

Blueprints Obtention by means of Using Digital Image Processing Algorithms

Fernando Martínez Santa^{#1}, Holman Montiel^{#2}, Fredy Martínez^{#3}

[#]Technology Faculty, Universidad Distrital, 68d Bis A south st. 49F-70, Bogotá, Colombia

¹fmartinezs@udistrital.edu.co

²hmontiela@udistrital.edu.co

³fhmartinezs@udistrital.edu.co

Abstract—This paper presents the proposal of an algorithm to obtain images similar to design plans or/and blueprints, starting from color images or real photographs of the objects. The proposed approach generates both, black & white plans and blueprint-like images, applying image processing operations like gray-scale conversion, image thresholding, image convolution, fast Fourier transformation, histogram stretching, among others. This process is divided in steps and grouped in a common pipeline to obtain the main edges of the image and to different branches to generate black & white plans and blueprints. The proposed process works over images with resolution higher than 900x600 pixels and with flat backgrounds, to obtain reliable results.

Keyword - Blueprints, Photographs, Image processing, Thresholding, Sobel edge detection, Image histogram, FFT

I. INTRODUCTION

Paper plans are usually used as tool to represent the main or starting idea of any kind of project, such as architectural, electrical, mechanical, etc. Frequently, design plans are called blueprints, referring to the old photographic printing process named *cyanotype*, with which cyan-blue copies were obtained from the original plan or paper. Blueprints are used not only as starting step in the process of manufacturing of a specific device or product, but also to recreate it. That is the case of 3D modelling process, for which, plans or blueprints of the devices are used as raw material to model it in a 3-dimensional environment, even some algorithms based on that concept have been proposed [1]–[3], as well as some applications [4].

In areas like architecture and computer animation, the use of objects such as: furniture, lamps, kitchen and desktop objects, bicycles, motorbikes and cars, among others, is really often; that is why 3D modelers are constantly creating those ones, which most of times have to be specific, for instance a specific brand and model of a car. Due to that, 3D artists used to base their projects on design plans or blueprints of the original objects to make a 3D model as similar to the original as possible. Obtaining design blueprints is really difficult even more when the object or product has been recently released. On the other hand, nowadays, photographs are very easy to be obtained or taken directly from the object, that is the reason why this paper proposes a blueprint obtaining approach starting from photographs or images of the objects, by means of using some digital image processing algorithms.

Digital image processing algorithms have been widely used to solve a lot of different problems such artificial vision [5], [6], robotics [7], satellite images analysis [8], among others. Some image processing techniques have been used specifically to scan, analyse and enhance plans [9]–[11] and to obtain them, for instance extracting plans from 3D point clouds [12], drawing blueprints using hardware acceleration [13], generation of blueprints for smart homes [14], or procedural generation of blueprints [15].

Generally, the main process to obtain a plan or blueprint from an image of photo is edge detection, for which there are different algorithms, most of them use bi-dimensional convolution operations. One of the most common edge detection algorithm is *Sobel* [16], [17], for which some improvements have been formulated [18], [19] combining it with some other ones. *Sobel* has been widely used for pattern recognition like license plate recognition [20], among others. Next sections show the pipeline (flow steps of the processed images) proposed approach, which uses the Sobel algorithm along with some other digital image processing ones, and produces both, images similar to the old-style blueprints and black-white plans, starting from images or photos.

II. METHODOLOGY

The blueprints and black-white plans generation process proposed here, has almost the same pipeline or process steps, and only change in the last ones, it means the steps before the final visualization. The common processing steps are: RGB to gray-scale image conversion, edge detection, band pass filter using FFT (fast Fourier transformation) and finally the zero-correction based on image histogram analysis. All this pipeline is shown in Fig. 1.

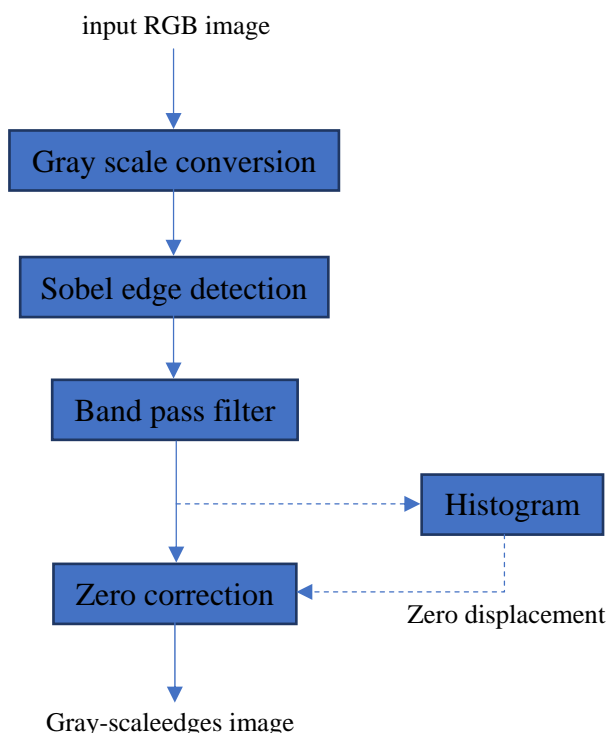


Fig. 1. Gray-scale edges image obtention pipeline

A. Gray-scale conversion

In this step the RGB input image is turned into a gray-scale one, carrying out a quite simple process: the RGB input image with $X*Y*3$ size is split in each of its color channels of $X*Y$ size, after each pixel is averaged with the same in all channels obtaining a gray-scale image which corresponds with the original one.

B. Sobel edge detection

Starting from the gray-scale image (just named I in this paper), and edge detection algorithm is used to obtain the basic edge lines of the blueprint. Most of the image edge detection algorithms are based on the bi-dimensional discrete convolution between the original image and a small convolution matrix (equations 1 and 2).

$$I_c(x, y) = I(x, y) * M_c(x, y) \quad (1)$$

$$I_c(x, y) = \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} M_c(i, j) I(x - i, y - j) \quad (2)$$

Specifically, *Sobel* algorithm uses two convolution matrices, one by horizontal edges and another one by the vertical ones. Equations (3) and (4) show those matrices, with which the images I_h and I_v are obtained after applying the convolution operation described in the equation (2), finally both of them are averaged obtaining the edges image I_e . An example pair of gray-scale image and resulting edges image is shown in Fig. 2.

$$M_h(x, y) = \begin{Bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Bmatrix} \quad (3)$$

$$M_v(x, y) = \begin{Bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{Bmatrix} \quad (4)$$

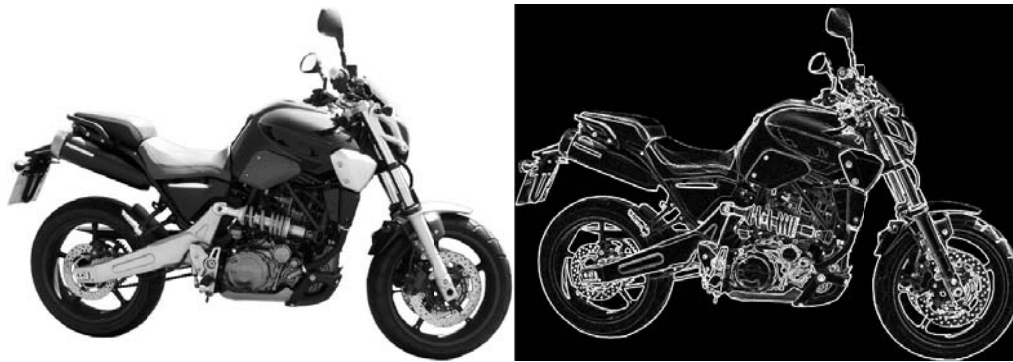


Fig. 2. Edge detection using *Sobel* algorithm. Gray-scale image (left), edge resultant image (right)

C. Band pass filter

When the *Sobel* operation is applied to a real photo, not only generates the main edges of the shape but a lot of different unnecessary sub-edges. In order to reduce this kind of noise, a band pass filter based on bi-dimensional FFT is applied to the edge image. The *Fourier* transformation turns the image into the frequency domain, where some operations are easier to do, in this case it is desired that just the lines of certain thickness keep on the image. The main lines of the edge image have from 3 to 10-pixel thickness due to the 3x3 convolution mask used by the *Sobel* algorithm and the original resolution (and average of 1000x800 pixels), then the proposed band pass filter allows to pass image frequencies higher than 3 and lower than 10 pixels, resulting a filter image as shown in Fig. 3.

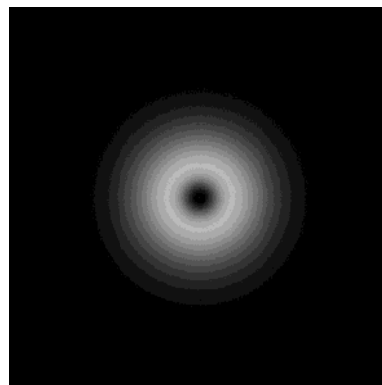


Fig. 3. Lines band pass filter image (let pass lines from 3 to 10 pixels)

To achieve the filtering process, it is necessary to follow these steps: the image is first turned into frequency domain, after the resulting image is multiply by the filter image (Fig. 3) and finally an inverse FFT is applied to obtain the resultant image. Fig. 4 shows the images before and after applying the proposed band pass filter by means of FFT operation.

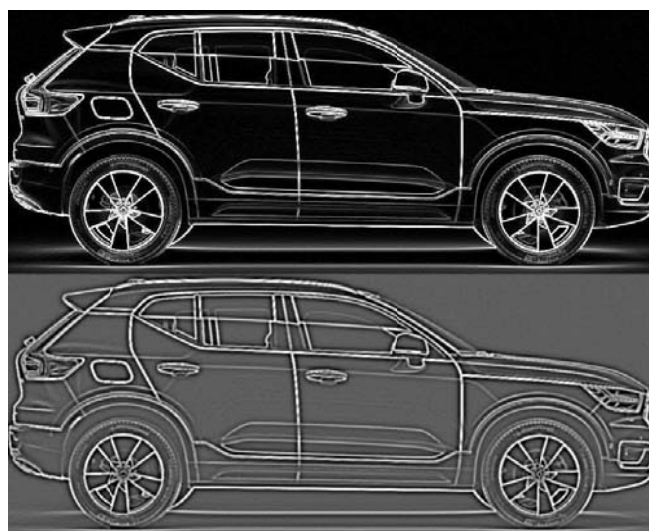


Fig. 4. Band pass filter using FFT. Edge image (up), filtered image (down)

D. Zero correction

The resultant image after applying the FFT band pass filter always look with a gray background (it has a zero displacement), this effect is produced to apply the inverse FFT. Due to the goal of this paper is to obtain a plan or blueprint it is necessary to obtain for this step, a black background. To correct this zero-displacement error, it is imperative to calculate it, this is able to be done through an image histogram. The histograms obtained from that kind of images are similar (see Fig. 5 left), they have only one big “peak” and a small “tableland” just before it. This tableland represents the gray background to be eliminated, then a subtraction operation is performed between each pixel of the image and a constant value K , which is defined as the maximum peak value M_p of the image histogram reduced in 10%, this point is the average one where ends the tableland and starts the peak in around 10 image histograms analyzed. Fig. 5 shows the histograms before and after applying the zero correction, and Fig. 6 shows the images before and after applying it.

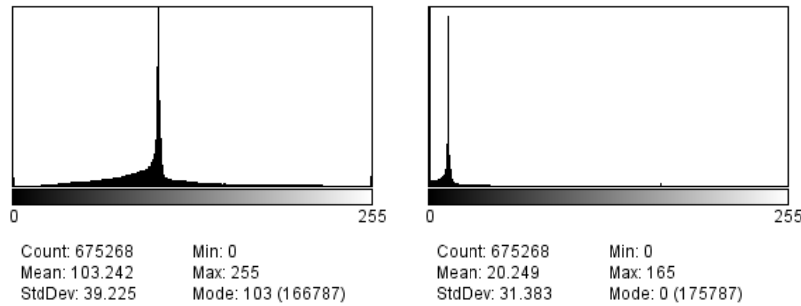


Fig. 5. Histogram of the filtered image. Before (left), and after doing the zero correction (right)

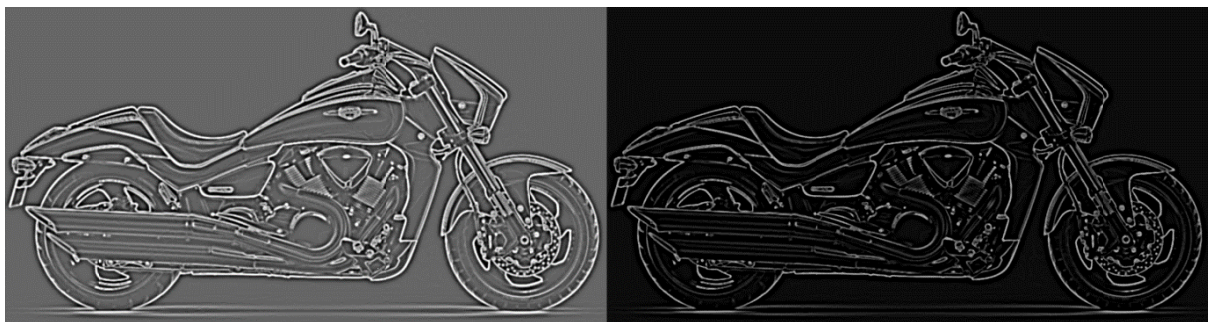


Fig. 6. Zero correction process based on histogram analysis. Before (left), and after doing it (right)

III. FINAL VISUALIZATION

The proposed approach automatically obtains both, black & white plans and blueprints starting from and RGB photograph or image. Fig. 7 shows an example of an original and the two resultant images.

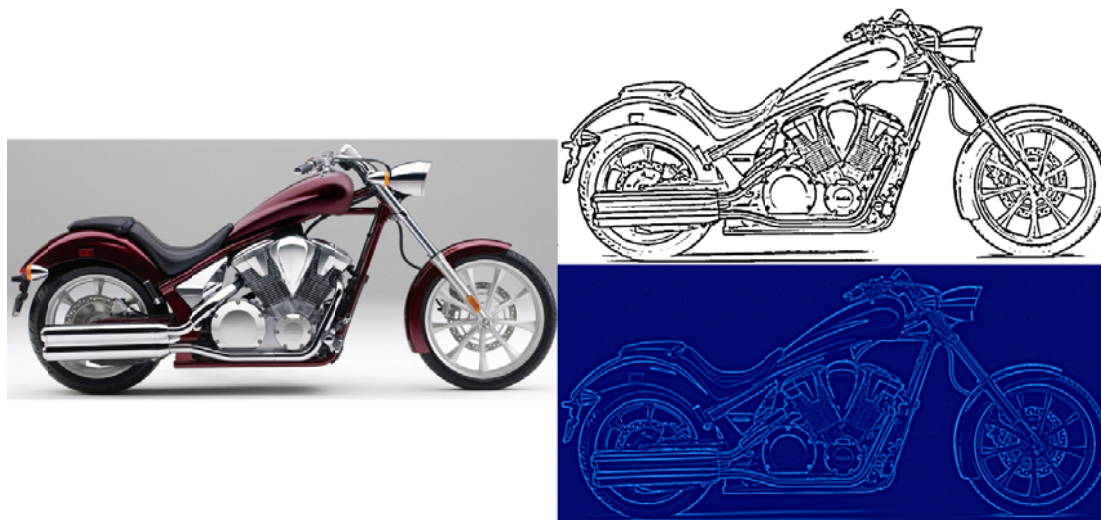


Fig. 7. Results of the proposed approach. Original image (left), black & white plan (right-up) and blueprint (right-down)

For doing this final visualization, two different pipelines were proposed, one for black & white plans and another one for blueprints, this section describes detailedly each one.

A. Black & white plans

The black & white plans are obtained applying a simple thresholding process to the edge filtered image, where the used threshold is selected using the same histogram analysis described in the zero-correction section, as the M_p increased in 80% (far from the main peak). After, the resultant binary image is inverted in order to obtain a white background image. Fig. 8 shows the image obtained after applying the thresholding and inversion process.

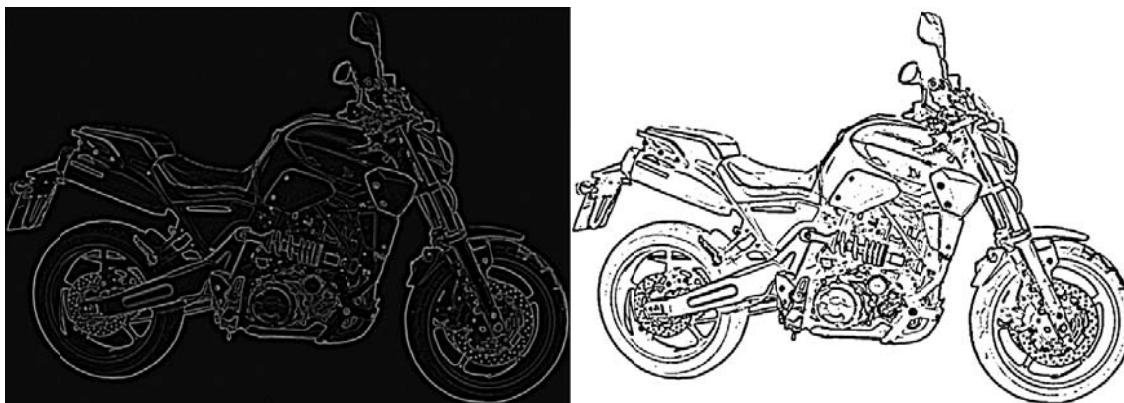


Fig. 8. Black & white plan obtention. Before (left), and after applying the thresholding process (right)

B. Blueprints

A blueprint refers in general to a design plan, but the word comes from the old photographic printing and copying process also named *cyanotype*, which produced plans of blue and cyan color. In order to obtain this blue effect, it is necessary to turn the image back into RGB color one, copying the gray-scale image (the edge filtered) three times, one by color channel. After, each color channel has to be modified to obtain the desired color. Due to the desired image has blue background and cyan lines, the first step is increase the zero level to the blue channel to turn the background into dark blue. This zero level is increased in a constant K , corresponding to the same value obtained in the zero-correction section (analyzing the image histogram). Likewise, the red channel is decreased using the same constant K , with the purpose of obtaining the cyan color for the lines, due to cyan color is obtained with high values in the green and blue channels and low values in the red one. After that, the contrast of the green channel is enhanced, looking for the lines to be highlighted without affecting the cyan and blue color already obtained, this contrast enhancement is done by using a histogram stretching. Finally, all the channels are merged in a RGB image. All the process proposed to obtain blueprint-like images is described as shown in the pipeline of Fig. 9 and Fig. 10 shows an example of using it.

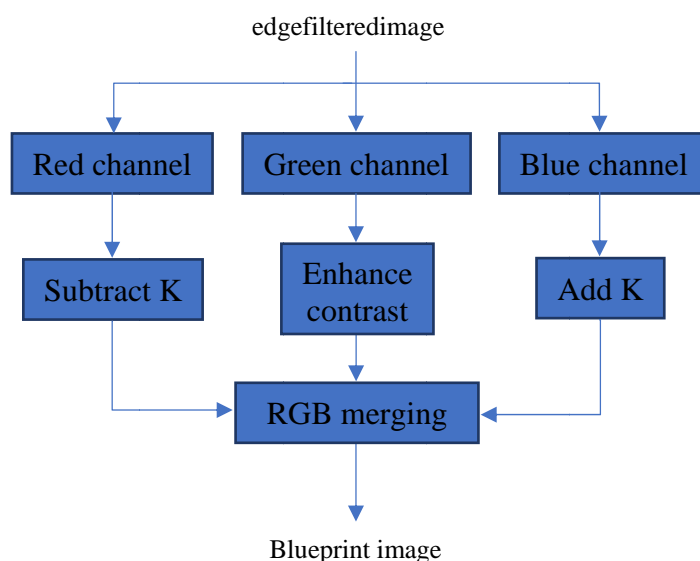


Fig. 9. Blueprint image obtention pipeline

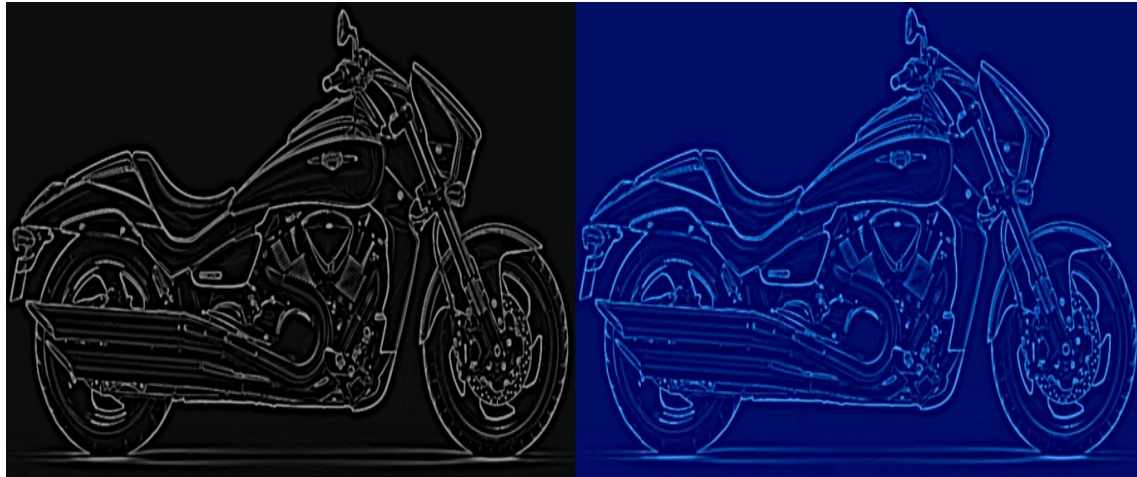


Fig. 10. Blueprint obtention. Before (left), and after applying the RGB merging (right)

IV. CONCLUSION

The proposed automatic algorithm is able to obtain B&W plans and Blueprint-like images from photographs applying some image processing operations. This approach is limited to images with a flat background, in order to obtain clean resultant images. In case of using any kind of backgrounds it is necessary to apply an additional background elimination algorithm, which will be developed for a future work.

The resolution of the input image is very important for obtaining reliable results. After applying the proposed approach over 30 images with different resolution, better results were obtained for the ones with a resolution equal or higher than 900x600 pixels. Low resolution images do not produce good results, due to the *Sobel* operation uses a fixed-size matrix kernel, which generates edges with almost the same thickness producing lines and edges too thick in low resolution images.

When real photographs are used, it is necessary to be aware of the lighting on the scene, specially outdoors, because shines and shadows on the objects can produce artifact edge lines (as shown in the motorbike tank in Fig. 2), which reduce the reliability of the resultant obtained image plan or blueprint.

ACKNOWLEDGMENT

This work was supported by Universidad Distrital Francisco José de Caldas, specifically by the Technology Faculty. The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank the research group ARMOS for the simulations and tests done. All the used photos, images and tools were taken from: www.baltana.com, www.freeimages.com and imagej.nih.gov/ij.

REFERENCES

- [1] T. Thormählen and H.-P. Seidel, "3D-modeling by ortho-image generation from image sequences," in *ACM Transactions on Graphics (TOG)*, 2008, vol. 27, no. 3, p. 86.
- [2] A. Rivers, F. Durand, and T. Igarashi, *3D modeling with silhouettes*, vol. 29, no. 4. ACM, 2010.
- [3] S. Tano, T. Kodera, T. Nakashima, I. Kawano, K. Nakanishi, G. Hamagishi, M. Inoue, A. Watanabe, T. Okamoto, and K. Kawagoe, "Godzilla: Seamless 2D and 3D sketch environment for reflective and creative design work," in *INTERACT'03*, 2003, pp. 311–318.
- [4] D. Li, Y. Zhu, Z. Du, and T. Hong, "Virtual tang-style timber-frame building complex," in *Advances in Artificial Reality and Tele-Existence*, Springer, 2006, pp. 880–888.
- [5] J. A. D. Cepeda and M. R. P. Pereira, "Identificación de formas geométricas y colores mediante procesamiento y reconocimiento de imágenes," *Tekhnê*, vol. 2, no. 1, pp. 9–17, 2004.
- [6] Ó. A. B. García, C. A. M. González, and E. J. T. Bello, "Aplicación de la transformada Wavelet para el reconocimiento de formas en visión artificial," *Tekhnê*, vol. 6, no. 1, pp. 3–8, 2009.
- [7] F. Martínez Santa, S. Orjuela Rivera, and M. Arbulú Saavedra, "Global navigation approach for assistant robot," *Tecnura*, vol. 21, no. 51, pp. 105–117, 2017.
- [8] E. Gómez Vargas, N. Obregón Neira, and D. F. Rocha Arango, "Métodos de segmentación de nubes en imágenes satelitales," *Tecnura*, vol. 17, no. 36, pp. 96–110, 2013.
- [9] Y. Shao, Z. Qi, and S. Yao, "Improved algorithm for scanned blueprint image preprocessing and enhancing," in *Image Matching and Analysis*, 2001, vol. 4552, pp. 28–34.
- [10] S. Ahmed, M. Liwicki, M. Weber, and A. Dengel, "Improved automatic analysis of architectural floor plans," in *Document Analysis and Recognition (ICDAR)*, 2011 International Conference on, 2011, pp. 864–869.
- [11] M. María and M. A. López-Álvarez, "Removing Background Color from Blueprints," in *Color and Imaging Conference*, 2011, vol. 2011, no. 1, pp. 48–51.
- [12] B. Okorn, X. Xiong, B. Akinci, and D. Huber, "Toward automated modeling of floor plans," in *Proceedings of the symposium on 3D data processing, visualization and transmission*, 2010, vol. 2.
- [13] M. Nienhaus and J. Döllner, "Blueprints: illustrating architecture and technical parts using hardware-accelerated non-photorealistic rendering," in *Proceedings of Graphics Interface 2004*, 2004, pp. 49–56.
- [14] J. Lu and K. Whitehouse, "Smart blueprints: automatically generated maps of homes and the devices within them," in *International Conference on Pervasive Computing*, 2012, pp. 125–142.

- [15] D. Sanchez, J. M. Solá-Sloan, and E. Lozano-Inca, "Procedural generation of building blueprints for real-time applications," in Proceedings of the 2010 Spring Simulation Multiconference, 2010, p. 230.
- [16] S. Gupta and S. G. Mazumdar, "Sobel edge detection algorithm," *Int. J. Comput. Sci. Manag. Res.*, vol. 2, no. 2, pp. 1578–1583, 2013.
- [17] O. R. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," in Proceedings of Informing Science & IT Education Conference (InSITE), 2009, vol. 40, pp. 97–107.
- [18] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, 2010, vol. 5, pp. 67–71.
- [19] Z. Jin-Yu, C. Yan, and H. Xian-Xiang, "Edge detection of images based on improved Sobel operator and genetic algorithms," in Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on, 2009, pp. 31–35.
- [20] P. K. Suri, "Vehicle number plate detection using sobel edge detection technique," 2010.