Disparity Map Estimation Using Uncalibrated Stereovision

Ashwin Unnikrishnan, Santosh Vasa
Khoury College of Computer Sciences, Northeastern University
unnikrishnan.a@northeastern.edu, vasa.s@northeastern.edu

Abstract

Stereo vision (stereopsis), which uses images from two cameras to triangulate and estimate distances, is the most prevalent method for depth estimation in computer vision and robotics. The goal is to take advantage of the parallax error. The same scene is captured from two separate perspectives, and depth is calculated using the parallax error. This strategy has been around for over a century and real life usage can be seen in nature as well, predators have their eyes set in such a way that they have maximum overlapping field of view, with which it can estimate distance the prey is at. In the field of robotics, this system is extremely beneficial. Offering them estimated object depths gives them a 3D understanding of the scene. We use this further to this used for 3D reconstruction of the place or obstacle avoidance for an autonomous vehicle. We use two web cameras stuck onto a hardboard as the stereo vision setup.

I. Introduction

Science and technology, particularly computer vision, have advanced rapidly. Computer vision systems, which employ computers to provide vision and simulation, are used in a variety of disciplines in daily life. Single, stereo or many camera systems can execute the vision activity. Using doubled or multi-camera systems, stereo systems can actualize computer vision events. Stereo vision systems are visualization techniques that allow point coordinates to be replicated in three dimensions on images captured by two cameras. Stereo vision systems, most of which are dual and multiple-vision-based. Portable autonomous robotic systems, 3D measurements, object tracking, film industry, augmented reality, and object recognition is just a few of the applications for these vision systems. Many autonomous cars have been developed in order to meet human requirements. Sensors, particularly cameras, are used by autonomous cars to collect the state of the environment. Computer vision will be used to process picture data from the surroundings.

The goal of a stereo vision system is to recover the three-dimensional (3D) position of features from projections in two-dimensional (2D) photographs. The ability to accurately discern the correlation between the left and right images is the main challenge in achieving this goal. Various solutions to this problem have been presented by researchers. A non-convergent stereo imaging system with two fixed cameras separated by a baseline distance and parallel optical axes whose parameters are calculated in the calibration step is a common stereo system. A left and right image can be obtained using this technique. The 3D coordinates of an image point in the scene are discovered by computing the displacement, or disparity, between two corresponding features in the left and right picture. After then, there are three primary processes in the computational stereo paradigm: stereo camera modeling, feature detection, and matching. The most difficult part is determining how the two stereo images coincide. Feature-based and area-based approaches have been used in the majority of cases. Feature-based solutions employ simple, geometric primitives such as line segments and planar patches . Such models are appropriate for simple, well-structured environments consisting of man-made objects. These techniques have generally been more successful overall as the matching process is much faster than the area-based techniques and there are fewer feature points to be considered. The area-based algorithms represent depth at each pixel in the image. These techniques promise more generality.

You can see the calibrated stereo in nature as well. Predators in general use stereo vision to estimate the depth(distance) from the prey. Predators have both the eyes setup in such a way that they have maximum overlapping field of view, which helps in calculating the depth. And unlike predators prey have eyes setup in such a way that they have wider field of view, this helps them in being attentive for attacks from every side.

In this paper, we calculate the disparity map using stereo vision setup where two web cameras are stuck on a hardboard such that the lenses align horizontally.

II. Related Work

There are plenty of practical applications that are accomplished by calibrated stereo vision systems, as evidenced by the following literature.

Huh et al. (2008) recommended using stereo vision sensors to create an obstacle detection system. In a related approach, the beginning point is selected first, and then potential barriers are identified[4].

Bhowmick et al. (2011) created a stereo vision system that

detects people and calculates their distance from the vehicle[5].

Lai et al. (2012) designed a stereo vision system for mobile robots to measure distance information[8].

The stereo vision could be used for lunar landing as well as proposed by Jie Ming, Huang Xianlin in his A Lunar Terrain Reconstruction Method Using Long Base-line Stereo Vision paper. The moon surface was reconstructed to understand and make landing possible and efficient.

III. Background

The first step to achieving stereovision is to calibrate the cameras. Camera calibration allows us to recover the 3 - dimensional data of a scene, lost in the process of perspective projection. To reconstruct any 3D scene from images, we need two parameters:

1. **Extrinsic parameters:** The Position and Rotation of the camera with respect to the world coordinate frame. The Translational and rotational vectors constitute the external parameters of a camera.
2. **Intrinsic Parameters:** The camera maps the 3D world coordinates to the image plane using the perspective projection. The parameters needed for this estimation are internal parameters which constitute focal length, center of optical axis and distortion parameters.

Once we have these parameters, the camera is calibrated for the 3D reconstruction. The figure 3.1 shows the mapping of world coordinates to the image plane.

Fig 3.1 - Perspective projection

These are the equations for perspective projection:

$$x_i = f\frac{x_c}{z_c}, y_i = f\frac{y_c}{z_c},$$

(x_i, y_i) coordinate represents the pixel of the projection of a point (x_c, y_c, z_c) on to an image plane.

Using homogeneous Coordinates (u,v), we can find the intrinsic parameters of a camera:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{f}_x & 0 & \mathrm{o}_x & 0 \\ 0 & \mathrm{f}_y & \mathrm{o}_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x}_c \\ \mathrm{y}_c \\ \mathrm{z}_c \\ 1 \end{bmatrix} \quad (1)$$

The above equation is a linear model for perspective projection where the 3x3 matrix is called the calibration matrix that takes a camera coordinate $(x_c, y_c, z_c, 1)$ and projects it to the image plane $(u, v, 1)$.

To convert the world coordinates into camera coordinates, we use the homogeneous coordinates as shown in the following equation.

$$\begin{bmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} & \mathrm{t}_x \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} & t_y \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x}_w \\ \mathrm{y}_w \\ \mathrm{z}_w \\ 1 \end{bmatrix} \quad (2)$$

The above equation transforms the world coordinates to the camera coordinates and this 4x4 matrix is called the extrinsic camera matrix.The above 2 matrices from equation 1 and 2 can be combined to get one projection matrix, that takes world coordinates to get pixel location in the image.Using an engineered pattern like a chessboard, we get set of world and image coordinates. By substitution, we get a set of system of equations which can be solved to get the projection matrix.

Simple Stereo: We can construct depth maps and estimate depth of an object by projecting the 2D image plane location on to the 3-Dimensions. The following equations project 2D points on to a 3D line.

$$x = \frac{z}{f_x}(u - o_x)$$
$$y = \frac{z}{f_y}(v - o_y)$$
$$z > 0$$

This gives us the line where the 3D location can exist. We cannot find a 3D location from a single camera view using geometry. We can use a stereo setup to find the 3D location using the concept of triangulation.

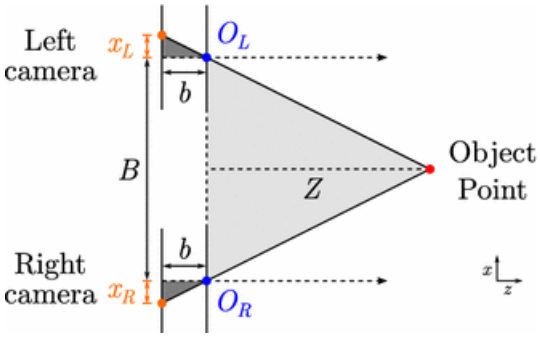

Fig 3.2- Triangulation

In simple stereo, we assume that both the cameras are pre-calibrated where the second camera is displaced horizontally from the first. In order to do depth estimation, we have to find a 3D point given a perspective project of the same.In fig 3.2, let us say the object point is (x,y,z) and the point projected on left camera is (u_l, v_l) and the right is (u_r, v_r). As we know that the second camera(right) is displaced from the left by the baseline b,We use the following perspective projection equation:

left camera,

$$u_l = f_x\frac{x}{z} + o_x, v_l = f_y\frac{y}{z} + o_y$$

right camera,

$$u_r = f_x\frac{x-b}{z} + o_x, v_r = f_y\frac{y}{z} + o_y$$

As there is no change in the Y direction (horizontally displaced right), we do not see a change in the v_r equation. solving for these equations we get,

$$z = \frac{bf_x}{(u_l - u_r)}$$

Here $(u_l - u_r)$ is disparity. We can see that disparity is inversely proportional to the depth Z. Larger the baseline, more accurate the disparity measures are as we can find disparity measures for farther objects. Although, higher baseline causes difficulties in finding the correspondence points (more on this later).

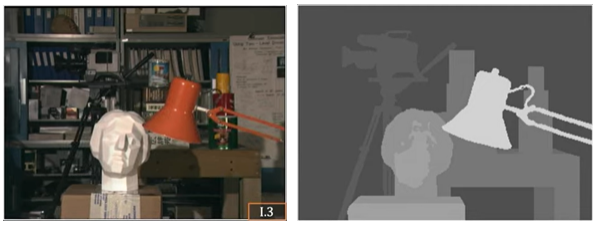

Fig 3.3- Disparity map

In simple stereo, as the cameras are only horizontally displaced, we already know the relative translation and rotation vectors between the two cameras. This makes it easy to do a depth correspondence match between two stereo images. A left pixel location (x_l, y) will only match to the right pixel (x_r, y), This reduces the search space from a complete image to a line in the image. TO find matching points, we can use any similarity metric such as sum of absolute differences, sum of squared distances , normalized co-relation and more.To increase the accuracy and reliability, we can find similarity between windows instead of a pixel location.

Un-calibrated Stereo: In the case of uncalibrated stereo, we will have compute the translational and rotational vectors of second camera with respect to the first camera. This is because the second camera in this case is not just displaced horizontally but also in other axes. The translational and rotational vectors are needed as the translation and rotation of second camera with respect to the first is not known.In this case, we assume to know the intrinsic parameters of the cameras and the cameras should be pointed towards a static scene. Using the intrinsic parameters, we can rectify the image from second image to satisfy the epipolar constraint and find the line of potential matches. We then perform same set of depth correspondence operation on the epipolar lines.

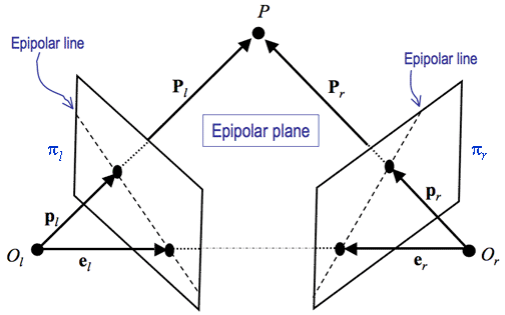

Fig 3.4- Epipolar Geometry

In the above picture, the relation ship between O_l and O_r is not known. To find this, we will compute some reliable correspondence matches. This can be found using any feature detectors such as SIFT, SURF and more. Here is an example of matched SIFT features we computed:

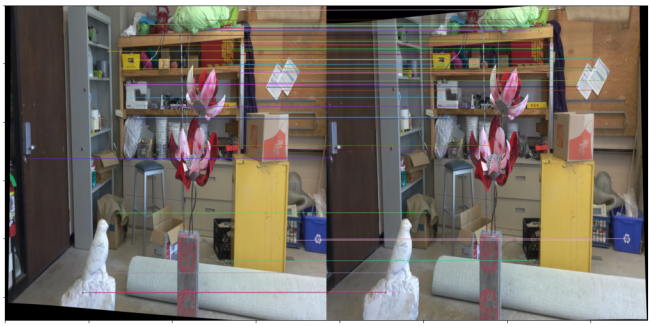

Fig 3.5- SIFT Feature Matches

For calibration purpose, that is to find the relationship between O_l and O_r, we just need 8 reliable points such as the ones shown in the fig 3.5. We can say that the stereo camera is calibrated as we have the relative translational and rotational vectors. The relative orientation and translation between the cameras is completely described by epipolar geometry.

Epipolar Geometry: To relate a 3D position (x_l, y_l, z_l) of scene point with respect to the first camera to its 3D position (x_r, y_r, z_r) with respect to right camera we will need the essential matrix.An essential matrix can be calculated by multiplying the translational and rotational vectors as following.

$$E = TxR$$

$$\begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \tag{3}$$

$$\begin{bmatrix} 0 & -t_z & ty \\ t_z & 0 & -t_x \\ t_y & t_x & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4}$$

It relates the points as shown below,

$$\begin{bmatrix} x_l & y_l & z_l \end{bmatrix} = \tag{5}$$

$$\begin{bmatrix} \mathrm{e}_{11} & \mathrm{e}_{12} & \mathrm{e}_{13} \\ \mathrm{e}_{21} & \mathrm{e}_{22} & \mathrm{e}_{23} \\ \mathrm{e}_{31} & \mathrm{e}_{32} & \mathrm{e}_{33} \end{bmatrix} \begin{bmatrix} \mathrm{x}_r & \mathrm{y}_r & \mathrm{z}_r \end{bmatrix} = 0 \tag{6}$$

Using the inverses of camera matrices for two cameras k_l and k_r, we can then convert the 3D points of left and right cameras into 2D points. By combining the inverse camera matrices with the essential matrix, we get the fundamental matrix as shown below:

$$(u_l, u_l, 1)K_l^{-1T} * E * K_r^{-1}(u_r, v_r, 1) = 0$$

$$F = \begin{bmatrix} \mathrm{f}_{11} & \mathrm{f}_{12} & \mathrm{f}_{13} \\ \mathrm{f}_{21} & \mathrm{f}_{22} & \mathrm{f}_{23} \\ \mathrm{f}_{31} & \mathrm{f}_{32} & \mathrm{f}_{33} \end{bmatrix} \tag{7}$$

We can see that if we can find corresponding 2D points in both the camera views, we can easily solve for the fundamental matrix. The essential matrix can then be identified from the fundamental matrix by multiplying the camera matrices. Using singular value decomposition, we can retrieve translational and rotational vectors from the essential matrix.

Depth Correspondence: After calibrating the stereo setup, the next step is to find the dense correspondences between the left and right images.As seen in the simple stereo setup, due to change in 1 axis that is the horizontal displacement of second camera, the search space of the corresponding match point decreases from an image to a line in the image. The stereo matching in uncalibrated stereo still remains as a 1D search with the help of epipolar geometry.As shown in the fig 3.4, the uncalibrated stereo setup consists of two epipolar lines, one in the right and one in the left. Due to the epipolar constraint, A point in the epipolar line of left plane corresponds to a point in the epipolar line in right plane.This reduces the search space as mentioned before. We can find the epipolar line in the right image plane using the fundamental matrix. By projecting the point in the left image plane, we get a line in the right image plane which represents the epipolar line or right camera. The epipolar constraint equation can be written as the following:

$$(f_{11}u_l + f_{21}v_l + f_{31})u_r + (f_{12}u_l + f_{22}v_l + f_{32})v_r +$$
$$(f_{13}u_l + f_{23}v_l + f_{33})u_r = 0$$
$$a_l u_l + b_l v_r + c_l = 0$$

The depth can then be calculated using the same triangulation concept as discussed earlier.

IV. Experiments and Results

Stereo implementation details: We tried multiple ways to implement Stereo Matching in OpenCV. Using a simple stereo setup data such as the following,

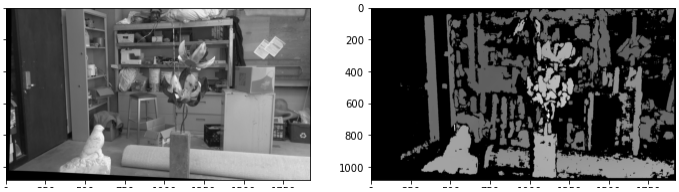

Fig 4.1 - Disparity map using simple stereo

Using StereoBM_Create from OpenCV with numDisparities = 176 and blockSize = 25, we were able to compute the stereo between left and right images. The Fig 4.1 shows a clear depth map of the scene.

Uncalibrated Stereo: For uncalibrated stereo, we have prepared a setup using two webcams. Using calibrateCamera function, we retrieved the intrinsic parameters of both the cameras. Using stereoCalibrate function, we were able to compute the essential matrix, fundamental matrix, decomposed translational and rotational vectors. As the setup is uncalibrated, we rectified the input images to meet the epipolar constraint using stereoRectify function. Out setup involved two wide angle cameras which have high distortion. We were able to remove the distortion in both the cameras using undistort function from OpenCV as shown below.


Fig 4.2 - Removing distortion a) distorted input image b) shows image after undistortion c) cropping the region of interest

3D Output: After stereo rectification and undistortion, we can us the resultant output to generate 3D images. This can be done by taking the blue channel from left image and red channel from right image. The output will be a 3D image as shown below.

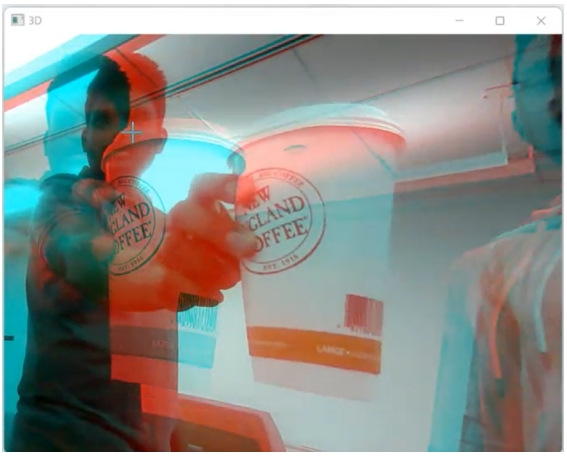

Fig 4.3 - 3D output

By applying stereo compute function from OpenCV on rectified left and rectified right images, we could retrieve a disparity map. To understand the best hyperparameters such as numDisparities, minDisparities, blocksize and more, we used GUI trackbars. Once the trackbars are in good position, we were able to retrieve a disparity map that's in usable form.

SIFT output: Instead of using the stereoCalibrate function to retrieve the essential and fundamental matrices, we also used the sift features and computed the same manually. SIFT feature detector matches can be computed using the OpenCV functions sift.detectAndCompute() and draw their keypoints using drawKeyPoints(). The following is an output of the same.

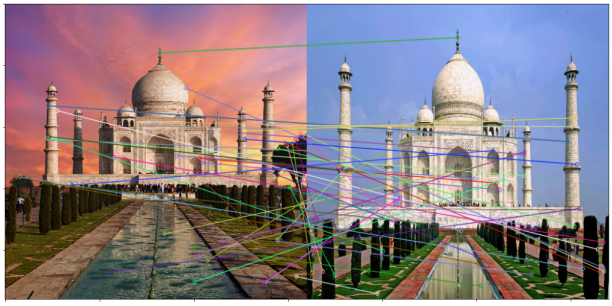

Fig 4.4 - SIFT output

Using flann and KNN, we generated good matches between the images.The following image selects a few best matches as shown. This method works better with images that does not have symmetry.

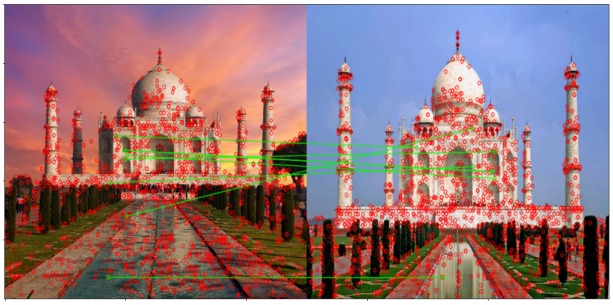

Fig 4.4 - Flann output

We can retrieve the fundamental matrix by applyting the findFundamentalMat() function on the best matches given by Flann. We can visualize the epipolar lines mapped from left best matches to right epipolar lines as shown below. This is done using the function computeCorrespondEpilines().

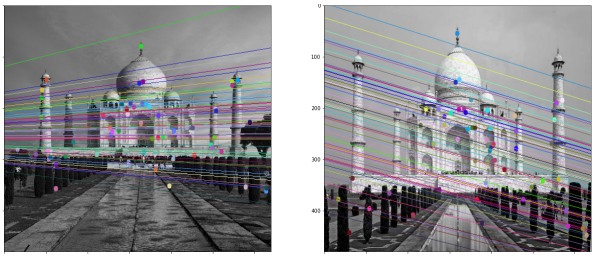

Fig 4.5 - Epipolar lines output

Using the StereoRectifyUncalibrated function, we can rectify the input images as shown below. This makes the corresponding rows as the search space.

Fig 4.5 - Stereo Rectification output

Disparity Map Output

Fig 4.5 - Disparity Map output

Using the function StereoSGBM_create(), we can then compute the depth map between the rectified images. In the above output, we can see the outline of my head and hands. Although the there is a lot of noise, we can find better key point matching algorithms to find more reliable matches and more reliable fundamental matrices.

V. Conclusion

In this research, we show how to estimate depth in a structured environment, by using two web camera that is aligned and stuck onto a hardboard to look like two eyes. The approach combines fish-eye lens picture distortion correction, a vanishing point-based segmentation procedure that allows the system to extract only the most significant line segments, and a correspondence algorithm based on a recursive hypothesis formulation and verification procedure. On real-time video streaming, the system has been effectively established.

VI. Future Work

Once the disparity is calculated we can use it to find the depth of all the points in the image. This can be used for 3D reconstruction. The 3D model can be used for a wide variety of applications ranging from identifying the lunar surface for proper landing to locating catheter in the heart.Reconstruction allows us to gain insight into qualitative features of the object.

For obstacle avoidance, we can add object detection on top of it to find the respective objects and create a 3D map, that can be later used to track the objects.

VII. References

[1] Shishir Shah and J. K. Aggamal . Depth Estimation Using Stereo Fish-Eye Lenses.

[2] Anwar Hasni Abu Hasan, Rostam Affendi Hamzah, Mohd Haffiz Johar. Region of Interest in Disparity Mapping for Navigation of Stereo Vision Autonomous Guided Vehicle .

[3] Anggi Gustiningsih Hapsani, Dahnial Syauqy, Fitri Utaminingrum, Putra Pandu Adikara, Sigit Adinugroho , Onward Movement Detection and Distance Estimation of Object Using Disparity Map on Stereo Vision.

[4]K. Huh, J. Park, J. Hwang, and D. Hong, ”A stereo vision-based obstacle detection system in vehicles,” Optics and Lasers in Engineering, vol. 46, pp. 168-178, 2008.

[5] B. Bhowmick, S. Bhadra, and A. Sinharay, ”Stereo vision based pedestrians detection and distance measurement for automotive application,” in 2011 Second International Conference on Intelligent Systems, Modelling and Simulation, 2011, pp. 25-29.

[6] G. Saygili, L. Van Der Maaten, and E. A. Hendriks, ”Adaptive stereo similarity fusion using confidence measures,” Computer Vision and Image Understanding, vol. 135, pp. 95-108, 2015.

[7] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. (2017). Stereo Vision Based Forward Obstacle Detection.

[8] X.-b. Lai, H.-s. Wang, and Y.-h. Xu, ”A real-time range finding system with binocular stereo vision,” International Journal of Advanced Robotic Systems, vol. 9, p. 26, 2012.

[9] Jie Ming, Huang Xianlin, A Lunar Terrain Reconstruction Method Using Long Base-line Stereo Vision

[10] Pytorch Team, “FCN”,https://pytorch.org/hub/pytorch-vision-fcn-resnet101/

[11] Paul-Louis Pröve, “An Introduction to different Types of Convolutions in Deep Learning”, towardsdatasscience (blog) , 22nd July, 2017, https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

[12] Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár and C. Lawrence Zitnick. “Microsoft COCO: Common Objects in Context.” ECCV (2014).

[13] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep residual learning for image recognition.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[14] Zhang, Yin Jin, Rong Zhou, Zhi-Hua. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics. 1. 43-52. 10.1007/s13042-010-0001-0.