International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Text to 3D Virtual Layout Translation with Intermediate Panoramic Translation

Nikhil Singh Rathaur[1], Shubham[2], Ramesh Naik S[3], Vinayaka Udupa[4]

[1,2,3,4]*Student, Dept. of Computer Science & Engineering, The National Institute of Engineering, Mysore, Karnataka, India*

Abstract - *Through deep learning we have achieved a lot in this area but layout designing and its 3-Dimensional visualization is still something that can only be done using some specialized softwares which can only be handled by some highly trained professionals. This project can be treated as a first step towards achieving an automated and economically viable solution through which any person will be able to visualize a 3-Dimensional Layout view by just giving a text description. We are trying to integrate the learning of four different researches into one single process pipeline. The four processes involved here in order are Attention Generative Adversarial Networks, content preserving image stitching to create a panoramic image, generating the wireframe and aligned lines from this panorama and finally generating the 3D layout using the panorama and wireframe. This project aims to achieve a considerably satisfactory output which scope for future improvements on the same.*

Key Words: **Generative Adversarial Networks, Architectural Modelling, Visualization, Scene Reconstruction, 3D Object, Panorama Creation, Manhattan Wireframe, Layout Generation**

1.INTRODUCTION

This project can be treated as a first step towards achieving an automated and economically viable solution through which any person will be able to visualize a 3-Dimensional Layout view by just giving a text description. We are trying to integrate the learning of four different researches into one single process pipeline. The pipeline starts with taking multiple input text descriptions which includes the walls, the room accessories, details about open spaces and windows, etc. This text description goes into an AttnGAN model which generates one square image for each description. Now we stitch these images together with regular boundary constraints and convert it into a long panoramic image. This panoramic image goes inside a Manhattan wireframe generator and we now have the Manhattan lines of the layout. Now the panoramic image and the wireframe go inside another final network called the LayoutNet which generates a possible 3-Dimensional view. So overall, we give an initial input of a text description and we obtain the final output as a 3-Dimensional layout. There's a good scope of improvement in this research since it's a maiden attempt to do achieve the intended results. The initial dataset for Attention Generative Adversarial Network is customized manually since the text descriptions weren't available in the original dataset with a total of 5000 image-text pairs. This project aims to achieve a considerably satisfactory output which scope for future improvements on the same.

2. EXISTING INDIVIDUAL PROCESSES

The StackGAN[1] architecture for text to 2-Dimensional image generation suggests a way to break the hard problem into more convenient sub-problems through a sketch-refinement process. The First stage of GAN sketches the simple shape and colors of the object based on the given text description, providing fist stage low-resolution images. The second stage GAN takes previous stage 1 results and text descriptions as inputs, and produces good-resolution images with very realistic details. It is able to correct defects in previous stage results and add variety of details with the correction process. To provide some variety of the synthesized images and make the training stable for the conditional-GAN, researchers introduced a Conditioning Augmentation method that provides smoothness in the latent conditioning manifold.

AttnGAN [2] for fine grained text to image translation is used here, Attentional Generative Adversarial Network (AttnGAN) that allows a very model which is attention-driven, with multiple refining stages for fine grained text-to-image generation. With an effective attentional generative network, the AttnGAN can produce veru minute details at different subregions of the picture/image by paying attentions to the relevant words in the natural language description. A key addition is deep attentional multimodal similarity model is proposed to compute the total loss that incurs while the text is matched with image.

Content Preserving Image Stitching with Regular Boundary Constraints[3]. This paper proposes a way to deal with preserving the content of various pictures while sewing with customary limit requirements, which then gives us the panoramic image as the final output with all contents preserved.

Robust Image Stitching with Multiple Registrations[4]. The problem is simplified into three stages: registration, which picks a single transformation of every single image to align it to the various other inputs, seam finding, which selects an

original image for each pixel in the final result, and blending, which corrects any errors in visualization.

Lifting 3D Manhattan Lines from a Single Image[5]. The line detection identifies a good number of different explicit lines that intersect or nearly intersect in an image, but relatively aa very few of them actually contribute to a physical 3D junction. Here, linear programming (LP) is used to identify a minimal set of least-violated connectivity limitations that are enough to unambiguously reconstruct the 3D lines.

Learning to Reconstruct 3D Manhattan Wireframes from a Single Image[6]. In this paper, the researchers propose a technique to get a minimized and exact 3D wireframe portrayal from a solitary picture by successfully exploiting worldwide underlying consistencies. This method trains a convolutional neural network to parallelly detect effective junctions and straight lines, as well as predict their 3D vanishing point and the depths.

Automatic 3D Indoor Scene Modeling from Single Panorama[7]. This system recovers the spatial layout by finding the floor, walls, and ceiling; it also recovers shapes of typical indoor objects such as furniture. Using sampled perspective subviews, we extract geometric cues (lines, vanishing points, orientation map, and surface normals) and semantic cues (saliency and object detection information)

LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image[8]. It proposes a calculation to anticipate room format from a solitary picture that sums up across scenes and point of view pictures, cuboid designs and more broad designs (e.g. L-shape room). This method operates directly on the one panoramic image, rather than decomposing into perspective images as proposed in other research work sin the same domain. This network architecture is similar to that of the older version called RoomNet, but it reflected significant improvements due to aligning the image based on vanishing points, predicting multiple layout component (corners, boundaries, size and translation), and fitting a constrained Manhattan layout to the resulting predictions.

3. PROPOSED SYSTEM

Our proposed system is a unique attempt to transform the process of 3-Dimensional interior designing. The existing solutions discussed in the previous section are all manual which means that you need to have an extensive training and hands-on experience of the softwares. In our proposed system, one only needs to provide a text description of the interior and the system tries to generate the most accurate layout according to its understanding in a virtual 3D sense. It is primarily based on following models: (a) AttnGAN (Attention Generative Adversarial Network) (b) Content Preserving Image Stitching (c) Reconstructing 3D Manhattan Wireframes from a Single Image (d) LayoutNet.

This process pipeline is shown in Fig-1 for your reference.

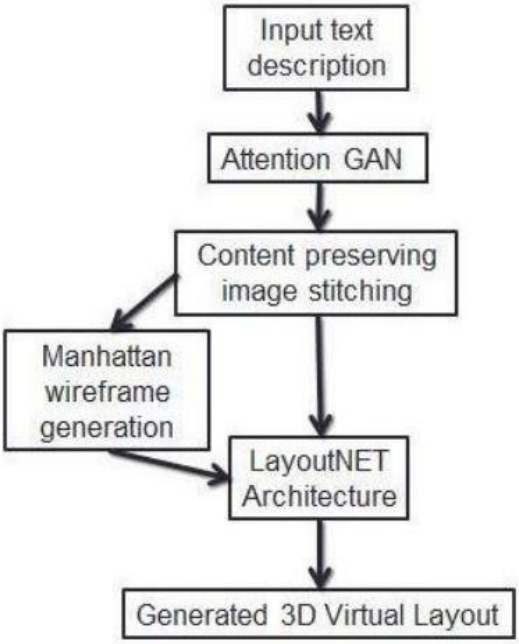

Fig-1: The workflow of the proposed system

3.1 AttnGAN

Attentional Generative Adversarial Network (AttnGAN) that allows a very model which is attention-driven, with multiple refining stages for fine grained text-to-image generation. With an effective attentional generative network, the AttnGAN can produce very minute details at different subregions of the picture/image by paying attentions to the relevant words in the natural language description. A key addition is deep attentional multimodal similarity model is proposed to compute the total loss that incurs while the text is matched with image.

This particular model is composed of two major components. The first part is an attentional generative network, in which we equip an attention mechanism for the generator to draw different sub-regions of the image by focusing on words that are more closely related to the sub-region being drawn. In more detail, apart from encoding the natural language description into a global sentence vector, each word in the sentence is also encoded into a word vector. The generative network uses the global sentence vector to produce a low-goal picture in the primary stage. In the subsequent next stages, it uses the vector associated with image in each sub-region to query word vectors to form a word-context vector. After this, it combines the regional image vector and the corresponding word-context vector to form a multimodal context vector, based on which the model generates new image features in the nearby sub-regions. Doing this produces higher resolution images at later stages.

Another important component in AttnGAN is a Deep Attentional Multimodal Similarity Model (DAMSM). With an attention mechanism, the DAMSM is ready to process the similitude between the created picture and the sentence utilizing both the global sentence level data and the fine-grained word level data. Thus, the DAMSM helps further in correcting any errors between the image-description pairs.

3.2 Content Preserving Image Stitching

After obtaining multiple images from the AttnGAN network, they are stitched together with regular boundary constraints and at the same time keeping the content of the images preserved. We address these limitations by formulating image stitching with regular boundaries in a unified optimization.

Beginning from the underlying sewing results delivered by conventional distorting based advancement, we acquire the sporadic limit from the twisted cross sections by polygon Boolean tasks which robustly handle subjective lattice structures, and by examining the unpredictable boundary build a piecewise rectangular boundary. In light of this, we further fuse straight line safeguarding and customary boundary limitations into the picture sewing framework and lead iterative enhancement to acquire an ideal piecewise rectangular limit, thus can make the limit of sewing results as close as conceivable to a square shape, while decreasing undesirable contortions. This technique is based on the following observations: (1) Rectangling and sewing are firmly related, and improving the two cycles all the while can assist with delivering better rectangular displays in a content-aware way. (2) The point of rectangling the panorama is to save however much picture content as could reasonably be expected in a rectangular window while keeping away from unforeseen mutilations. To achieve this, the uneven boundary should not be simply into one independent rectangle. It was suggested to instead use a better partwise rectangular boundary to make sure of the regularity while avoiding excessive errors. Doing it this way also has another benefit that treats conventional rectangular boundaries as a different case, and will ensure rectangular results are proper.

3.3 Reconstruct Wireframes from a Single Image

It's a strategy to get a minimized and exact 3D wireframe portrayal from a solitary picture by adequately exploiting global structural normalities. This technique trains a convolutional neural network to simultaneously identify notable intersections and straight lines, and also make predictions over 3-Dimensional depth and other vanishing points. When comparing it to other latest techniques and methods, this architecture is much more primitive and more compact, which leads to better 2D detection of the lines and wireframe. With global structural prior assumptions such as Manhattan, this technique moreover recreates a complete 3D wireframe layout. When compared to other algorithms to detect the wireframes, this algorithm (a) requires only single model, exploiting the warped relationship between the geometrical structures; (b) identifies the difference between different kinds of junctions: the physical intersections of lines and planes the “C type junctions”, and the other “T type junctions”; (c) recreates a complete 3-Dimensional wireframe of the structure from different lines and joints detected in one specific RGB image.

3.4 Reconstructing 3D Interior Layout

This algorithm is used to predict the layout of the interior of a room from a single RGB image that generalizes across various panoramas and long perspective pictures, polygon-based layouts and more primitive layouts. This technique operates straightaway on perspective panoramic image, instead of breaking and dividing into straight square images. This build and architecture of this network is similar to that of RoomNet, but it displays vast corrections because image aligning is based on vanishing points, ability to predict various elements of a standard layout (corners, limiting boundaries, shapes and translation), and adjusting a Manhattan layout to the output results or predictions. This technique proves effective in terms of speed & accuracy in contrast with other works based on panoramas, achieved better accuracy for other perspective images, and can handle any polygon-based Manhattan layout.

The approach of LayoutNet can be understood in three steps. First, the framework observes the vanishing points and aligns the picture to be level to the floor. This practice makes sure that wall-wall boundaries are perpendicular lines and and hence it reduces scope for error. In the second step, the vertices or corner junctions and the maps for boundary probability are predicted straightway on the RGB picture using a Convolutional Neural Network with an encoder-decoder structure with a provision of skipping connections. Both- boundaries and corner vertices provide a full representation of the interior room layout. Predicting them together as a part of a single network provides a better estimation. In the third and final step, the parameters of the 3D layout are enhanced to fit and adjust the boundaries and corners that were predicted.

4. IMPLEMENTATION AND RESULTS

4.1 Generating square images from text

As proposed, the system starts with the AttnGAN model. The text descriptions are documented at a single place with the name of the file being same as the corresponding image. The dataset used in this case is the Stanford 2D 3D dataset. We trained the model the model on 20 epochs with 1050 iterations. On a NVIDA DGX GPU, the training completed in 21 hours for one single image. We also took help from

OpenAI's CLIP[10] to align the context of description with the image sub-region. We did this this twice to obtain 2 different images in such a way that they have some overlapping region can be overlapped to get one single panoramic image. We did I twice to obtain 2 different images.

Description 1 and Description 2 gave the images given in Fig-2 and Fig-3 respectively.

Fig-2: Generated image from Description 1

Fig-3: Generated image from Description 2

4.2 Panorama Generation using SIFT

SIFT[11] or Scale Invariant Feature Transform is an algorithm which is used to establish a match between two images (maybe of different dimensions). We used this algorithm to detect the key points in both the images. When key points in both the images are established, the algorithm tries to match the key points which exist in both the images. Doing this ensures that we get the common points in both the images and then we can overlap them with each other hence converted into a panorama.

To detect the keypoints, we first converted the images into grayscale to avoid any kind of RGB interference in the detection process. Applying the SIFT algorithm to obtain the keypoints in both the images gives the images obtained in Fig-4 and Fig-5.

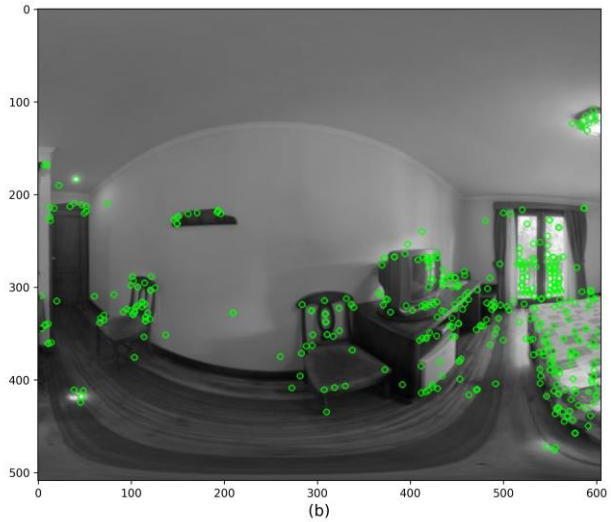

Fig-4: Keypoints detection plot for Image-1

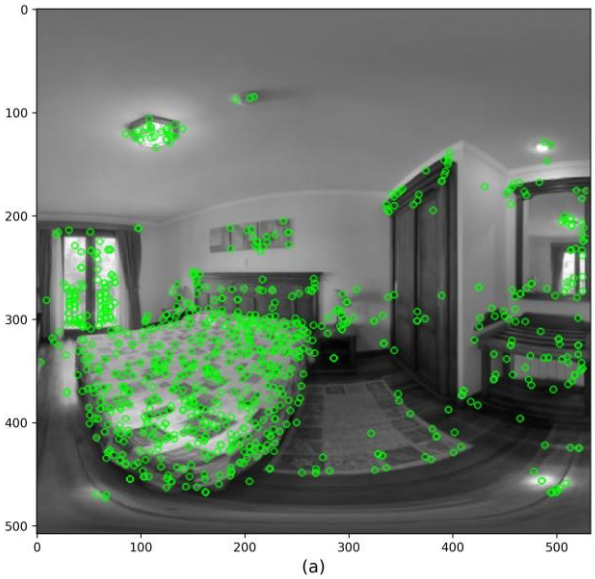

Fig-5: Keypoints detection plot for Image-2

Now that we have the keypoints, we can try matching up those keypoints of each image to get the common section image that will help us overlap these images and convert it into a panorama. We use a method called BF feature matching or in expanded terms as Brute Force Feature

Matching[12]. It accepts the descriptor of first feature in initial first set and tries matching with all different features in other set using some Euclidean distance calculation. And the most accurate one is returned. For BF matcher, we use CV to create an object of BF matcher.

We generate a plot to visualize the matches made by the BF Matcher. The plot is shown in Fig-6.

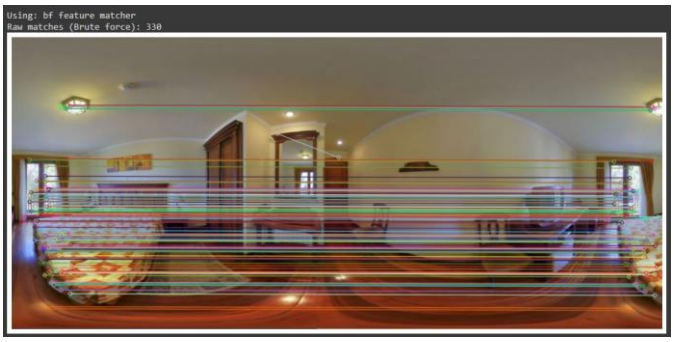

Fig-6: Matching the keypoints using Brute Force Matching

Observing Fig-4, Fig-5 and Fig-6 together, we can see that the keypoints which weren't common in both the images are not considered in the matcher plot. Also, we can see in Fig-6 that that the total Brute Force raw matches were 330 which means that a total of 330 keypoints in both the images were same and they matched with each other.

Now to make the matching region overlap with each other, we use another technique called the homography. In the area of computer vision and graphics, any two pictures of a similar planar surface in space are connected by a homography (considering the pinhole camera model). It has many practical applications in the real world, such as the rectification of an image ,the registration of an image, or camera motion based tasks —rotation and translation— between two different pictures or images. Once the extraction of camera rotation and translation is done from an estimated homography matrix, this data can possibly be used for navigation, or to introduce 3-Dimensional model and objects inside an image or video, so that they are produced and aligned with the appropriate perspective and appear to as it was a part of the original multimedia.

This process of homography join both the images side by side by overlapping the common points in both images without any major loss to the content and hence the content is majorly preserved.

The final overlapped image or simply the panoramic image is given in the Fig-7.

Fig-7: Final obtained panorama

4.3 Generating the Wireframe Aligned Lines

Now that we have obtained the panoramic image, we need to visualize the lines where the joints lie. To do this, use the concept to probability to find the joints. To perform this and do a visualization of the same, we use the Resnet-50 model. To register both the low-level features and high-level features, every block of the ResNet-50 has in itself a series of convolution layers in which the total number of channels and the vertical length is reduced by a factor of 8 (= 2×2×2) & 16 (= 4×2×2), respectively. Now the obtained features from each layer are upsampled to the same width 256 (one-fourth of input image width) and reshaped to the same vertical length. After concatenation, the final feature map is of size 1024 × 1 × 256. The activation function after every convolution layer is ReLU with an exception of the final layer where we required Sigmoid for y_w and an identity function for y_c and y_f.

The ground truth visualization of the panorama is given in the Fig-8

Fig-8: Visualization of 1D Ground Truth Representation

The grayscale bar at the top in Fig-8 represents the probability of the existence of a wall-wall joint (y_w). The green line below it is the existence of boundary for the ceiling, probability for which is denoted by (y_c). Similarly, the third line denotes the existence of the boundary for the floor, probability for which is denoted by (y_f).

These parameters together define the understanding for the align lines to be generated.

The aligned lines map after processing all the lines through custom Resnet-50 model is given below in Fig-9.

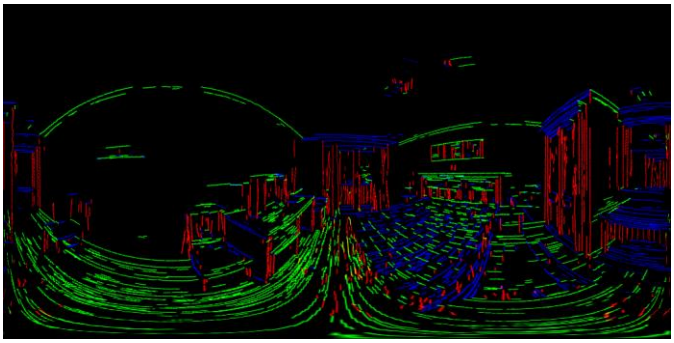

Fig-9: Aligned Lines Wireframe Map

4.4 Generating the 3D Layout

After obtaining the panoramic image and the aligned wireframe lines, we try straightening of all the curved lines and then try overlapping the panorama by customizing all the curved lines as in the aligned wireframe map. The entire information regarding the corners of the lines is stored in a JSON file. To start the conversion, we begin with converting the corners to layout. Since we are creating a 3D object, we chose to create the final output in form of a '.ply' file. PLY is a computer file format known as the Polygon File Format or the Stanford Triangle Format. We start by creating the ply's points and faces.

Now when it is done, we dump all the results in an organized manner using the Open3D python library to create an object using all these input constraint and masks.

Running the obtained 3D object file gives us our final result shown in Fig-10

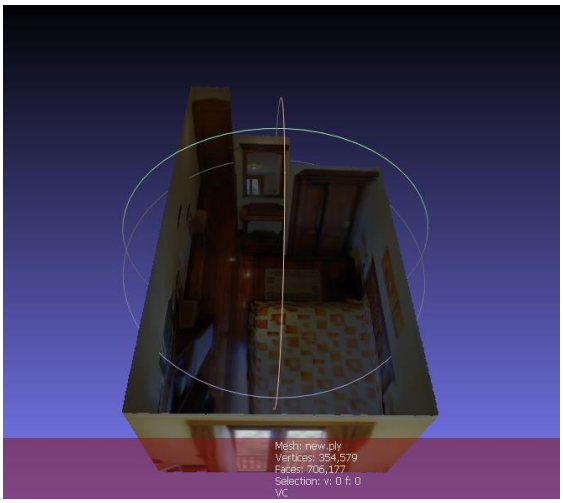

Fig-10: Final 3D Generated Layout

Views from another 2 angles are given Fig-10 and Fig-11.

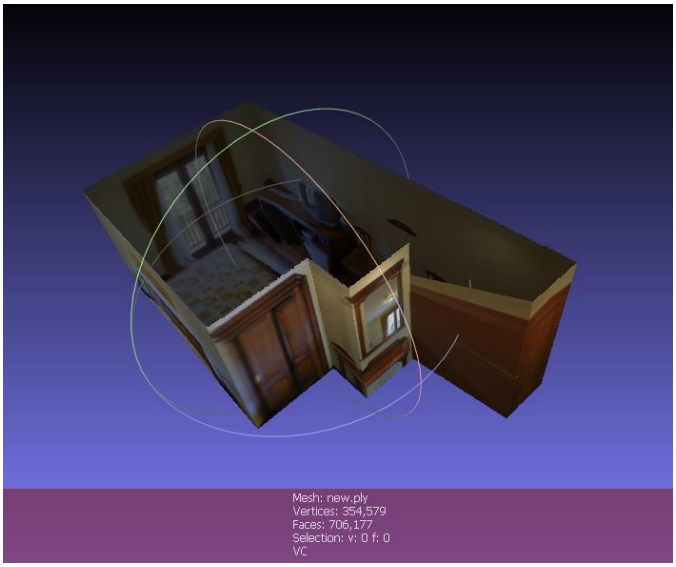

Fig-11: Top Front View

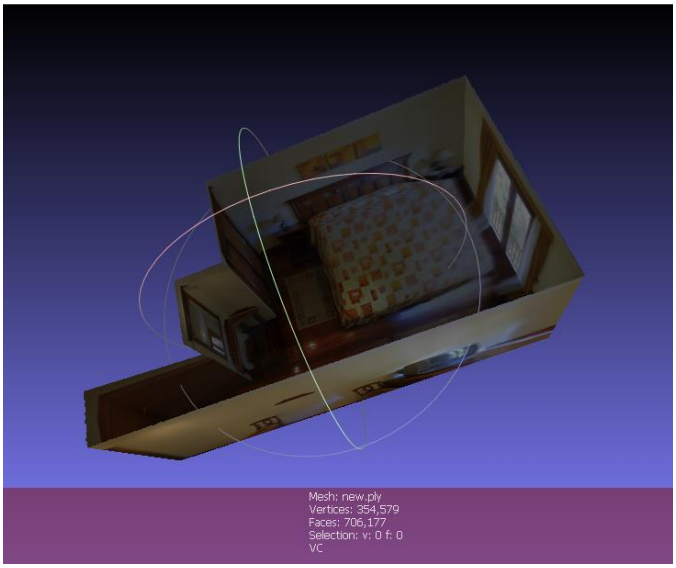

Fig-12: Top Side View

5. CONCLUSION

This technique has a great potential to be used in an effective manner by architects for fastrack layout generation in with a good quality of reconstruction. There are also a lot of existing applications like AutoCAD, usage of which require great technical skills and practice and can only be used by experienced professionals. But out solution of generating an entire layout just from text is of great advantage for people who want to design their own homes, rooms and offices without requiring any technical skills and without hiring any designers and architects and thus our solution brings on table a great potential with varied applications. This can also be used as a tool for people to have an idea about how will a room look after it is constructed and will offer them a chance to change things without committing to the construction work.

REFERENCES

[1] H. Zhang et al., "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5908-5916, doi: 10.1109/ICCV.2017.629.

[2] T. Xu et al., "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 1316-1324, doi: 10.1109/CVPR.2018.00143.

[3] Y. Zhang, Y. -K. Lai and F. -L. Zhang, "Content-Preserving Image Stitching with Piecewise Rectangular Boundary Constraints," in IEEE Transactions on Visualization and Computer Graphics, doi: 10.1109/TVCG.2020.2965097.

[4] Herrmann C. et al. (2018) Robust Image Stitching with Multiple Registrations. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11206. Springer, Cham. doi: 10.1007/978-3-030-01216-8_4

[5] S. Ramalingam and M. Brand, "Lifting 3D Manhattan Lines from a Single Image," 2013 IEEE International Conference on Computer Vision, Sydney, NSW, 2013, pp. 497-504, doi: 10.1109/ICCV.2013.67.

[6] Y. Zhou et al., "Learning to Reconstruct 3D Manhattan Wireframes From a Single Image," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 7697-7706, doi: 10.1109/ICCV.2019.00779.

[7] Y. Yang, S. Jin, R. Liu, S. B. Kang and J. Yu, "Automatic 3D Indoor Scene Modeling from Single Panorama," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 3926-3934, doi: 10.1109/CVPR.2018.00413.

[8] C. Zou, A. Colburn, Q. Shan and D. Hoiem, "LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2051-2059, doi: 10.1109/CVPR.2018.00219.

[9] 1. Budroni A, Boehm J. Automated 3D Reconstruction of Interiors from Point Clouds. International Journal of Architectural Computing. 2010;8(1):55-73. doi:10.1260/1478- 0771.8.1.55

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, & Ilya Sutskever. (2021). Learning Transferable Visual Models From Natural Language Supervision.

[11] Lindeberg, Tony. (2012). Scale Invariant Feature Transform. 10.4249/scholarpedia.10491.

[12] A. Jakubović and J. Velagić, "Image Feature Matching and Object Detection Using BruteForce Matchers," 2018 International Symposium ELMAR, 2018, pp. 83-86, doi: 10.23919/ELMAR.2018.8534641.