# Color Point Pair Feature Light

Luis Ronald Istaña Chipana and Manuel Eduardo Loaiza Fernández



October 10, 2021

# Color Point Pair Feature Light

Luis Ronald Istaña Chipana[1,2][0000−0003−3470−2759] and Manuel Eduardo Loaiza Fernández[2][0000−0003−0555−9666]

[1] Universidad Nacional San Agustín, Arequipa, Perú http://www.unsa.edu.pe
listana@unsa.edu.pe
[2] Department of Computer Science, Universidad Católica San Pablo, Arequipa, Perú
http://www.ucsp.edu.pe {luis.istana, meloaiza}@ucsp.edu.pe

**Abstract.** Object recognition in the field of computer vision is a constant challenge to achieve better precision in less time. In this research, is proposed a new 3D descriptor, to work with depth cameras called CPPFL, based on the PPF descriptor from [1]. This proposed descriptor takes advantage of color information and groups it more effectively and lightly than the CPPF descriptor from [2], which uses the color information too. Also, it is proposed as an alternative descriptor called CPPFL+, which differs in the construction taking advantage of the same concept of grouping colors, so it gains a "plus" in speed. This change makes the descriptor more efficient compared to PPF and CPPF descriptors. Optimizing the object recognition process [3], it can reach a rate of ten frames per second or more depending on the size of the object. The proposed descriptor is more tolerant to illuminations changes since the hue of a color is more relevant than the other components.

**Keywords:** 3D descriptor · object recognition · RGB-D cameras.

## 1 Introduction

In the area of computer vision, a basic task is the recognition of objects in different scenarios with the aim of understanding that is in the environment. With both information, it can be explored with different applications in the fields of medicine, mechanics, safety, etc. [4]. Taking advantage of the recent technology described as RGB-D cameras, which provide more information than an ordinary camera, it opens up a world of possibilities for the development of new techniques of object recognition in a 3D scenario.

Currently, there are techniques to apply in different cases to the recognition of 3D objects. We can briefly mention 2D descriptors based on image recognition techniques. Which extends the recognition of RGB images to RGB-D images, or add an extra value to the image matrix. We have "Speed up Robust Features" (SURF) [5], "Pyramid histogram of oriented gradient" (PHOG)[6]. These techniques can be easily adapted to using learning networks as applied in unsupervised learning [7] or deep learning implementation[8].

Another approach to solving the problem is 3D descriptors, which have as input to clouds of 3D points or meshes. We have as an example the implementation of "Fast Point Feature Histograms" (FPFH) [9], "Signature of Histograms of OrienTations" (SHOT) [10], or Viewpoint-oriented Color Shape Histogram (VCSH) [11]. But, a technique that stands out among this type of descriptor is the Point Pair Features (PPF) presented by B. Dros et al. [1], which shows great potential due to two factors, the first factor is the relationships between 3D points which is a form of interpretation of the surface. To take advantage of existing color information, a variant called Color Point Pair Features (CPPF) was proposed by [2], this extends the original descriptor by increasing the color information to create a new descriptor.

## 2 3D Descriptors

### 2.1 Point Pair Feature (PPF) Descriptor

A type of 3D descriptor is the PPF descriptor presented by [1], which is based on the geometric relationships of distances and angles that have the pairs of 3D points that are part of an object. While in the recognition phase it uses the pairs relative to each evaluated point belonging to the stage where the recognition of an object is desired. It describes in detail the data that this descriptor uses, the properties that it has, the processes that this method uses to achieve object recognition, and then a color-based variant. The set of features have been taken from the geometric relationships that have the points of an object, such as distance or existing angles. With two 3D points, m1 and m2, 4 different values are obtained, a Euclidean distance and 3 different angles as can be seen in Fig. 1. All of these values will be used for the creation of the CPPFL descriptor and are very important because they will be used to calculate the position and orientation of the object sought.

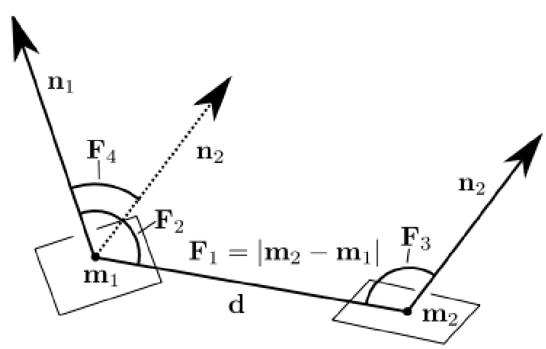



Fig. 1: Point Pair Feature (PPF) Descriptor Components by [1].

### 2.2 Color Point Pair Feature (CPPF) Descriptor

As an extension of the PPF method, we have the CPPF descriptor[2]. This 3D descriptor adds 6 values, using 3 values per color, to represent the color information corresponding to each pair of 3D points, extending the vector from 4 to 10 values. We see the descriptor in [3] CPPF, which it is created from the values of two points taking into account its positions $p_i$ and $p_j$, its normal $n_i$ and $n_j$, finally, its colors $c_i$ and $c_j$. This result in the equation 1. This allows pairs of points with similar color values to be compared in addition to the original information proposed by the PPF method.

$$F_{CPPF} = CPPF(p_i, p_j, n_i, n_j, c_i, c_j) \tag{1}$$

## 3 CPPFL proposed descriptor

The Color Point Pair Feature Light (CPPFL) descriptor is based on the PPF descriptor and adds the color components. Unlike the CPPF descriptor, which also adds the color value, the proposed descriptor encapsulates the color value efficiently for use in object recognition. This difference improves recognition times as well as accuracy. first, it covers the creation of this descriptor and then the results obtained by comparing it with its predecessors.

### 3.1 Color Point Pair Feature (PPF) Descriptor

As mentioned above, our descriptor will use the PPF descriptor as a base, the next step is to see the transformation of a color input to a representative and divisor value. That is a function that receives all 3 parameters of the RGB color space and returns a single one as in equation 2.

$$F(R, G, B) = C_1 \tag{2}$$

The result of this transformation will be used directly as the key of the hash table without further calculation or operations on it. The first step is to split the RGB color space using the color hue used by HSV and HSL. The R, G, and B values correspond to the components of the RGB color space. While the MAX and MIN values are equivalent to the largest and smallest of these components respectively. Equation 3 converts an input in RGB format to an identifier of a primary or secondary color.

$$C = \begin{cases} floor(\frac{G-B}{MAX-MIN} + 5.5) \mod 6 & \text{si } MAX = R \\ floor(\frac{B-R}{MAX-MIN} + 1.5) & \text{si } MAX = G \\ floor(\frac{R-G}{MAX-MIN} + 3.5) & \text{si } MAX = B \end{cases} \tag{3}$$

Applying 3 to the RGB color space yields a division as seen in Fig. 2. A uniform division can be observed for the 6 main and secondary colors. At this point, there is an ambiguity regarding the area that belongs to the color white, and in counterpart to the color black.

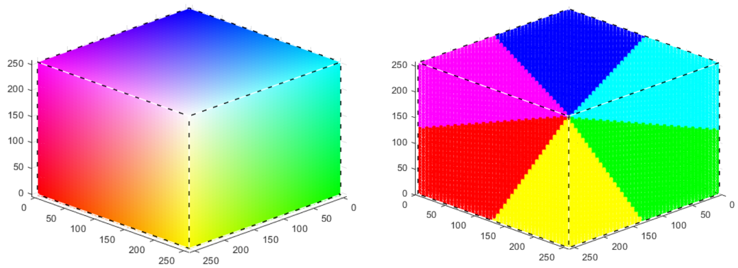
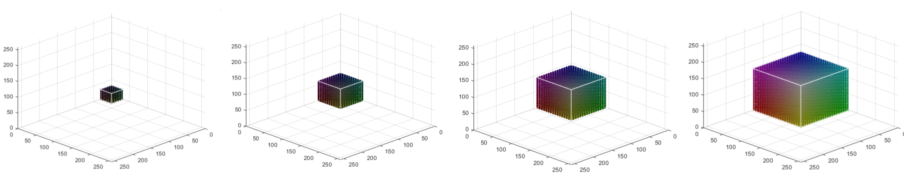


Fig. 2: Division of the RGB color space into color groups according to the proposal.

To resolve this ambiguity, an additional division can be performed taking into account monochromatic colors. Consequently, three values are defined: $(\alpha, \beta, \gamma)$ that will serve to limit monochromatic color spaces. Depending on the value, they will have different sizes as can be seen in Fig. 3. Joining all, we get the next rules:

1. If the largest value is less than $\alpha$, the color is very dark and can be considered black with a code equal to 6.
2. If the smallest value is greater than $255 - \beta$, the color is very clear and can be considered white with a code equal to 7.
3. If the difference of these 2 values is less than $\gamma$, there is neutrality in terms of color and can be considered gray with a code equal to 8.
4. If not in the above cases, the corresponding code is calculated using equation 3.



(a) Black space with $\alpha = 32$ (b) Black space with $\alpha = 64$ (c) Black space with $\alpha = 96$ (d) Black space with $\alpha = 128$

Fig. 3: Space reserved for the color black with different values of $\alpha$.

To summarize, the function that returns an identifier or code for a color in RGB format or equation 2 will return a value depending on the most representative color to the input. All the keys that are obtained are integers, which can be used as part of the key in a hash table. The colors recognized with their corresponding code can be seen in Table 1.

| principal color | key | secondary color | key | monochromatic color | key |
|---|---|---|---|---|---|
| Green | 1 | Yellow | 0 | Black | 6 |
| Blue | 3 | Magenta | 4 | Gray | 7 |
| Red | 5 | Cyan | 2 | White | 8 |

Table 1: Colors and their corresponding key for use in a hash table.

This value will be the new representative in the descriptor structure, which can be used directly as the key of the hash table. The value of the unknown $\alpha$ must be a low number, because achromatic colors are common, but do not have a wide range. Being a lighter version in terms of colors and even starting from the PPF descriptor, it will be called CPPFL.

### 3.2 Creating the CPPFL+ variant

Based on the CPPFL descriptor, and its strategy of dividing the color space into 9 different groups, there is also the possibility of increasing the speed by taking advantage of this information. If each 3D point has a color, these can be grouped according to the proposed division. If two points end up grouped under the same color, they usually belong to the same space and do not provide much descriptive information. On the other hand, if two points differ in color, they generally have a prudent distance between them and also tend to have more discriminating information since it is very likely that they belong to different surfaces. This idea extends to the creation of a new descriptor, which will be a subset of the CPPFL descriptor. This descriptor will have the name CPPFL+, which follows the same procedures with the difference that points of the same color are not used for its creation. With this, the number of points evaluated is much lower but maintains a similar descriptive capacity.

## 4 Test with database captured with a Kinect device

To test the effectiveness of our proposal to the state of the art. Testing has been done on the database provided by [12] which consists of 17 scenes with 27 models divided into 6 classes. In this stage, we will have two phases, one for learning for the creation of the 3D descriptor, and another for the recognition of the object using the 3D descriptor. Both phases are from the original PPF descriptor, the process is adapted to work with color information. During the learning phase, only the times are measured.

### 4.1 Learning phase using the proposed 3D descriptors

To test the creation time of the descriptors using the 3D models in the Kinect database, the models can be seen in Fig. 4. We first create the descriptor for each model using the following descriptors: PPF, CPPF, CPPFL, and CPPFL+. In all four types, we use the same data settings: the size of voxels and the same 3D point cloud of the model with color information. This process of creating each 3D descriptor is repeated and we obtain as a result an average value for each type of descriptor.

Fig. 4: Models captured with a Kinect camera, belonging to the Kinect database [12].

As noted in Fig. 5, each model has its own descriptor creation time for the 4 mentioned. The lowest values are always obtained by CPPFL+ and as a second place to the CPPFL descriptor. The values obtained are linked to the number of points that each model has after it has been reduced by a Voxel filter.

### 4.2 Recognition phase using the proposed 3D descriptors

For this part, the new descriptor is tested compared to the other 3D descriptors. We first evaluate the effectiveness and then the efficiency of the different descriptors. The values of correctly hit of the position in different scenarios are shown in Table 2. Each row represents a count of the correctly successful models using the PPF, CPPF, CPPFL, and CPPFL+ descriptors, while each column represents the count by a specific type of model for valid combinations of that model.

It is noted that the PPF descriptor is always below the descriptors that use color. As for the three descriptors that are supported by the color information have similar performance, this is because this task is to locate the object in the

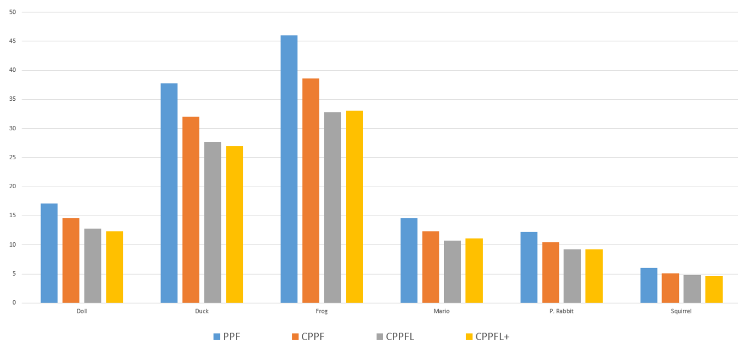



Fig. 5: Time required to create the PPF, CPPF, CPPFL, and CPPFL+ descriptors for kinect database models.

| Own dataset | Doll | Duck | Frog | Mario | P. Rabbit | Squirrel |
|---|---|---|---|---|---|---|
| ppf | 0.28 | 0.0 | 0.12 | 0.15 | 0.39 | 0.56 |
| cppf | 0.28 | 0.0 | 0.15 | 0.15 | 0.37 | 0.48 |
| cppfl | 0.28 | 0.66 | 0.32 | 0.23 | 0.49 | 0.41 |
| cppfl+ | 0.43 | 0.66 | 0.32 | 0.23 | 0.49 | 0.33 |

Table 2: Accuracy of 3D descriptors for translation and orientation.

presented environment. Where it is observed that the proposed descriptors have a better percentage of recognition.

To verify that the proposed descriptor recognizes in real-time, the time it takes for each recognition is displayed, even if it fails. These times are seen in table 3. We have the average recognition time values between the original and proposed descriptors applying the optimized process. These times allow the execution of several frames per second. In addition, it is noted that the CPPFL+ descriptor offers better times in all cases.

| Descriptor | Doll | Duck | Frog | Mario | P. Rabbit | Squirrel |
|---|---|---|---|---|---|---|
| ppf | 2.395 | 5.023 | 4.427 | 1.370 | 1.485 | 0.538 |
| cppf | 0.405 | 0.599 | 0.426 | 0.298 | 0.280 | 0.146 |
| cppfl | 0.506 | 0.833 | 0.839 | 0.244 | 0.263 | 0.102 |
| cppfl+ | 0.317 | 0.510 | 0.370 | 0.202 | 0.193 | 0.086 |

Table 3: Recognition time in seconds

# 5 Tests with own data obtained with the RealSense camera

## 5.1 RealSense camera 3D models

To verify the effectiveness and efficiency obtained with the proposal of this research. The descriptor was tested with an RGB-D image stream, as detailed below. To test the effectiveness of the proposal, 3D models have been used taking into account their difference in color, size and shape, as seen in Fig. 6.


Fig. 6: Models captured with a RealSense camera.

The values of the models used are fixed, so they can be compared between the descriptors: CPPF, CPPFL, and CPPFL+. Resulting in the values in Table 4. Each row represents a model, while each column represents the descriptor used, except the last one, which is the number of 3D points characteristic of the models to observe the time-quantity relationship. Because the objects are relatively smaller than the models in the Kinect database, they have a much shorter processing time than those obtained with the Kinect database because after being reduced it does not have more than 100 characteristic 3D points.

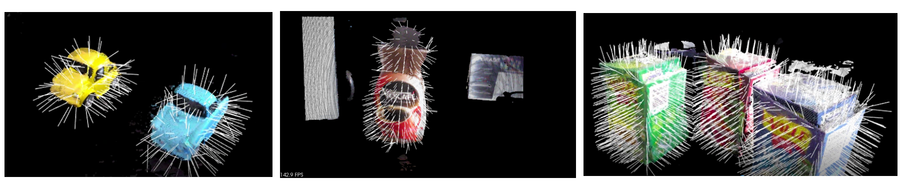
## 5.2 Recognition with own data captured with the RealSense camera

For this part, the same camera with which the models were captured was used. When testing with a data stream that varies by test, these are not directly comparable between descriptors because each frame of the video stream represents a unique scenario.

| own dataset | CPPF(ms) | CPPFL(ms) | CPPFL+(ms) | points |
|---|---|---|---|---|
| yellow car | 1 | 1 | 1 | 17 |
| blue car | 1 | 1 | 1 | 21 |
| coffee | 2 | 1 | 1 | 50 |
| blue box | 4 | 3 | 2 | 71 |
| red box | 2 | 2 | 2 | 66 |
| green box | 3 | 3 | 2 | 69 |

Table 4: Time required for model creation using CPPF, CPPFL, CPPFL+ descriptors for models captured with a RealSense. Adding the number of 3D points for each model after being processed.

We have 3 different sets of scenarios tested. The first is two small vehicles of similar shape, but different colors as seen in Fig. 7a. A similar shape makes it necessary to use color to differentiate those, this could be very important in real applications. The second is of a bottle of coffee being clogged with other objects as in Fig. 7b. It is important to have a tolerance to interference because the environment does not always provide all the information about the object. The third is three boxes of equal shapes, but of different colors, as seen in Fig. 7c. For these tests, geometrically similar objects are used, being boxes almost all the values on position and normal are very similar. The colors of the boxes are red, green, and blue. They are not entirely that color and they are affected by different illuminations.



(a) Models of cars being recognized.

(b) Recognition of a bottle of coffee.

(c) Models of boxes being recognized.

Fig. 7: Video captures of the tests.

| Video | Precision % | CPPFL(ms) | CPPFL+(ms) |
|---|---|---|---|
| 2 cars | 95 | 100-200 | 75-150 |
| Clogged coffee bottle | 80 | 250-800 | 200-600 |
| 3 different color boxes | 85 | 350-900 | 270-650 |

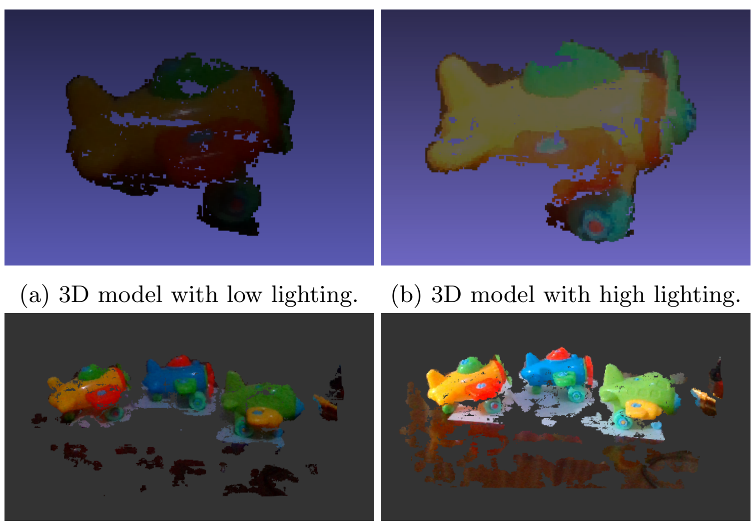
Table 5: Performance of the CPPFL descriptor with an RGB-D camera.

The results of object recognition are presented in Table 5, where each row corresponds to a video. The Precision column indicates a value relative to the number of correct positionings of 3D models across the set of scenarios. As for the other columns they represent the milliseconds required for such a process. These values are displayed as ranges because it is not always the same time that each frame, or RGB-D image, of the video, is rendered.

## 6 Testing with lighting changes

To verify the tolerance to lighting changes in the recognition of 3D objects is that some tests were performed with the proposed descriptor CPPFL. The two most extreme cases are when there is no lighting, and when there is too much lighting to the point of seeing everything white. Consequently, two more practical cases can be raised, when the scenario is dark or clear.

In Fig. 8c shows a scenario of three models of aircraft with low lighting and Fig. 8d shows the same scenario, but with better lighting. These models are created to have a lighting level close to the lighting ends.



(a) 3D model with low lighting. (b) 3D model with high lighting.

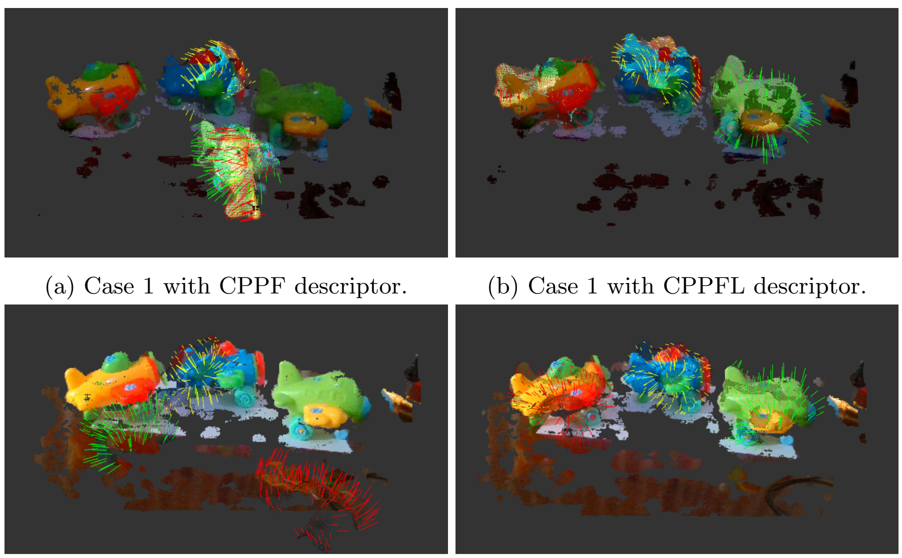
(c) Models with low lighting. (d) Models with lots of lighting.

Fig. 8: Models of 3 aircraft.

From these scenarios, the 3D models for each aircraft are extracted. Fig. 8a has a model with low lighting and Fig. 8b has the same model with good lighting. As it can be seen the dark model is more difficult to notice because the low lighting makes the different colors less light. In the model with better lighting, it is easier to distinguish the colors present and that will be used by the CPPFL descriptor.

Two test cases are created with these models using the CPPF and CPPFL descriptors because both handle color and the models are geometrically similar. In each case, the descriptors of these models are used to recognize them on a stage with different lighting. That is, the descriptors of low-light models are used to recognize objects on a stage with good lighting, while the descriptors of models with good lighting are used to recognize these same 3D objects on a stage with bad lighting. It should be noted that the tests are done on a video stream captured by the RGB-D RealSense camera.

In the first test case in Fig. 9 the detection and alignment of the descriptors of the light models on a dark environment are observed. As for performance, it is observed in Fig. 9a a misalignment and detection of the models using the CPPF descriptor. However, in Fig. 9b it can be seen that there is a better positioning of the models using the proposed CPPFL descriptor, this recognition remains stable for most of the frames in the video.



(a) Case 1 with CPPF descriptor.

(b) Case 1 with CPPFL descriptor.

(c) Case 2 with CPPF descriptor.

(d) Case 2 with CPPFL descriptor.

Fig. 9: (a) Recognition in case 1 with CPPF. (b)Recognition in case 1 with CPPFL. (c) Recognition in case 2 with CPPF. (d) Recognition in case 2 with CPPFL

In the second test case in Fig. 9 the detection and alignment of the descriptors of the dark models on a light environment are observed. Fig. 9c , it is shown that the CPPF descriptor suffers from same detection and alignment problem as in the first case. Even in cases where the overlay seems correct, it is more due to a coincidence because the original colors are no longer present. In contrast, the proposed CPPFL descriptor obtains good detection and alignment of the models.

Then the tolerance that the descriptor has to light changes is highlighted, which is something that happens as the hours of the day change.

To quantify these results, we selected 100 frames from a one-minute RGB-D video. These 100 frames will be processed using the CPPF and CPPFL descriptors. From each video, the models detected and aligned correctly using each descriptor are counted, and in Fig. 10 the results are shown by comparing both descriptors. As can be seen, the CPPFL descriptor finds in a greater number of frames the 3D models of the objects that are present in the video, showing better performance than the CPPF descriptor in scenarios with variable lighting.

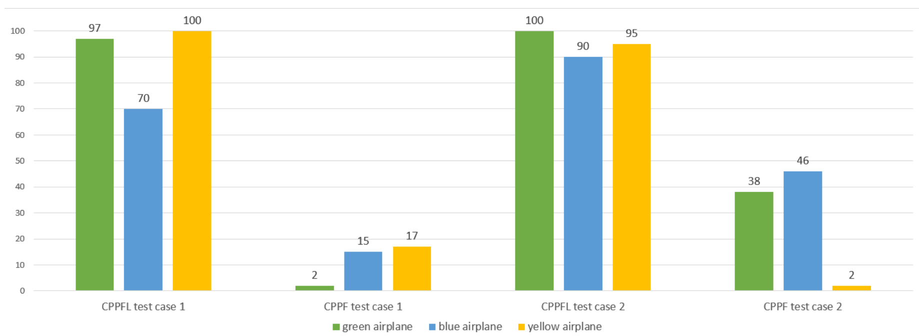



Fig. 10: Test results in different lighting conditions for CPPF and CPPFL descriptors.

## 7 Conclusions

In this academic article, a new descriptor has been proposed for the recognition of 3D objects using depth cameras (RGB-D), which demonstrates an improvement in the detection and alignment of the position and orientation of an object sought in a 3D scenario. Based on the results obtained in this article, the following conclusions were reached:

1. The proposed CPPFL descriptor uses a 6-value vector, where the first 4 retain geometry information and the other 2 stores color-related information.
2. In the process of creation and recognition of 3D objects, different factors are observed to take into account such as the minimum space between points, the maximum number of neighbors to be analyzed, the number of pairs of points used to optimize the result.
3. The CPPFL descriptor works in the RGB color space using segmentation of 9 base colors, 6 chromatic, and 3 achromatic.
4. After optimizing the process, the CPPF descriptor can process the data in almost real-time with good results.

5. The proposed descriptor use segmentation of the principal colors to make it tolerant to lighting changes

As for the accuracy of the CPPFL descriptor in the evaluation of the detection of a 3D object in a scenario, there is an improvement of 8having a significant improvement over the CPPF descriptor.

## 8 AKNOWLEDGMENT

M. E. LOAIZA acknowledges the financial support of the CONCYTEC – BANCO MUNDIAL Project "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica" 8682-PE, through its executing unit PROCIENCIA, within the framework of the call E041-01, Contract No. 038-2018-FONDECYT-BM-IADT-AV

## References

1. B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Proceedings of the IEEE International Conference on Computer Vision.* IEEE, 2010, pp. 998–1005.
2. C. Choi and H. I. Christensen, "Rgb-d object pose estimation in unstructured environments," *Robotics and Autonomous Systems*, vol. 75, pp. 595–613, 2016.
3. S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *European Conference on Computer Vision.* Springer, 2016, pp. 834–848.
4. R. Szeliski, *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.
5. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision.* Springer, 2006, pp. 404–417.
6. A. Saïdani and A. K. Echi, "Pyramid histogram of oriented gradient for machine-printed/handwritten and arabic/latin word discrimination," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR).* IEEE, 2014, pp. 267–272.
7. L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics.* Springer, 2013, pp. 387–402.
8. I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
9. R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on.* Citeseer, 2009, pp. 3212–3217.
10. S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.
11. W. Wang, L. Chen, Z. Liu, K. Kühnlenz, and D. Burschka, "Textured/textureless object recognition and pose estimation using rgb-d image," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 667–682, 2015.
12. F. Tombari and S. Salti, "3d keypoint detection benchmark," https://vision.deis.unibo.it/keypoints3d/, 2011, [updated 2018-12-15].