Sayeed Mohammed Shaik, 2022, 10:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Generating 3D Images from a Still Image Using Image Processing

Sayeed Mohammed Shaik, Associate Prof. Dr. R. Kamalraj
MCA Department,
Jain University,Bangalore,
mca.sayeed@gmail.com, r.kamalraj@jainuniversity.ac.in

Abstract- Images are a great visual communication tool and a most shared Media. Whether we are reliving a moment, communicating with an audience, or providing information. We can create an enhanced viewing experience by generating a dynamic three-dimensional composition from the extracted features of the still Image. By identifying elements of a still Image. Extracting Visual and Spatial Features from a still Image. Generating Mesh Objects in a three Dimensional workspace. Rendering the finalized composition to a dynamic view. By utilizing recent techniques and methods in Computer Vision and Image processing the process can be automated enabling users to create matte painting with ease and reduce human labor to perform the same task if done manually. Further with an Interactive View the User can interact with the matte painting in real time in an interactive environment such as a Canvas. Overall the User will have an easy to use tool to create matte paintings and an interactive environment to view the matte painting in the with controlled camera movement creating an enhanced viewing experience from a Still Image.

Keywords- Computer Vision, Pattern Recognition, Image Processing.

I. INTRODUCTION

Biological visual systems evolved from a simple unicellular organism that follows light stimuli to humans who recognize and process a wide range of patterns. We utilize the patterns to navigate the world around us. We also represent information through paintings, books and now digital media.

With computers we can capture and store visual information as image and manipulate them further to enhance or create. With computer vision and image processing the creation processes can be automated. Machines can be trained to extract, interpret, and generate outputs.

There are a good amount of recent implementations of such models. One Shot 3D Photography is one of such implementations. It generates a 3D photo with interactive parallax by taking an Image as input. The paper is the inspiration and main reference for this project.

An image is represented as an array of Pixels. A single pixel may not convey the full information required for high-level tasks. By recognizing patterns, computers can distinguish between various objects. Further the color information needs to be processed into values that algorithms can work with. Many libraries and technologies provide the necessary tools and algorithms for image processing.

Let's have a brief overview at using numpy and skimage for image processing. Numpy represents the image in a n-dimensional numpy array and offers the functionality of importing images into the array and vice versa. After loading the Image as an array we can proceed to perform matrix operations to analyze and extract the Information we require

II. METHODOLOGY

The above image is of 16 pixels with a shape of (4, 4, and 3). Each pixel is an array of rgb which corresponds to a color value. [255,255,255] being white and [255, 0, 0] being red. Each pixel is represented as a combination of rgb. We can reduce computational power required by normalizing the values. The process can be equipped the other way to enhance the values too.

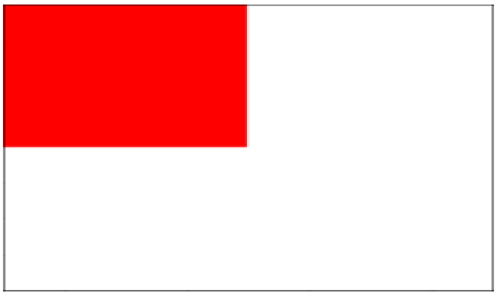

Fig 1. 4X4 RGB Image.

There are many operations we can perform on the Image array to extract the information we require. we are going to briefly go through a handful of methods using skimage which later we adapt for the ML model.

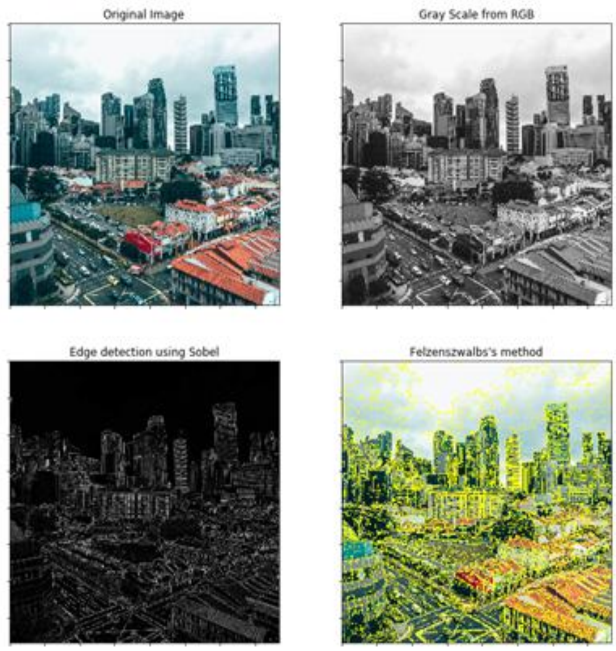

Fig 2. features extraction using skimage.

In the above figure specific filters are applied to extract further information from the pixel values. By converting the rgb Image to gray we have reduced the number values we need to compute. Further matrix Kernels are applied throughout the image such as sobel for edge detection. Further the image is segmented using Felzenszwalb's efficient graph based segmentation. These are some of the examples of the operations we can perform among many others. These are a few methods among many other methods that skimage offers. Some other notable methods Skimage offers to extract the image features include. Binary Texture Classification, Super Pixels These filters we are applying on the image to extract the features can be interpreted as a kernel operating on the Image. The kernel is a 3x3 matrix. The kernel can be applied across the whole image or regions of the image based on the requirement.

To adapt the Image as input for the model. The values need to be converted into a tensor. The tensor is given as input to the model. By performing operations on the given tensor such as Convolution we can extract feature map from the input. Additional hidden layers and activation layers are utilized to fine tune the model to perform a given task. The tensor data will be in the format of (CxWxH) channel for width and height forming a tensor with three dimensions. The model needs additional dimension for class. For background the default class is 0 and the other classes follow as per the coco dataset labels further the mask for instance segmentation is fine tuned for performance.

The models are trained to perform tasks such as classification and segmentation. by adding mode layers the model can be configured to output specific targets as defined by the user. The models are trained to perform specific tasks. The trained models can be combined to produce an aggregated output. The model adapts itself as per the data it intakes and calculates the weights from the operations. For every forward run of the model the model back propagation to adjust the weights.

Once the model is trained with a good amount of data the model learns to recognize the data and perform operations according to its weights and biases. Models need to be in evaluation to halt the computational graph. Where as in training mode the model learns from the operations. In evaluation mode the model is only required to produce the output.

Models can be combined and utilized to perform an aggregated output where underlying models are equipped to perform specific tasks. For this project the models being used are Mask R-CNN with a ResNet-50-FPN backbone for object detection and instance segmentation. The other model that is being used in the project is Monodepth which is based on the pix2pix architecture to estimate the depth of a given image.

III. RESULTS

Fig 3. Instance Segmentation.

By utilizing Mask RCNN we generate masks for the objects detected in the image. In the fig 3 the model has predicted four classes of objects in the image (color separated). The predicted masks are used to extract the objects. if the input image consists of multiple instances of the same class, each object can be isolated to select a specific object for further operations.

The predicted mask will be of tensor with four dimensions. The tensor needs to be converted back to three dimensions and to a numpy array to utilize the mask for Image processing operations.

Fig 4. Depth Estimation.

The depth of the given Image is estimated using Monocular depth estimation. It detects corners and the edges of the objects and provides depth information as brightness of the pixel. In fig4 the depth of the base can be viewed as a gradient. The bottle's position can be estimated relative to the depth of the base gradient.

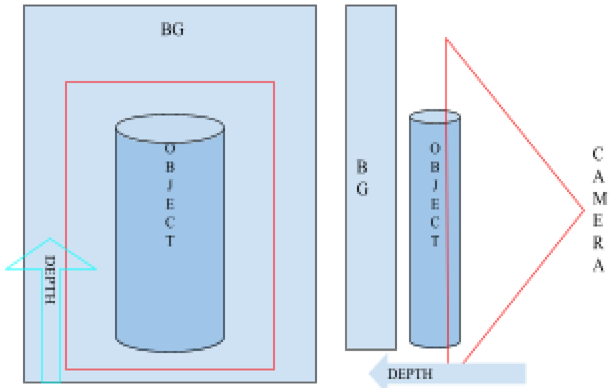

Fig 5. Object Positioning.

Once the objects are separated with their respective masks. we need to fill the missing areas with the local pixel values and textures. if the operation is not done the parallax effect will expose artifacts and the missing areas. With in painting this task can be performed.

After filling in the missing areas the objects which are now separate images can be composited in the canvas or 3-D space with locations relative to their depth. The extracted objects are planes. They can be further enhanced by applying normal and bump maps which can be extracted from the image. For simple objects which can be constructed using simple geometry such as a sports ball the mesh can be replaced with a three dimensional geometric shape.

In the case of a sports ball it can be replaced by a sphere and the texture is applied. For more complex geometry mesh can be applied a displace transform for additional effect. The meshes are called as the distance gets further to add the parallax effect. Parallax effect creates the illusion of depth. In the fig 5 the red rectangle represents the camera view.

The background is scaled to accommodate camera movement. When the scene has multiple objects excluding Background the objects are placed as per their relative position according to the Depth Estimation. The objects also need to be scaled relative to their position. Further the viewing experience of the User can be enhanced by adding premade 3-D Assets for classes such as Trees, Plants etc. Additional effects

Fig 6. **Text Here Your fig Title.**

As seen above in the fig 6 the detected object of a sports ball is rendered as a sphere in the canvas with the texture obtained.

IV. CONCLUSION

By combining Models to automate labor intense tasks such as extracting the subject from an image by generating masks. Acquiring Depth information and compositing the objects in an interactive canvas. The users can speed up the process of creating a Matte painting. Further with access to files generated in the process the users can utilize these files to composite a matte painting using an application of their preference.

By allowing the models to train as the user operates to generate the matte painting. models can adapt to operate the tasks with higher performance. We live in a world rich in visual patterns to which we Humans have adapted to not only recognize but to communicate utilizing the patterns. With the development in computational technologies. We are now able to train intelligent systems that can interpret the world around us and assist us.

Deep Learning Models are already assisting us in identifying abnormal cells in Medical Imagery, search for celestial objects, and other image processing tasks. These Models are also assisting us in Autonomous Vehicles. With these Technologies can aid us take better decisions, reduce human labor, and help us build a better world?

REFERENCES

[1] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

[2] Van der Walt, S. et al., 2014. scikit-image: image processing in Python. PeerJ, 2, p.e453

[3] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[4] Dai, Zihang & Liu, Hanxiao & Le, Quoc & Tan, Mingxing. (2021). CoAtNet: Marrying Convolution and Attention for All Data Sizes.

[5] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[6] VLMo: Unified Vision-Language Pre-Training with Mixture-of-Modality-Experts arXiv:2111.02358 [cs.CV]

[7] skorch: A scikit-learn compatible neural network library that wraps PyTorch, Marian Tietz and Thomas J. Fan and Daniel Nouri and Benjamin Bossan, 2017.

[8] Ren\'{e} Ranftl and Katrin Lasinger and David Hafner and Konrad Schindler and Vladlen Koltun, Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer, TPAMI 2020.

[9] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, Deep Residual Learning for Image Recognition, arXiv 2015.

[10] Lundh, F., 1999. An introduction to tkinter. URL: www.pythonware.com/library/tkinter/introduction/ index. htm.

[11] fig 2 Photo by Lily Banse on Unsplash

[12] fig 3 & 4 Photo by Artem Beliaikin from Pexels