Известия Гомельского государственного университета
имени Ф. Скорины, № 6 (141), 2023

Информатика

УДК 004.925.83:004.93 EDN: KPHNIN

Method of construction of three-dimensional structures based on key corner points

N.A. AKSIONOVA, D.S. SYCH, A.V. VARUYEU

The article describes a method of construction of three-dimensional structures based on the computer vision technology that provides for automation of the process of 3D modeling of objects. A two-dimensional image is reconstructed into a relevant three-dimensional structure based on recognized key points of architectural plans. A Python-based module for pre-processing of an image and for identification of corner points of interest is developed. A method of construction of three-dimensional structures in Blender3D editor with the use of computer vision elements is presented. Module testing and approbation are demonstrated.
Keywords: pattern recognition, 3D-modeling, points of interest, computer vision, Blender, Python, OpenCV.

В статье описывается способ построения трехмерных структур с использованием технологии компьютерного зрения, позволяющий автоматизировать процесс 3D-моделирования объектов. На основе распознанных ключевых точек архитектурных планов двумерное изображение реконструируется в соответствующую ему трехмерную структуру. Разработан модуль предварительной обработки изображений и нахождения особых точек углов на языке программирования Python. Предложена методика построения трехмерных структур в редакторе Blender3D с использованием элементов машинного зрения. Приведено тестирование и апробация работы модуля.
Ключевые слова: распознавание образов, 3D-моделирование, особые точки, компьютерное зрение, Blender, Python, OpenCV.

Introduction. The state-of-the level of development of software and information technologies makes it possible to represent the reality around us in the format of 3D modeling. The technology of augmented reality (AR technology) provides for overlaying of digital prototypes on actual objects in real-time mode [1]. One of the most demanded areas for development of AR technology lies the field of design and architecture. The analysis of international experience in architectural visualization has shown that augmented reality systems are mostly focused on describing and displaying 3D models of buildings, structures and construction objects. At the same time, the automation of 3D modeling process is a challenge.

Architectural visualization is a modeling of a development facility in a digital (virtual) space. The existing AR-systems provide for ability to perform a panoramic flight, to add animated elements, to conduct section by section review and to get acknowledged with floor plans.

At the same time, the resource capacity of visualization of three-dimensional structures of AR-systems requires the computing system to perform technological operations in a mode close to real time. Increasing the resolution requirements of the final image and the detail of the AR system makes the calculations complex, which leads to significant delays in rendering.

For example, to obtain a high-quality image with a resolution of 2500 × 1400 on a computer with CPU Intel Core i7-980 and RAM 24Gb requires a calculation time of at least 2,5 hours. Such a long interval excludes the mode of direct user interaction with the AR system. Current ways to reduce time delays: reducing the level of detail of AR-objects and using distributed rendering tools.

The procedure for analysis of the existing AR-systems was followed to assess the functional capabilities to recognize architectural plans in 3D modeling software for the subsequent modeling process.

The listed user applications are created for each project by inclusion of existing 3D models with use of such standard file formats as FBX, 3DS, OBJ. Digital 3D objects are displayed in real-time mode, but their designing and realization in 3D-modeling software takes a fair amount of time [2]. Thus, a search for methods of automation of construction of three-dimensional structures with use of computer vision elements is a relevant aspect of the main challenge of 3D modeling.

Blender is a software complex for three-dimensional modeling of open source objects. This package distinguishes itself with high flexibility and self-supportability, since it contains a comprehensive set of software tools to maintain the whole process chain of dynamic computer visualization of any complexity.

Support of Python programming language, which is interpreted by Blender, elevates Blender to the level of a full-scale modeling tool comparable in features with MathLab and similar software. This feature ensures full control of the platform tools and gives access to extensive module library created by the research community. Blender API (Application Programming Interface) Python facilitates the process of creation of additional modules extending the environment functionality.

Thus, due to the above set of features, Blender may be used as a universal software environment for objects modeling and natural research development.

The authors developed the software module «Modeling-assistance» in Python. The module is based on the function of determining the angle of development plans for further modeling of architectural projects, buildings, houses, apartments, etc. This process facilitates the construction of a three-dimensional model, ensures that initial adjustments are made in the future to a three-dimensional model that does not require changes in the design documentation itself. An author-developed variant of construction of three-dimensional structures in Blender involving use of computer vision elements is suggested in the articles. Visualization of the models developed with the use of augmented reality technology is presented.

Description of the research methodology. To apply the augmented reality system to architectural projects, a method of detection of characteristic features of input images is selected [3]. The concept of feature detection by computer vision refers to methods that are aimed at computing image abstractions and highlighting key features on it. Since the input information is a layout of apartments, houses, buildings, etc., isolated points representing structure corners are chosen as features. A key point is a certain area of an image, which distinguishes from the remaining parts on the given image.

To detect points of interest on an image, a computer vision algorithm needs to be realized. Algorithms of corner detection serve as the main algorithms of computer vision. The articles [4], [5] of the named study describe Python-based development of Shi-Tomasi and Harris corner detectors with the use of OpenCV, present a mathematical tool of these algorithms, demonstrate a comparative analysis and testing (approbation) of the developed software modules.

The developed «Modelling-assistance» software module is an addon for 3D-modeling software. Addons may not be started by themselves, but they are to be imported in software, therefore the module does not contain a directory with binary files. The module contains only the source code directory, the module package, and the text file with instructions on how to import the module into modeling software and how to package the module from the source, if necessary.

The OpenCV library realizes the algorithm of Shi-Tomasi corner detection, which is enclosed in «GoodFeaturesToTrack» function. The first function parameter is the image object in greyscale. This function adds some logic to the algorithm. First of all, it selects N nest corners detected on the image. The target number of corners to be detected is represented as the second function parameter. In other words, it converts an image matrix into a list of coordinates. Secondly, it applies a threshold limit for removal of the worst corners from the list. The threshold limit serves as the third function parameter. Thirdly, this function is able to remove corners located too close to each other, and this value serves as the fourth function parameter.

The method of 3D-model construction in Blender editor is demonstrated in Figure 1.

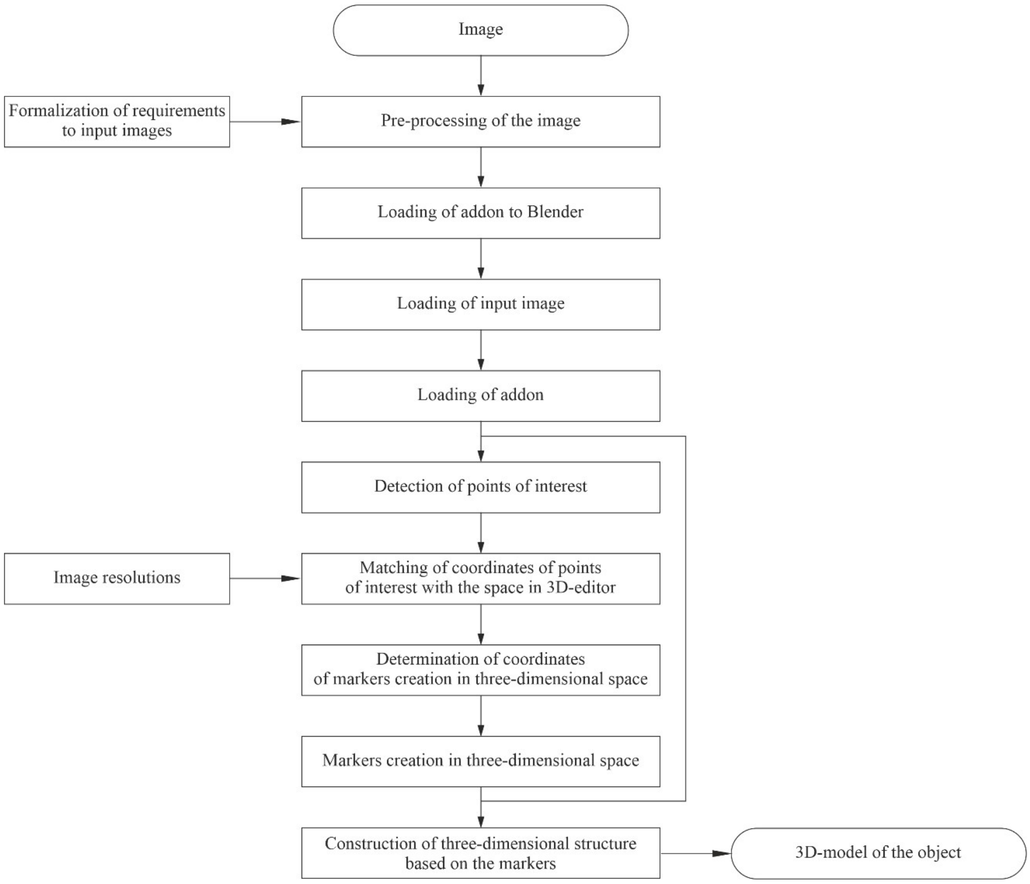

Figure 1 – Method of 3D-model construction based on key points

A digital image of a given size serves as input. The image is imported in Blender after pre-processing. To use the image, it is recommended to delete all texts, arrows and legends. Such extra data complicate the recognition process.

The recommended parameters of input image:

– high resolution with no compression artefacts;
– export to PNG format;
– conversion of color image into binary image.

«GoodFeaturesToTrack» corner detection function returns the array of coordinates of points of interest in the space of pixel coordinates of the image, where the reference point is located in the upper right corner. X and Y coordinates in Blender 3D space are specified in the metric, while the reference point is located in the upper right corner. This specific feature means that the corner coordinates shall match to the editor space.

Let x and y be the input coordinates of the image. To convert the reference points, the array of coordinates needs to be converted. For the above purpose, the size (S) of the reference image is required. The fact that the reference image is shifted relatively to its original position by 50 % up and 50 % sidewards needs to be taken into account. Thus, the reference point is automatically located in the center of the image. The conversion formula is as follows:

$$f(x) = \frac{\left(x/d_X + O_X \right) \cdot S \cdot d_X}{\max(d_X, d_Y)},$$

$$f(y) = \frac{\left(y/d_Y + O_Y \right) \cdot S \cdot d_Y}{\max(d_X, d_Y)},$$

where d_X and d_Y are the sizes of the image in pixels, O_X and O_Y are the values of the image shift in percent, S is the size of the reference image.

When the points from 2D and 3D spaces are matched, the values generated by the function of detection of specific corner points need to be deleted and the zero-coordinates by Z need to be matched to them. Thus, the coordinates of corner markers are received. The obtained coordinates are local. Therefore, such transformations as shifting, rotation or scaling are unavailable. This problem may be solved by adding the created object to the parent one. The sequence of «Create a 3D-model» case was presented in the article [6].

When switching to a distributed form of rendering, the load balancing between data processing nodes is planned by means of an external module. An example of such plug-in VST modules is VRay, Arnold, Redshift, Octane. Delays in the network environment of the cluster should be minimal and their value should be significantly less than the processing time of the visualization package for an individual node. Network storage for receiving input data for rendering and saving the results must be built using SAN or NAS technologies to meet the requirements of reliable data storage [7].

3D modeling stage shall finish with 3D model file export to OBJ format. The file itself is a text file with the list of vertexes and their coordinates, the list of faces and edges joined by these vertexes.

OBJ file format supports both approximate and accurate surface geometry coding. In case of approximate coding, the surface grid is not limited with triangular faces. Upon a user's choice, polygons like quadrangles may be used. In case of accurate coding sweet lines and surfaces, such as NURBS, are used. OBJ format provides for coding the information on color and texture. Such information is stored in a separate file with .MTL extension (material template library). It does not support any animation. The format is detected both as ASCII, and as a binary coding, but only ASCII is an open code. OBJ file format, due to its neutrality and openness, is one of the most popular exchange formats for 3D graphics. It also becomes popular in 3D printing industry, since the industry moves towards full-color printing.

The results of implementation of the method. Prior to using the addon module, scene settings in Blender editor need to be verified. Module running assumes that the scene contains imported reference image and that the given image is active.

Verification stages:

1 The object is selected.
2 The object is a «link» to the image.
3 The image is imported correctly.

The reference objects in Blender are realized as extended «blank» objects. A blank object is what happens in 3D space, but is not a 3D model. A reference to an image is one of possible blank object types. Then the image used is loaded to OpenCV library. It is converted into gray shades and corner points of interest are searched for.

The addon shall be imported in Blender 3D. To do this, the main menu is used. The settings window is opened in «Edit» menu, and then the user may use «Install» button on «Addons» tab. The file browser opens; once the script with the addon is opened, the addon is imported. Provided that there are no errors, a successful image import message is displayed and the addon is added to the list of installed addons.

By pressing the button the file browser is opened, where the file of the reference image may be found. By default the imported image is automatically oriented by the current view, but this function may be turned off in the same file browser.

When the image is imported and selected, the user may track the corners on it. If plug-in is imported correctly, «Track corners» option appears in «Object» menu. The addon code is run, when this option is selected. Corners shall appear as points on the selected image, as well as the addon setting dialog shall appear in the lower part of the screen. The configuration dialog window is rolled down by default, the user may find it and roll out by clicking the arrow icon in the corner. It is important that the reference image is selected by the moment of clicking the button, otherwise the addon is not able to detect the image and is not able to work properly.

By changing the values in the dialog window displayed, the user may specify how many corners need to be detected. In practice, the user shall increase the number of corners to be detected till all visible corners are detected. The user may also change the quality of corners detected by using the threshold value for corner detection. Another available feature is the minimum distance between the corners. By changing this parameter, the user may filter point clusters. It is recommended to specify such parameter as wall thickness in pixels.

At that moment, one of the challenges has been identified. When the configuration values are changed by drag-and-drop using input as a component of the user slider interface, the main addon function is realized for each slider step, noticeably slowing down the application. To solve the challenge, the computer vision algorithm was updated to have higher performance and lower resource consumption. The problem cause has been eliminated in the collection of the standard library of Python used for matching the values generated by the computer vision algorithm with the coordinates of 3D space in Blender. The problem was solved by introduction of NumPy library, the logics of which is realized on low-level faster programming languages.

To construct a model above the set of detected points, snapping of vertexes needs to be activated. Snapping of vertexes is a tool of Blender 3D, which is automatically transfers the transformed object to the point close to the mouse pointer. Snapping is activated by clicking on the magnet icon in the upper part of 3D view, while the vertex mode is activated in the pull down menu close to the magnet icon (Figure 2).

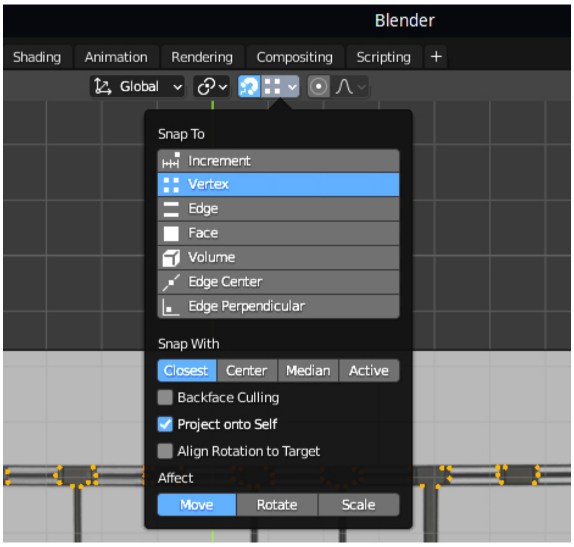

Figure 2 – Snapping of vertexes

Snapping of vertexes is used to facilitate extrusion of vertexes. Extrusion is an operation of zooming objects by their copying and snapping of relevant components to each other. The building contours are created by linking corners. To do this, a vertex corresponding to an angle is selected and is extruded to another angle. Since extrusion is a transformation, the moved corner is linked to another corner due to the activated snapping to the vertex.

The problem is that in case of snapping of points, there occurs existence of two vertexes with the same coordinates. It may cause trouble with grid coordination, model selection and export. To solve this problem, automatic vertex merging needs to be turned on. This tool automatically deletes vertex duplicates. Vertex merging is turned on in the upper right corner of 3D view in edit mode in the roll down list of parameters (Figure 3).

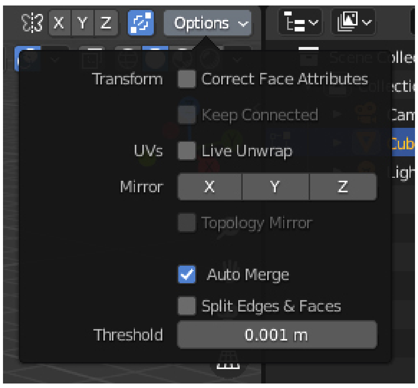

Figure 3 – Turning on Auto Merge tool to delete vertex duplicates

When all auxiliary tools are turned on, the actual modeling process may be started. Building contour is created by dragging points to their relevant neighboring points. If by any cause a vertex for some specific point is absent, the user may temporarily turn off vertex snapping by holding Ctrl.

It can be noticed that regardless that corner detection searches for all points that may be useful, there is still a probability that some points are not detected. It is possible in cases when certain design elements contain round objects. But the advantage of the approach is that it does not limit users, since the user still may add own models, such as rounds, to the grid. If the user follows all the instructions correctly, his actions result in building contour (Figure 4).

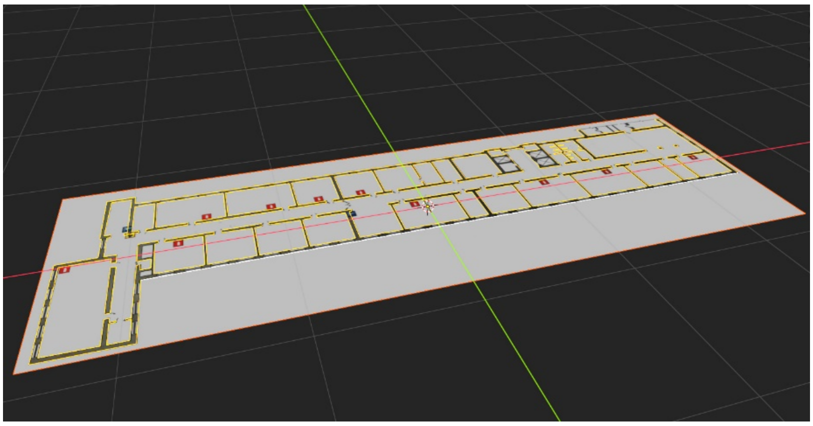

Figure 4 – Resulting building sketch

The next step is to use automatic triangulation to fill in the contour with faces. The filled in contour may be extruded up to giving the wall volume. The user presumably extends the wall till the height of the doors of the building.

At each level of the extended contour the user may alter it by, for example, linking void spaces above the doors. The user shall control the generated face orientation, since the automatic face orientation is not always correct.

Surface normal is a vector in space, which is perpendicular to the face and which indicates its direction providing for «internal» and «external» surfaces. By default, Unity does not display internal faces, therefore correct model display requires all surface normals be outwards.

It also shall be noted that doors, windows and stairs shall not be mandatory created during the process. Since the result of the process is a simple grid, the user may change it with standard tools. For example, the user may use Array modifier to create the array of cubes with the dimensions of one building window, or Boolean modifier to disjoint one volume from another.

Upon all these operations a model is ready for export (Figure 5). The next step is exporting the model from Blender 3D. This is done using file menu. Such formats as OBJ and FBX are available.

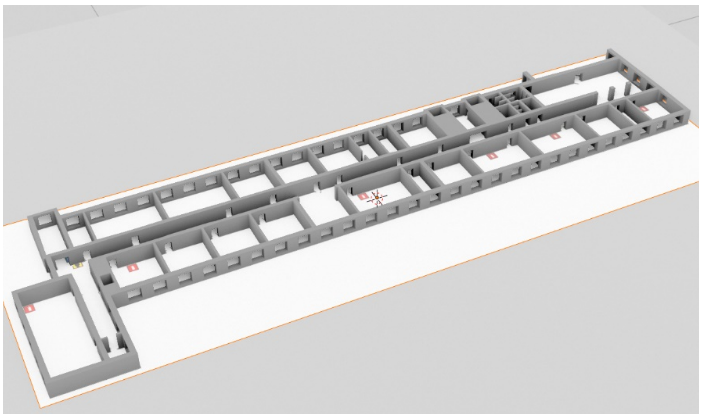

Figure 5 – Resulting 3D model

Conclusion. Development of object modeling methods in the environment of open software platform of Blender demonstrated its competence as a tool suitable for solution of three-dimensional spaces modeling challenges with use of computer vision elements. By means of Python-based software module, with the use of Blender API, a number of typical goals of automated search for key points on images of architectural plans were achieved and 3D models of augmented reality objects were constructed.

References

1. Aksionova, N. A. Development of SDK for a mobile application using augmented reality technology / N. A. Aksionova, A. I. Kucharav // Bulletin of F. Skorina Gomel State University. – 2021. – Vol. 126 (3). – P. 81–84.

2. Aksionova, N. A. Presentation and implementation of educational material using augmented reality technology / N. A. Aksionova, A. V. Varuyeu, A. I. Kucharav // Trends and prospects for the development of science and education in the context of globalization : electronic collection of materials of the International scient. conf. – 2021. – Vol. 71. – P. 235–239.

3. Varuyeu, A. V. Information capacity of objects in conjunction with augmented reality / A. V. Varuyeu // Bulletin of F. Skorina Gomel State University. – 2021. – Vol. 126 (3). – P. 92–95.

4. Demidenko, O. M. Development of a machine vision system for image recognition of design estimates / O. M. Demidenko, N. A. Aksionova // Nonlinear Phenomena in Complex Systems. – 2022. – Vol. 25 (2). – P. 159–167. – DOI : 10.33581/1561-4085-2022-25-2-159-167.

5. Demidenko, O. M. 3D-modeling of augmented reality objects using Shi-Tomasi corner detection algorithms / O. M. Demidenko, N. A. Aksionova, A. V. Varuyeu, A. I. Kucharav // J. Phys. – 2021. – Vol. 2091 (012058). – DOI : 10.1088/1742-6596/2091/1/012058.

6. Aksionova, N. A. Construction of three-dimensional structures based on the recognition of key points / N. A. Aksionova, A. V. Varuyeu, O. M. Demidenko // Information systems and technologies : materials of the international scientific congress on informatics. – Minsk : BGU, 2022. – Vol. 3 (2). – P. 99–105.

7. Demidenko, O. M. Techniques of adapting a calculating process to operating load of a local area network / O. M. Demidenko // International Conference on Information, Control, and Communication Technologies (ICCT), Astrakhan, 03–07 October 2022. – Astrakhan, 2022. – P. 1–6. – DOI : 10.1109/ICCT56057.2022.9976851.

Гомельский государственный
университет имени Франциска Скорины Поступила в редакцию 31.10.2023