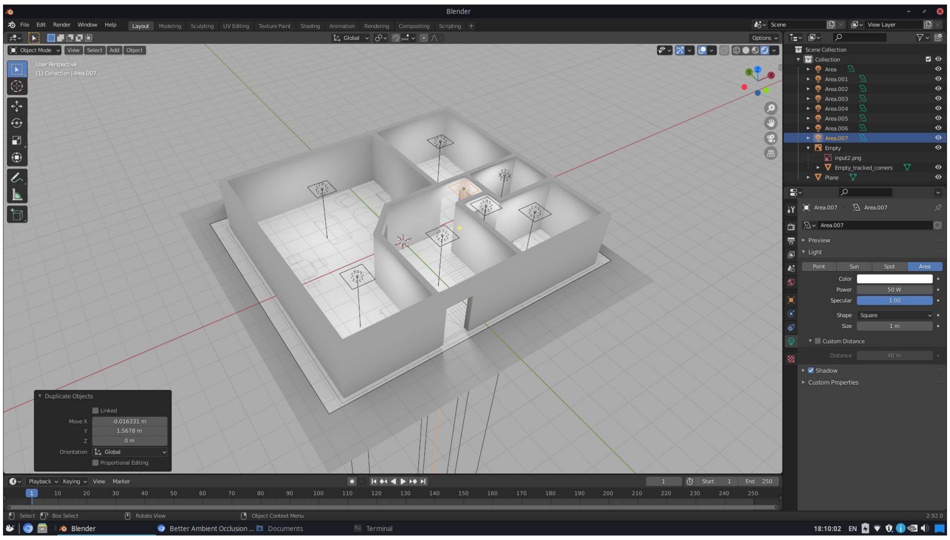


FIG. 9: Extruded filled outline.

In order to build a model on top of the set of detected points, vertex snapping must be enabled. Vertex snapping is a Blender 3D tool that automatically moves an object that is being transformed to the point that is close to the mouse pointer. Snap-ping is enabled by clicking the magnet icon at the top of the 3D view, and vertex mode is enabled in dropdown 1menu near the magnet icon.

Vertex snapping is used to assist vertex extrusion. Extrusion is an operation of increasing amount of objects dimensions by copying it and linking corresponding components to each other. Building's outline is created by connecting corners. This can be done by selecting a vertices that corresponds to a corner, and extruding it to the corner. Since extrusion is a transformation, dragged corner will be snapped to another one as vertex snapping is enabled.

The problem is that when points are snapped, there are actually two vertices having the same coordinates. This can cause issues with mesh consistency, selection and model export. In order to solve this issue, automatic vertex merging should be enabled. This tool automatically removes vertex duplicates. Vertex merging is enabled at the top right corner of the 3D view, in Edit mode, in options dropdown.

If user has correctly followed all instructions, result of this actions is buildings' outline. Next step is to use automatic triangulation to fill this outline with faces. Filled outline can be extruded upwards in order to give wall their volume (Figure 9). Presumably, user extends walls up to the height of doors in this building.

5. Conclusions

As a result of this research, a comparative analysis of Harris detector and Shi–Tomasi performance on images of project plans has been carried out. Considering the fact that the Shi–Tomasi method is based on the Harris detector, we found a lot in common when comparing their results in practice. The Shi–Tomasi method is also not invariant to changes of scale and noise, partially invariant to changes in brightness and independent of rotation. On the whole it shows better results and less often makes mistakes when finding singular points on diagonal edges and circles. The developed Shi–Tomasi detector software module will be used for further implementation of the space exploration process in Blender. This utility is an additional module with which to import into Blender images of floor plans on which all found special points of angles are marked [2].

References

[1] O.M. Demidenko, N.A. Aksionova, A.V. Varuyeu, A.I. Kucharav. 3D-modeling of Augmented Reality objects using Shi-Tomasi corner detection algorithms. J. Phys. CS. **2091**, 012058 (2021).

[2] N. Aksionova, A. Kucharav. Development of SDK for a mobile application using augmented reality technology. Bulletin of F. Skorina Gomel State University. **126**, 81 (2021).

[3] R. Haralick, L. Shapiro. *Computer and Robot Vision.* (Addison-Wesley, Reading, Massachusetts, USA, 1992). Pp. 630.

[4] C. Harris, M. Stephens. A Combined Corner and Edge Detector. In: *Proceedings of the 4th Alvey Vision Conference.* (1988). Pp. 147–151.

[5] J. Shi, C. Tomasi. Good Features to Track. In: *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition.* (1994). Pp. 593–600.

[6] H. Moravec. Rover visual obstacle avoidance. In: *International Joint Conference on Artificial Intelligence.* (Vancouver, Canada, 1981). Pp. 785-790.

[7] V. Rodehorst, A.F. Koschan. Comparison and Evaluation of Feature Point Detectors. In: *Proceedings of the 5th International Symposium Turkish-German Joint Geodetic Days Geodesy and Geoinformation in the Service of our Daily Life.* (Berlin, Germany, 2006). Pp. 1–8.