U-CATCH: Using Color ATtribute of image patCHes in binary descriptors

Ozgur Yilmaz[1] and Alisher Abdulkhaev[2]

[1,2]Turgut Özal University, Department of Computer Engineering, Ankara Turkey

March 29, 2016

Abstract

In this study, we propose a simple yet very effective method for extracting color information through binary feature description framework. Our method expands the dimension of binary comparisons into RGB and YCbCr spaces, showing more than 100% matching improvement compared to non-color binary descriptors for a wide range of hard-to-match cases. The proposed method is general and can be applied to any binary descriptor to make it color sensitive. It is faster than classical binary descriptors for RGB sampling due to the abandonment of grayscale conversion and has almost identical complexity (insignificant compared to smoothing operation) for YCbCr sampling.

Keywords: Binary Feature Descriptor, Color, BRIEF, LATCH, RGB, YCbCr

1 Introduction

Local feature descriptors are widely used in applications such as 3D reconstruction [6], object detection[5], scene recognition[17]. SIFT[7], HOG [8], SURF[**?**], GLOH[9] use histogram of image gradients and/or gradient orientations, and they have been very successfully applied to many problems in computer vision that require correspondence matching.

A taxonomy of descriptors (BRIEF, ORB, BRISK, LDAHash, DAISY [18], SIFT, SURF, etc) based on their computational and storage requirements are given in [10]. Heinly et al., emphasize that binary descriptors are faster to compute and match also requiring lower storage compared to non-binary descriptors.

Binary descriptors [10] provide a great advantage among feature descriptors in terms of speed without losing too much accuracy. The reason is twofold: 1. relative intensity captures most of the information in image regions for matching [16], 2. similarity evaluation of two binary vectors can be done much faster compared to real-valued vectors of non-binary descriptors. This similarity (Hamming distance) of two binary vectors can be performed by counting the incompatible bits via XOR operation and XOR can be done very fast on modern CPUs [2].

Binary descriptors compare the pixel or average patch values according to a spatial sampling pattern and generate a binary vector. They exploit a fixed sampling pattern but the generation of the pattern varies: BRIEF (Binary Robust Independent Elementary Features) uses a random sampling pattern, while LATCH (Learned Arrangements of Three Patch Codes) exploits learned patch triplet arrangements.

Although, BRIEF and LATCH descriptors perform well at feature matching, there is no utilization of color attribute, which is a very important property of visual objects. Binary descriptors transform the input images into gray scale. Since modern cameras are RGB, classical binary descriptors throw away information that might be valuable for discrimination. In literature, there are some studies [3] which use color channels independently on existing feature extraction algorithms. For example RGB-SURF algorithm computes features from each color channel and then combine them to get the overall color representation. In this study we propose a new perspective for exploiting color information in images using binary descriptors. We sampled the pixel locations in three dimensions, without treating each channel separately or dividing the image into color channels. We show that the cross-channel comparison via 3D sampling captures the color property of image patches very effectively and efficiently. This approach can be used in any binary descriptor regardless of their sampling patterns. In this paper, we applied our idea on BRIEF and LATCH descriptors. In our experiments, we sampled pixel locations in two color spaces: RGB and YCbCr. Since RGB and YCbCr color spaces represents chromaticity in distinct ways, we experimented with both of them for completeness and show 100% improvement in matching performance.

2 Review of Binary and Color Descriptors

2.1 BRIEF

BRIEF[2] was the first proposed binary descriptor with surprisingly well matching performance. It suggests that for discrimination among feature locations in the image, relative intensity holds most of the information, which is also exploited in local binary pattern approaches [16]. It samples the pixel locations around a key point randomly due to a fixed distribution (Equation 4). However it has been shown that non-random, i.e. learned sampling pattern can be superior [11]. BRIEF compares two pixel intensities in order to build a binary vector. Binary comparison of intensity values is very noise-sensitive especially around relatively uniform regions. In order to make the descriptor less sensitive to noise, smoothing is applied on the window (size 48x48). Gaussian smoothing is used, with the standart deviation of 2 and kernel window size of size 9x9. Effect of pre-smoothing is very significant as stated in [2].

Spatial arrangements of binary tests are randomly sampled from the probability distribution function shown in equation 4. For each sampling pair, the descriptor compares the intensity in the location of the first pixel ”x” to that of second pixel ”y” in patch ”p”; if the intensity is greater then it assigns 1 in the final descriptor ”f” and 0 otherwise (Equation 1)

$$f(p; x, y) = \begin{cases} 1 & p(x) < p(y) \\ 0 & otherwise \end{cases} \tag{1}$$

The comparisons result in a binary bitstring ”b_f” (Equation 2).

$$b_f(p) = \sum_{1 \leq i \leq n_d} 2^{i-1} f(p; x, y) \tag{2}$$

The distance between two features can be computed very efficiently using the Hamming function.

Extensions of BRIEF:

- ORB (Oriented fast and Rotated BRIEF) descriptor[11] adds rotation invariance into existing BRIEF descriptor. Another contribution of ORB is the use of unsupervised learning in order to select optimal pixel location pairs.

- BRISK (Binary Robust Invariant Scalable Keypoints)[12] use concentric ring-based sampling patterns. BRISK compares two pixel pairs far from each other to compute the patch orientations and pixel pairs close to each other to compute the values of the descriptor itself.

- FREAK (Fast REtinA Keypoint)[13] also uses concentric rings arrangement with samples exponentially more pixel locations in the inner rings and uses unsupervised learning to choose optimal set of pixel location pairs.

- LDB (Local Difference Binary)[14] descriptor compares mean intensities in grids. LDB also compares the mean values of horizontal and vertical gradient differences.

LDA-Hash[15] generates binary feature vectors yet the algorithm first computes SIFT and project the feature vector onto more discriminant space and uses a threshold in order to build binary vector. Thus, although the output is binary, the feature computation is substantially different from BRIEF and its variants, because LDA-Hash first extracts the SIFT descriptors and then generates binary vectors. Similarly, even though BinBoost [19] generates binary feature vectors by utilizing binary hash functions, it relies on computation of gradient orientations in rectangular image regions.

2.2 LATCH

LATCH is the state-of-the-art binary descriptor giving comparable matching performance with best of non-binary descriptors [1]. Though, BRIEF makes pre-smoothing of an image, this smoothing may cause to loss of high-frequency regions where key points are often detected. To overcome this issue, LATCH compares the patch triplets instead of pixel pairs. LATCH evaluates (equation 3) the similarity of the ”anchor” patch ($P_{t,a}$) to its two ”companion” patches ($P_{t,1}$ and $P_{t,2}$) by computing their Frobenious norm and assigns 1 if Frobenious norm of difference of ”anchor” and ”companion1” patches is bigger than Frobenious norm of difference of ”anchor” and ”companion2” patches; otherwise assigns 0. LATCH takes the windows of size 48x48 around key points as BRIEF, and size of anchor and companion patches are fixed to 7x7.

$$g(W, s_t) = \begin{cases} 1 & \text{if } \|P_{t,a} - P_{t,1}\|_F^2 > \|P_{t,a} - P_{t,2}\|_F^2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

LATCH does not sample the locations of patch triplets randomly. Instead, patch triplets are learned from Liberty, Notre Dame, and Yosemite dataset collections by supervised learning. C++ implementation of LATCH descriptor is available, and we based our experiments on this implementation.

2.3 Other Color Descriptors

Most of the extensions of existing feature extractors with color, ”partially” use color attributes of image patches. [3] handled color extensions of local descriptors for Video Concept Detection. They present a RGB-SURF, RGB-ORB, OpponentSURF and OpponentORB descriptors. In contrast to conventional descriptors, RGB-SURF and RGB-ORB apply original SURF or ORB descriptors directly to each of the RGB channels and concatenate feature vectors obtained from each channel, thus the size of the vector is tripled. In the case of OpponentSURF/OpponentORB, algorithm initially transform the RGB image to the opponent color space and use the transformed channels O_1, O_2 and O_3. O_3 is the luminance channel, i.e. the one that the original SURF/ORB descriptors use. The other two channels (O_1 and O_2) capture the color information, where O_1 is the red-green component and O_2 is the blue-yellow component. Similar with RGB-SURF/ORB, the original SURF or ORB descriptors are then applied separately to each transformed channel and the final feature vector is the concatenation of the three feature vectors extracted from the three channels independently.

In [4] a wide set of color descriptors and their invariant properties are examined for recognition accuracy. Color descriptors that are based on color histograms and SIFT based orientation histograms are compared. Koen van de Sande et al. [4], state that SIFT descriptor is not invariant to light color changes, because the intensity channel, i.e. gray scale is a combination of the R, G and B channels. They proposed HSV-SIFT, where original SIFT descriptors is implemented to each of the channels of HSV color space.

In general, most of the color approaches in feature extraction treat channels independently, missing the rich cross-channel statistics. Opponent color space alleviates this problem to some extent and show best results in object recognition, yet there is no mixing between red-green and blue-yellow components, more importantly the size of the vector needs to be tripled. Another reason for us to avoid opponent channel approach is for minimizing the computational load for extraction because it at least requires a color transformation operation to the whole image.

3 Method

Conventional binary feature descriptors convert the image into gray scale, then binary comparisons are performed on a fixed size window around the detected keypoint. BRIEF descriptor uses randomly generated pixel location pairs and compare the grayscale value of these pairs. The image is smoothed before pixel value comparison for robustness to noise. Pixel value comparisons creates a binary feature vector, the size of which is determined by the total number of comparisons. Instead of single pixel value comparison at two locations, LATCH utilizes small image patches at three pixel locations. LATCH uses pixel patch triplets in order to avoid any potential loss of high frequencies in image. LATCH compares the sum of pixel intensities of patch windows of size $\left[7{,}7\right]$. Both BRIEF and LATCH uses the detection windows of size $\left[48{,}48\right]$ in their OpenCV implementations. We used the same parameters in our experiments for consistency. It should be noted that the both of the descriptors focus on intensity level of image in their comparisons. However RGB image carries both the color and intensity information. On the other hand YCbCr represents the intensity and color attributes better. We have focused on making comparisons, either BRIEF and LATCH, on Color Space in order to catch the color information in addition to intensity information. Another advantage of our approach is decreasing the computational cost by avoiding color transformation.

3.1 Sampling in RGB Space

BRIEF and LATCH uses the grayscale pixel values in binary comparisons which loses the color information. However, a very straightforward extension of these descriptors can be designed which utilizes the color information. In order to use color property we made binary comparisons on the RGB image. We have used the same parameters with OpenCV implementations of BRIEF and LATCH.

Additional advantage of this approach is not to transform the original image into gray scale, thus decreasing the computational cost. We call the RGB binary comparison as "3D RGB Sampling". 3D RGB Sampling is visualized in Fig.1. BRIEF uses the pixel locations that are sampled from an isotropic Gaussian distribution (4). Sampling from this distribution generates pixel locations that are closer to detected key points in 2D image.

$$\mathbf{(X,Y)} \sim i.i.d.Gaussian(0, \frac{1}{25}S^2) \tag{4}$$

In addition to this distribution, we generate uniformly distributed numbers {1,2,3} that indicate the color channels. Instead of sampling pixel intensities in 2-Dimensional space 1, we sample them in 3-Dimensional Space Fig.1, i.e. sampling is done among all channels of an image: R-channel, G-channel and B-channel. With this sampling approach, we create three dimensional pixel locations. 2D random distribution in Fig.2 is stretched among 3rd dimension and got the distribution Figured in Fig.1. In 3D Sampling, some comparisons might be made on the same color channel giving a similar comparison as in conventional binary descriptors in single images. On the contrary, the comparison might be cross-channel, e.g. the first could be chosen from R-channel, while the second one could be chosen from G-channel. Two points that are selected from the same channel is equivalent with 2D Sampling, while the points that are selected from different channels imply comparing the R-channel intensity with the intensity of the G-channel. Basically, cross-channel comparison asks the question "Whether the points are more reddish or more bluish or more greenish?". While 2D comparisons in BRIEF or latch descriptors answers the question "Which point's intensity is greater?". Existing binary descriptors are not seeking for any color-based information, while our approach extracts color information. With this simple modification we show dramatic improvement in feature matching performance. Some sample point selections and their meanings are given in Table 1.

In the case of color-enhanced LATCH descriptor, in a similar vein pixel patches are sampled from 3D RGB Space instead of grayscale patches.

3.2 Sampling in YCbCr Space

RGB binary comparison ignores the intensity of pixels since it treats color values individually and does not take into account the grayscale value. While, 3D RGB Sampling improves the performance, it can be further improved by making binary comparisons in YCbCr color space. YCbCr represents intensity and color attributes separately, so it will be very beneficial for using intensity and color at the same time. Conversion from RGB to YCbCr is almost equivalent to the cost of conversion from RGB to Gray. YCbCr separates the intensity (luma) from color information (chroma). "Y" component refers to luma component, while "Cb" and "Cr" refers to blue-difference and red-difference chroma components, respectively. In this color space we can handle the both luminance information and color information separately. That's why the performance of

Table 1: Various selections of pixel points used in 3D RGB space comparison 1.

1$^{\text{th}}$ Point	2$^{\text{nd}}$ Point	Explanation
[5,5,1]	[8,8,1]	1. Which pixel intensity is greater; [5,5] on R-channel or [8,8] on R-channel?
[5,5,1]	[5,5,3]	1. "Whether the pixel intensity at [5,5] on R-channel is greater than pixel intensity at [5,5] on B-channel or not?" 2. "Which point is more reddish/bluish?"
[3,3,2]	[17,22,3]	1. "Whether the pixel intensity at [3,3] on R-channel is greater than pixel intensity at [17,22] on B-channel or not?" 2. "Which point is more greenish/bluish?"
...	...	...

binary descriptors can be further improved by sampling in YCbCr color space as it is demonstrated in experimental results section4. 3D YCbCr Sampling is visualized in Fig.1.

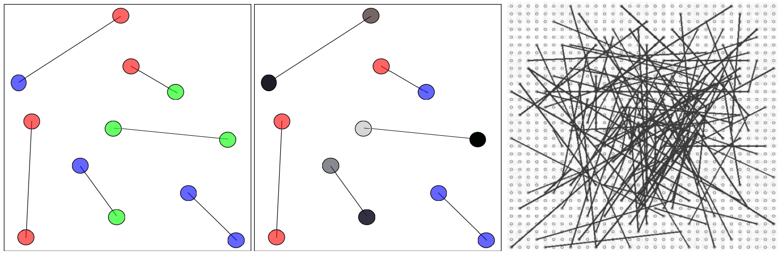


Figure 1: **Left.** Points shown with red, green, blue colors indicate the pixel location that are taken from R-, G- and B-channels of RGB image, respectively. Stick with red and green colors at the ends, for example, compares the pixel intensity of R-channel with the pixel intensity G-channel of RGB image (see Table 1). **Middle.** Points shown with gray, blue and red points indicate the samples taken from Y-(luma), Cb-,Cr-(chroma) channels of YCbCr image, respectively. Tones of gray in Y-channel represent the pixel intensities in Y-channel. **Right.** 2D binary sampling pattern with the distribution shown in equation 4 used in [2].

Luma component in YCbCr space represents the light intensity, while chroma represents the color. Luma component holds brightness information, therefore we sampled from Y-channel separately as in classical binary descriptors and sampled from Cb- and Cr-channels as we did in RGB 3D case explained above. We do not mix intensity and color comparisons. For color information extraction, channel numbers are randomly generated, that indicate the Cb and Cr channels. So, in this YCbCr comparison, some number of comparisons are done in Y-channel only, and the other comparisons were done among Cb- and Cr-channels. 3D YCbCr Sampling is visualized in Fig.1. Examples of 3D YCbCr sampling and their explanations are given in Table 2. With this kind of comparison in color space, resultant bits of binary descriptors include color information in addition to intensity information. In 3D YCbCr Sampling, descriptor performance is improved further as it is demonstrated in experimental results section4. Binary comparisons in YCbCr space gives the best results almost in all sequences. But, if computational cost is very crucial, then 3D RGB Sampling can also be used, while it performs significantly better than 2D Gray Sampling.

Table 2: Various selections of pixel points used in 3D YCbCr space comparison. Note that there is not any samplings among Y and Cb/Cr channels 1

.

1^{th} Point	2^{nd} Point	Explanation
[5,5,1]	[8,8,1]	1. Which pixel's luminance value is greater; [5,5] on Y-channel or [8,8] on Y-channel?
[3,3,2]	[17,22,2]	1. "Whether the pixel chrominance at [3,3] on Cb-channel is greater than pixel chrominance at [17,22] on Cb-channel or not?"
[5,5,2]	[5,5,3]	1. "Whether the pixel chrominance at [5,5] on Cb-channel is greater than pixel chrominance at [5,5] on Cr-channel or not?" 2. "Which point is more bluish/reddish?"
...	...	...

Very similar to the extension in RGB case, LATCH descriptor in YCbCr is done by sampling the pixel patches from either "Y" channel or among "Cb-", "Cr-" channels. Therefore, with 3D YCbCr sampling in LATCH descriptor we achieve simultaneous intensity/color comparisons avoiding the loss of high frequencies of the image by using patches instead of pixels.

3.3 Feature Extraction Procedure & Performance Evaluation

In experiments, feature extraction and matching procedures are taken from [2]. Interest points were detected from the first image, and mapped into second image using the homography matrix, They are called interest points of the second image. Interest point detection in the second image is avoided in order not to account the repeatability performance of the detector in evaluating the descriptor performance. Step by step details of feature mapping and performance evaluations are given below:

1. Read two images that are going to be matched.

2. Detect the keypoints from first image using one of detectors and select **N** points among them. (detectors of SURF, ORB, CenCurE, SIFT, FAST, etc).

3. Map those points to second image using homography matrix. Refer these points as a interest points in the second image. (Not detecting keypoints from the second image)

4. Discard the points that are too close to borders of both first and second image.

5. Extract the descriptor for keypoints locations of first image.

6. Extract the descriptor bits from the mapped pixel locations at the second image.

7. For a point in a first set of descriptors (image 1), search for the nearest descriptor in the second set of descriptors (image 2), and call them as a match.

8. Compute the number of correct matches, defined by the homography.

9. Evaluate the performance by dividing the **number of correct matches** by **N** -Total number of handled keypoints.

Relative improvements due to color information are calculated as follows:

$$\mathbf{RI}\% = \left(\frac{P' - P}{P} * 100\%\right) \tag{5}$$

"P - 2D Performance", "P' - 3D Performance" and "RI - Relative Improvement".

4 Experimental Results

We have used the Oxford dataset [1] in our experiments (whole set of results on the dataset are given as a supplementary document). Oxford dataset consist of 8 image sets with 5 different changes in imaging conditions:

- viewpoint changes (graf, wall imagesets)
- scale changes (bark, boat imagesets)
- image blur (bikes, trees imagesets)
- JPEG compression (ubc imageset)
- illumination (leuven imageset)

[1] www.robots.ox.ac.uk/~vgg/research/affine

We used all imagesets except ”boat” imageset because it is in grayscale, unable to give color information which is the purpose of this study. Detailed information about the sequences can be obtained from the Oxford dataset’s site. Each set contains 6 images presenting increasing difficulty in the conditions stated above. We make descriptor matching between the first image and each of remaining five images. Feature decription and matching stages are stated in Section 3.3. We compared our approach with BRIEF and LATCH desciptors. In our experiments we did not exploit the ”rotation invariance” attribute of LATCH, because we mapped the detected keypoint on the first image into second image by the use of homography matrix rather than detecting the keypoints from both images independently. Gil [1] showed that LATCH descriptor outperforms the BRIEF due to its robustness against ”rotation” changes. But in our experiments, because we did not use ”rotation invariance”, LATCH sometimes falls behind BRIEF. Our purpose in showing results in LATCH without rotation invariance is presenting the generality of color inclusion into binary descriptors, whether it utilizes pixelwise or patchwise comparison.

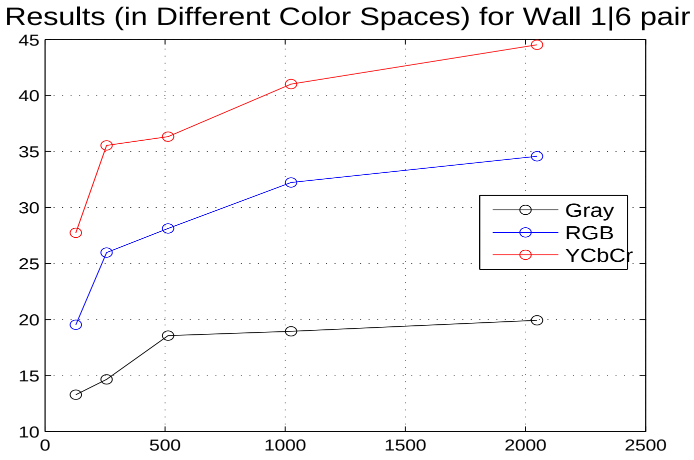


Figure 2: Performances of Gray, RGB and YCbCr Samplings vs. Number of Binary Comparisons.

Performances as a function of the number of binary comparisons In [2], performance of BRIEF descriptor with respect to the number of binary comparisons (tests) is examined and it is observed that BRIEF requires at most 214 bits for Wall dataset to reach the performance of U-SURF descriptor, which requires 2048 bits. However, saturation can be observed in BRIEF for binary comparisons bigger than ∼512 bits. Our RGB-BRIEF and YCbCr-BRIEF outperforms the classical BRIEF at each number of tests (Figure 2). More importantly color enhanced binary descriptors do not saturate for small number of bits, but they go up. It leads to higher performances when we have computational power to handle more bits in binary feature extraction, thus it is looking to the future for faster hardware whereas classical binary descriptors will be unable to exploit them. Yet, if due to speed limitations it is required to use small number of bits, then our RGB-BRIEF and YCbCr-BRIEF still outperforms the classical BRIEF (Please see Supplementary Document for more results).

For detailed matching results, image matching performance between first and the rest of the images are given separately in bar charts (eg. 1|2, 1|3, 1|4, 1|5 and 1|6 in Figure 3) and Table 3 for gray and and color descriptors. The matching performance drops with image number due to decreasing similarity with the first image. Please note that the term "Gray" for an algorithm corresponds to the original algorithm, i.e. BRIEF-Gray is the feature descriptor proposed in [2]..

1. Wall imageset: Comparative results are shown in Figure 3.

- wall1|2: Both BRIEF and LATCH were very successful. Color-enhanced features gives the same, even better results. Matching score are written at the bottom of each bar. [2]
- wall1|3: Similar results with wall1|2.
- wall1|4: Since viewpoint changes more, color-enhanced BRIEF and LATCH gives significantly better performances. Gray BRIEF gives 77% performance, while RGB BRIEF and YCbCr BRIEF gives the 84% and 90%, respectively. This improvement is due to the usage of color which helps to withstand against changes in viewpoint.
- wall1|5: The improvement due to color is much more visible in this case. The relative performance improvement is 12% and 51% for RGB-BRIEF and YCbCr-BRIEF respectively.
- wall1|6: Improvements become dramatic because of much challenging viewpoint change. RGB-BRIEF shows 51% and YCbCr-BRIEF shows 95% matching improvement.

2. Graf imageset: Figure 3

In Graf imageset, impact of color information can be observed much more clearly. The improvement in matching performance is experienced even in matching the first and second images. RGB-BRIEF relative improvements are: 21%, 14%, 65%, 54%, 52% for each successive matchings. YCbCr-BRIEF's relative improvements are: 46%, 46%, 102%, 122%, and 103% respectively. The performance is **doubled** by the usage of color information, which does not add any computational cost.

3. Trees imageset: Figure 4

Again we observe significant improvements in performance. RGB-BRIEF relative improvements are: 21%, 14%, 65%, 54%, 52% for each successive match-

[2] Due to ignorance of rotation invariance of LATCH, its matching results are worse compared to matching scores of BRIEF both in Gray and Color cases.

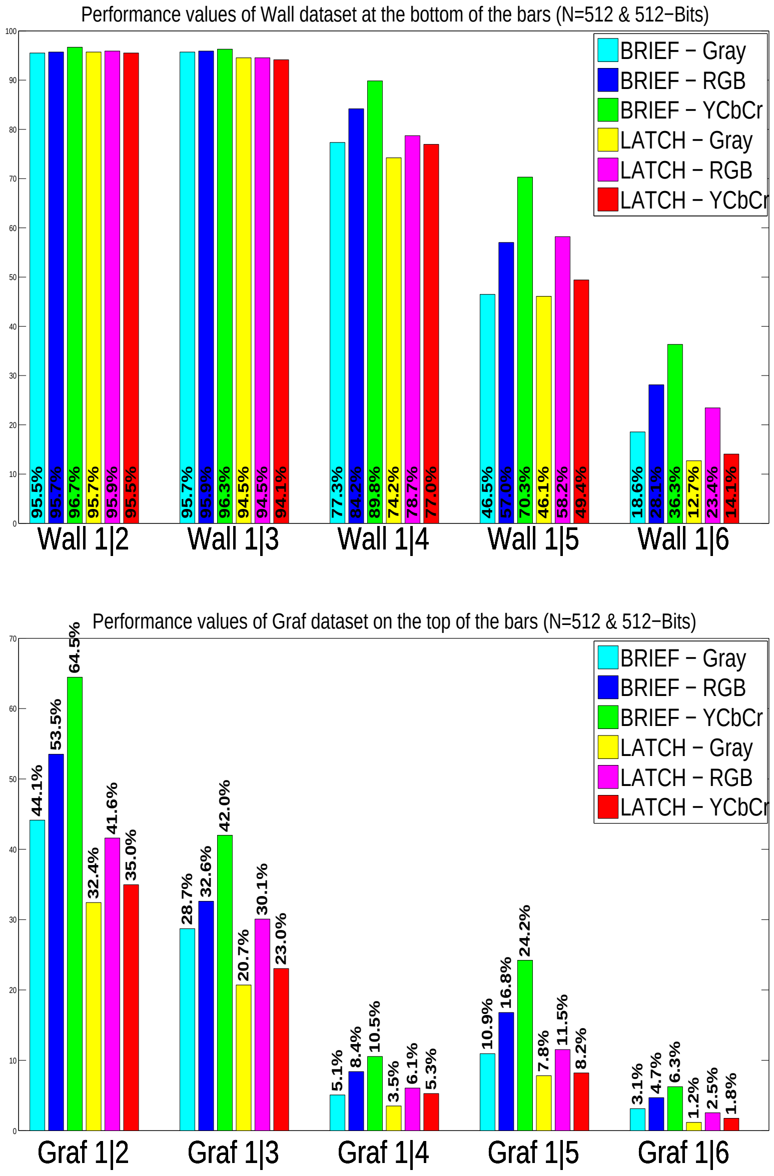


Figure 3: 2D and 3D Binary Comparisons in Different Color Spaces: Wall and Graf imagesets. 512 Bits descriptors are used. N:Number of Keypoints to be matched.

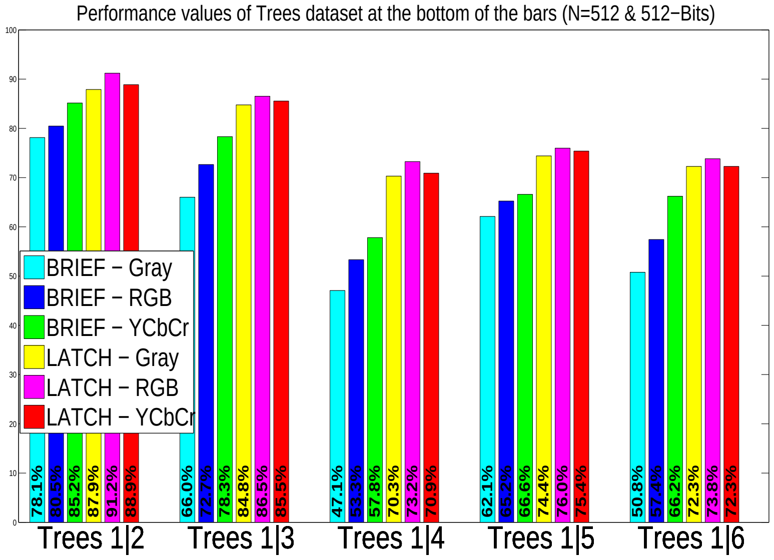


Figure 4: 2D and 3D Binary Comparisons in Different Color Spaces: trees imageset. 512 Bits descriptors are used. N:Number of Keypoints to be matched.

ings. YCbCr-BRIEF's relative improvements are: 46%, 46%, 102%, 122%, and 103% respectively. [3]

4. Ubc, Bark, Leuven and Bike imagesets: [4]

In three of the rest of the imagesets (Ubc, Leuven, Bike), BRIEF and LATCH descriptor performance saturate at very high matching scores. It is crucial that color descriptors do not fall behind the gray scale versions in the case of saturation. The performance for Bark set is very low even for the first two images, yet color information adds significant improvement (please see Supplementary Document for the bar plots).

[3] Since imagining condition in "trees" imageset is blur rather than rotation (Note that we turned off the rotation invariance of LATCH descriptor), LATCH outperforms the BRIEF in this imageset, while in previous two imagesets it does not.

[4] Figures are given in Supplementary Document

Table 3: Performances of BRIEF and LATCH in different color spaces: **Gray(2D)**, **RGB (3D)** and **YCbCr (3D)** Sampling
RGB-RI: Relative Improvement in RGB space5
YCbCr-RI: Relative Improvement in YCbCr space5

Dataset	**BRIEF (%)**					**LATCH (%)**				
	Gray	**RGB**	**RGB-RI**	**YCbCr**	**YCbCr-RI**	**Gray**	**RGB**	**RGB-RI**	**YCbCr**	**YCbCr-RI**
wall 1|2	95.5	95.7	**∼0%**	96.7	**∼1%**	95.7	95.9	**∼0%**	95.5	**∼0%**
wall 1|3	95.7	95.9	**∼0%**	96.3	**∼1%**	94.5	94.5	**∼0%**	94.1	**∼0%**
wall 1|4	77.3	84.2	**∼9%**	89.8	**∼16%**	74.2	78.7	**∼6%**	77.0	**∼4%**
wall 1|5	46.5	57.0	**∼23%**	70.3	**∼51%**	46.1	58.2	**∼26%**	49.4	**∼7%**
wall 1|6	18.6	28.1	**∼51%**	36.3	**∼95%**	12.7	23.4	**∼84%**	14.1	**∼11%**
graf 1|2	44.1	53.5	**∼21%**	64.5	**∼46%**	32.4	41.6	**∼28%**	35.0	**∼8%**
graf 1|3	28.7	32.6	**∼14%**	42.0	**∼46%**	20.7	30.1	**∼45%**	23.0	**∼11%**
graf 1|4	5.1	8.4	**∼65%**	10.5	**∼106%**	3.5	6.1	**∼74%**	5.3	**∼51%**
graf 1|5	10.9	16.8	**∼54%**	24.2	**∼122%**	7.8	11.5	**∼47%**	8.2	**∼5%**
graf 1|6	3.1	4.7	**∼52%**	6.3	**∼103%**	1.2	2.5	**∼108%**	1.8	**∼50%**
trees 1|2	78.1	80.5	**∼3%**	85.2	**∼9%**	87.9	91.2	**∼4%**	88.9	**∼1%**
trees 1|3	66.0	72.7	**∼10%**	78.3	**∼19%**	84.8	86.5	**∼2%**	85.5	**∼1%**
trees 1|4	47.1	53.3	**∼13%**	57.8	**∼23%**	70.3	73.2	**∼4%**	70.9	**∼1%**
trees 1|5	62.1	65.2	**∼5%**	66.6	**∼7%**	74.4	76.0	**∼2%**	75.4	**∼1%**
trees 1|6	50.8	57.4	**∼12%**	66.2	**∼30%**	72.3	73.8	**∼2%**	72.3	**∼0%**
ubc 1|2	100	100	**∼0%**	99.8	**∼0%**	100	100	**∼0%**	99.8	**∼0%**
ubc 1|3	100	100	**∼0%**	100	**∼0%**	100	100	**∼0%**	98.4	**∼0%**
ubc 1|4	100	99.8	**∼0%**	99.4	**∼0%**	100	100	**∼0%**	96.7	**∼-3%**
ubc 1|5	97.9	97.3	**∼0%**	95.1	**∼-3%**	99.6	99.4	**∼0%**	95.5	**∼-4%**
ubc 1|6	94.7	95.5	**∼1%**	93.2	**∼-2%**	97.9	97.9	**∼0%**	93.2	**∼-5%**
leuven1|2	95.7	95.5	**∼0%**	95.5	**∼0%**	97.3	96.5	**∼-1%**	97.5	**∼0%**
leuven1|3	94.7	94.3	**∼0%**	95.5	**∼1%**	95.9	94.3	**∼-2%**	95.3	**∼-1%**
leuven1|4	94.1	91.8	**∼-2%**	94.5	**∼0%**	95.1	93.6	**∼-2%**	94.9	**∼0%**
leuven1|5	92.2	90.2	**∼-2%**	91.8	**∼0%**	95.1	91.6	**∼-4%**	93.4	**∼-2%**
leuven1|6	89.5	88.3	**∼-1%**	89.6	**∼0%**	93.2	90.6	**∼-3%**	91.6	**∼-2%**
bikes 1|2	94.5	94.5	**∼0%**	94.7	**∼0%**	96.3	96.1	**∼0%**	97.1	**∼1%**
bikes 1|3	95.7	94.7	**∼-1%**	94.7	**∼-1%**	96.1	96.1	**∼0%**	96.3	**∼0%**
bikes 1|4	95.3	95.1	**∼0%**	93.2	**∼-2%**	95.5	96.1	**∼1%**	95.9	**∼0%**
bikes 1|5	92.0	94.9	**∼3%**	91.6	**∼0%**	95.7	95.1	**∼-1%**	96.5	**∼1%**
bikes 1|6	86.3	86.3	**∼0%**	85.0	**∼-2%**	90.0	88.7	**∼-1%**	91.2	**∼1%**
bark 1|2	4.3	11.7	**∼172%**	18.0	**∼318%**	2.0	7.0	**∼250%**	3.5	**∼75%**
bark 1|3	0.0	0.4	**-**	0.2	**-**	0.2	0.2	**∼0%**	0.4	**∼100%**
bark 1|4	0.0	0.2	**-**	0.0	**-**	0.4	0.0	**-**	0.4	**-**
bark 1|5	0.1	0.8	**∼700%**	1.6	**∼1500%**	0.2	1.0	**∼400%**	0.2	**∼-80%**
bark 1|6	0.2	0.4	**∼100%**	0.8	**∼300%**	0.2	0.4	**∼100%**	0.2	**∼-100%**

5 Discussion

We offer a very simple extension of existing binary descriptors: make binary comparison in 3D color image volume, not the 2D grayscale image. It is interesting that this idea was not tested before. We show dramatic matching improvements in standart imagesets. If comparison is performed in RGB space, it also yields a speedup due to the fact that grayscale conversion can be avoided. If YCbCr color space is used, performance is even better, and the computational cost is almost identical to 2D-Gray binary descriptors.

YCbCr color space sampling is superficially similar to OpponentORB [3, 4]. However, even though opponent channels mix color information to some extent, each channel is still treated separately. Yet, random binary comparisons in Cb/Cr fully mix the color information due to intrachannel and interchannel components.

In the future, we are planning to learn optimal sampling patterns [20, 11, 19] in color spaces. This optimization might be more crucial in color than grayscale because of the curse of dimensionality. We would like to make more comprehensive tests on the color binary descriptors, specifically on object/scene detection applications.

6 Acknowledgments

This research is supported by The Scientific and Technological Research Council of Turkey (TUBİTAK) Career Grant, No: 114E554.

References

[1] Levi, G., Hassner, T.: LATCH: learned arrangements of three patch codes. In: Winter Conference on Applications of Computer Vision (WACV), IEEE (2016)

[2] Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. Computer Vision ECCV 2010 6314 (2010) 778–792

[3] Markatopoulou, F., Pittaras, N., Papadopoulou, O., Mezaris, V., Patras, I.: A study on the use of a binary local descriptor and color extensions of local descriptors for video concept

[4] van de Sande, K.E., Gevers, T., Snoek, C.G.: Evaluating color descriptors for object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1582–1596

[5] Murphy, K., Torralba, A., Eaton, D., Freeman, W.: Object detection and local- ization using local and global features. Lecture Notes in Computer Science 4170 (2010) 382–400

[6] Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. Computer Vision ECCV 3951 (2006) 404–417

[7] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Interna- tional Journal of Computer Vision 60(2) (2004) 91–110

[8] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. Com- puter Vision and Pattern Recognition 1 (2005) 886–893

[9] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 3951 (2005) 1615–1630

[10] Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. European Conference on Computer Vision, Springer 7573 (2012) 759–773

[11] Rublee, E., Rabaud, V., Konolige, K., Bradskim, G.: Orb: An efficient alternative to sift or surf. IEEE International Conference on Computer Vision, IEEE (2011) 2564–2571

[12] Leutenegger, S., Chli, M., Siegwart, R.Y.: Brisk: Binary robust invariant scalable keypoints. IEEE International Conference on Computer Vision, IEEE (2011) 2548– 2555

[13] Alahi, A., Ortiz, R., Vandergheynst, P.: Freak: Fast retina keypoint. Computer Vision and Pattern Recognition (CVPR), IEEE (2012) 510–517

[14] Yang, X., Cheng, K.T.: Local difference binary for ultrafast and distinctive feature description. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE 36 (2013) 188–194

[15] Strecha, C., Bronstein, A., Bronstein, M., Fua, P.: Ldahash: Improved matching with smaller descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE 34 (2011) 66–78

[16] Ojala, T., Pietikainen, M., Maenpaa, T.: A generalized local binary pattern op- erator for multiresolution gray scale and rotation invariant texture classification. Advances in Pattern Recognition ICAPR 2013 (2001) 66–78

[17] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 2., IEEE (2006) 2169–2178

[18] Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide- baseline stereo. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(5) (2010) 815–830

[19] Trzcinski, T., Christoudias, M., Fua, P., Lepetit, V.: Boosting binary keypoint descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2874–2881

[20] Trzcinski, T., Lepetit, V.: Efficient discriminative projections for compact binary descriptors. In: Computer Vision–ECCV 2012. Springer (2012) 228–242

SUPPLEMENTARY DOCUMENT: U-CATCH: Using Color ATtribute of image patCHes in Binary Descriptors

Ozgur Yilmaz[1] and Alisher Abdulkhaev[2]

[1,2]Turgut Özal University, Department of Computer Engineering, Ankara Turkey

March 29, 2016

1 Performances of other image sets (Ubc, Bark, Leuven, Bikes) in different color spaces.

Performance of gray descriptors, both BRIEF and LATCH, already reaches to a very high level on each Ubc1 image pairs. RGB and YCbCr descriptors also perform well on image pairs of Ubc image set. Differences between 2D and 3D samplings seem to be insignificant.

In Bark image set1, 3D-Color sampling (RGB-BRIEF, YCbCr-BRIEF, RGB-LATCH, YCbCr-LATCH) outperform 2D-Gray versions, significantly. In Bark1|3 and Bark1|4 pairs Gray-BRIEF could not catch any true matches, while RGB-BRIEF and YCbCr-BRIEF gave 0.4% and 0.2% of true matches, respectively.

In the rest two image sets, Leuven and Bikes, all types of descriptors give the similar performances as shown in 2.

2 Performances of Gray, RGB and YCbCr descriptors with respect to number of tests.

Effect of the number of binary comparisons on Gray, RGB and YCbCr descriptors (BRIEF) are shown in figs. 3 to 15. As shown in these figures, RGB-BRIEF and YCbCr-BRIEF descriptor performance continue to increase with the number of tests.

3 Effects of the patch sizes

In order to examine the consistency of our color descriptors, we looked for the effect of patch sizes. If all descriptors behave in a similar way, then it would strengthen the stability of proposed descriptors and it was as thought. All Gray, RGB and YCbCr descriptors (BRIEF) behave in a similar manner for each patch size as shown in 16.

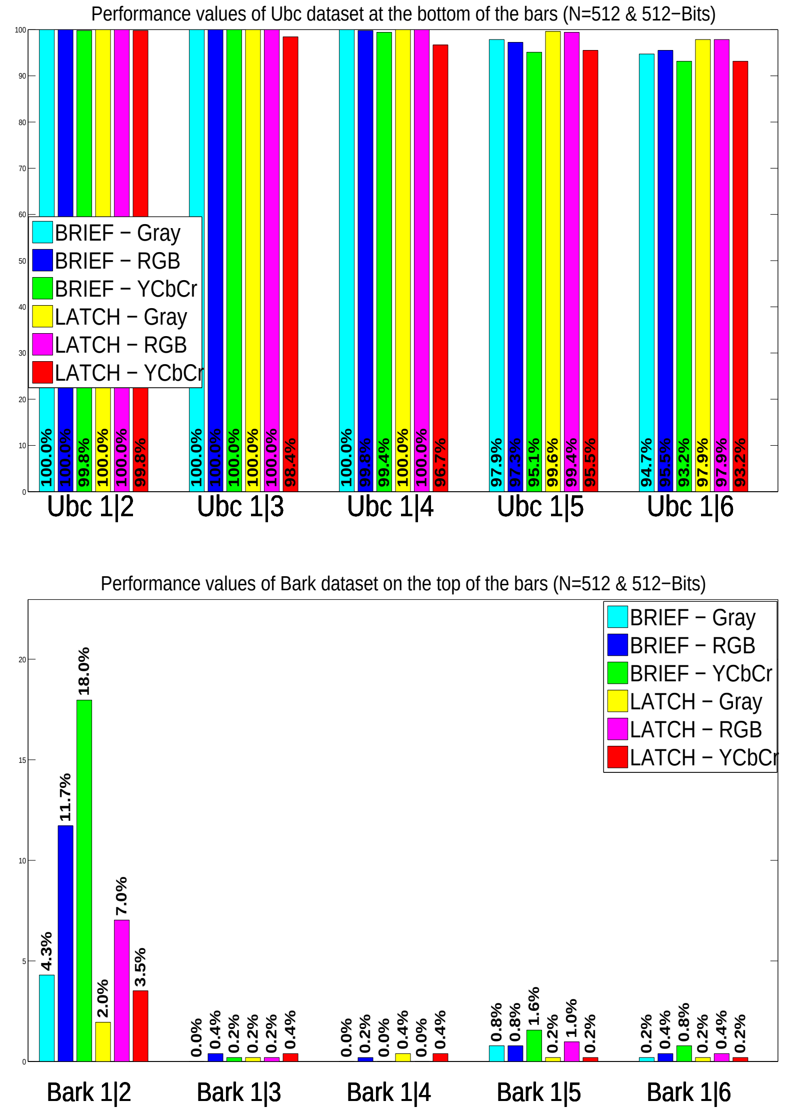


Figure 1: 2D and 3D Binary Comparisons in Different Color Spaces: Ubc and Bark imagesets. 512 Bits descriptors are used. N:Number of Keypoints to be matched.

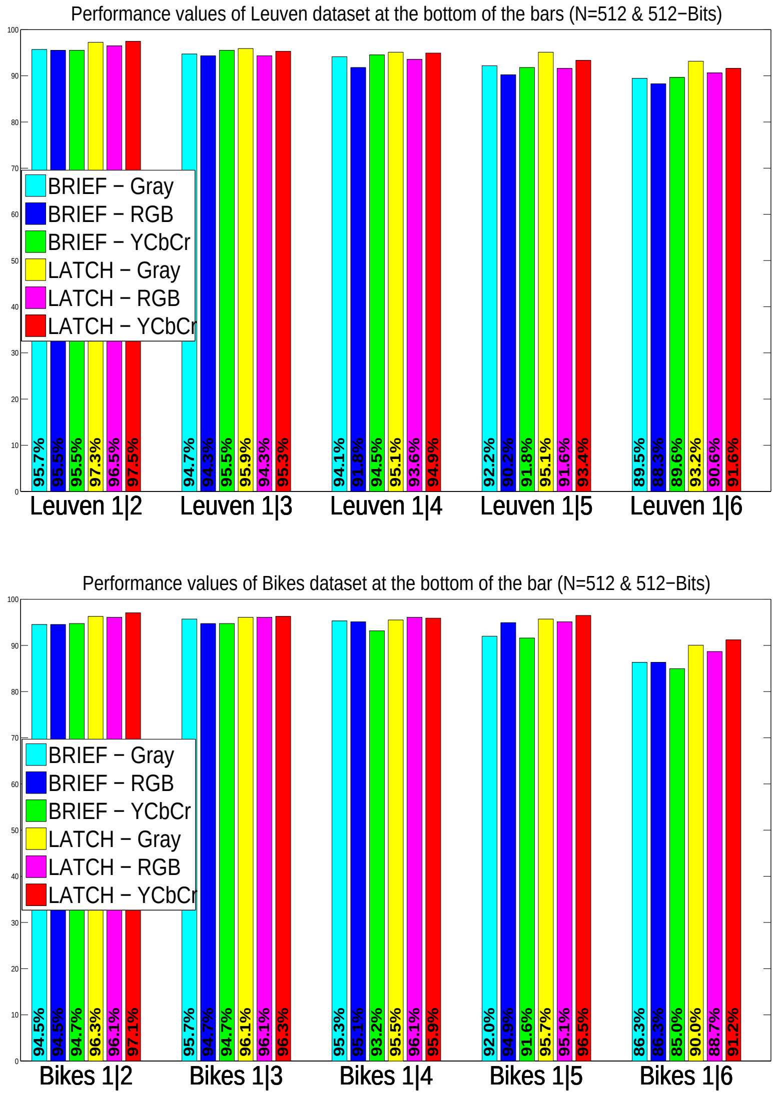


Figure 2: 2D and 3D Binary Comparisons in Different Color Spaces: Leuven and Bikes imageset. 512 Bits descriptors are used. N:Number of Keypoints to be matched.

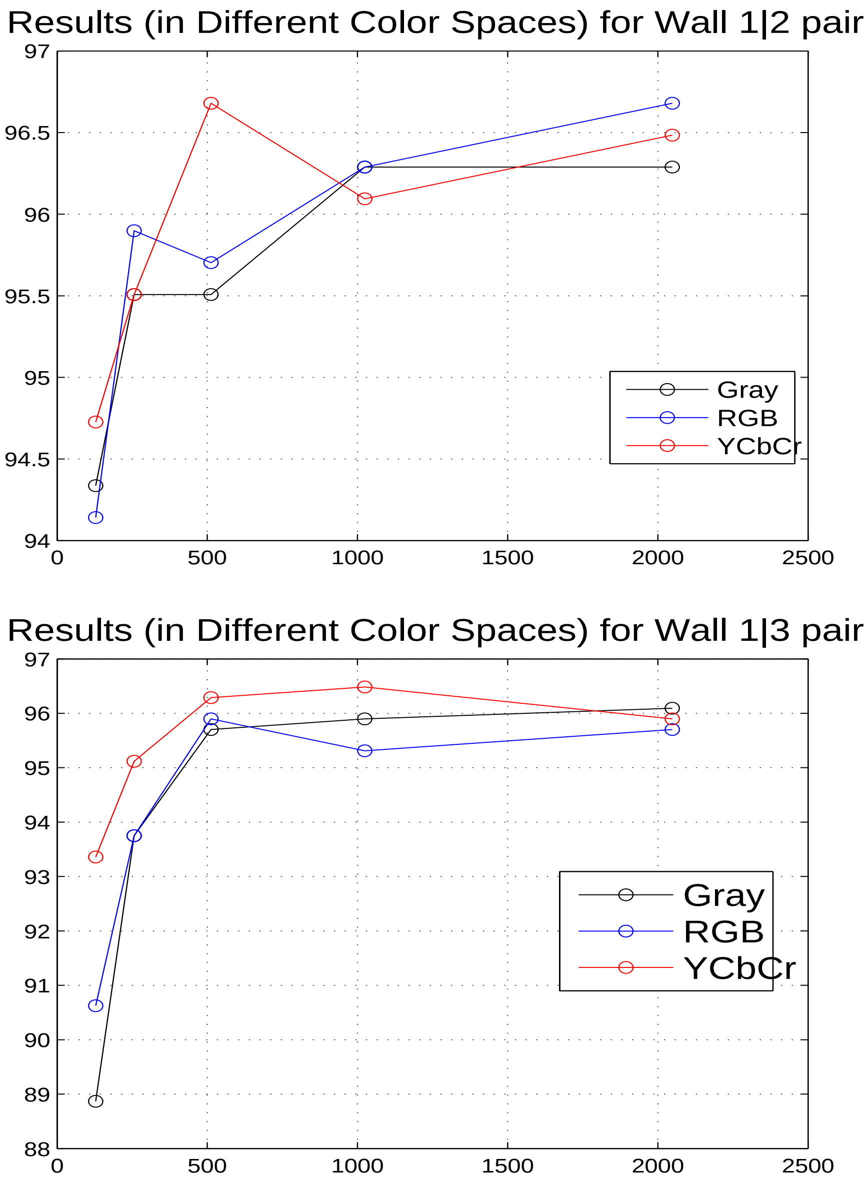


Figure 3: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

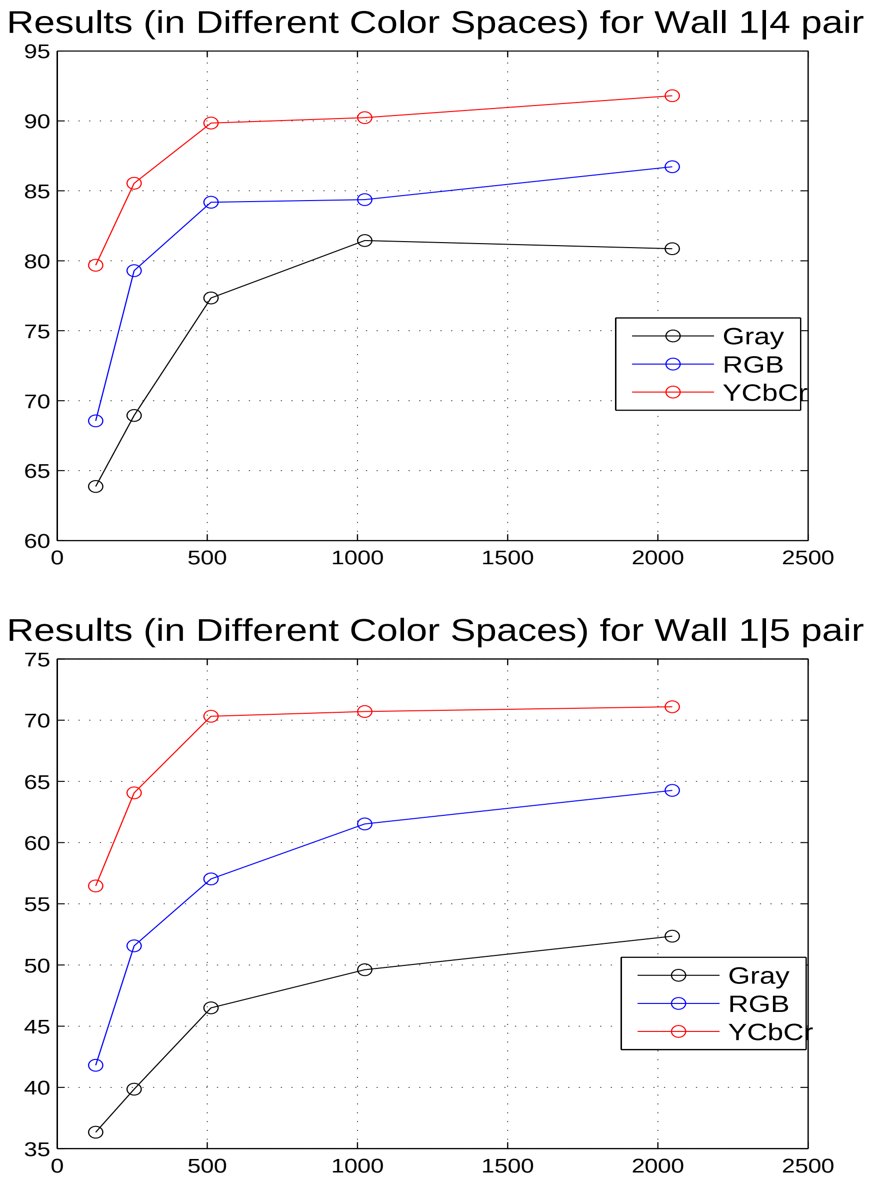


Figure 4: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

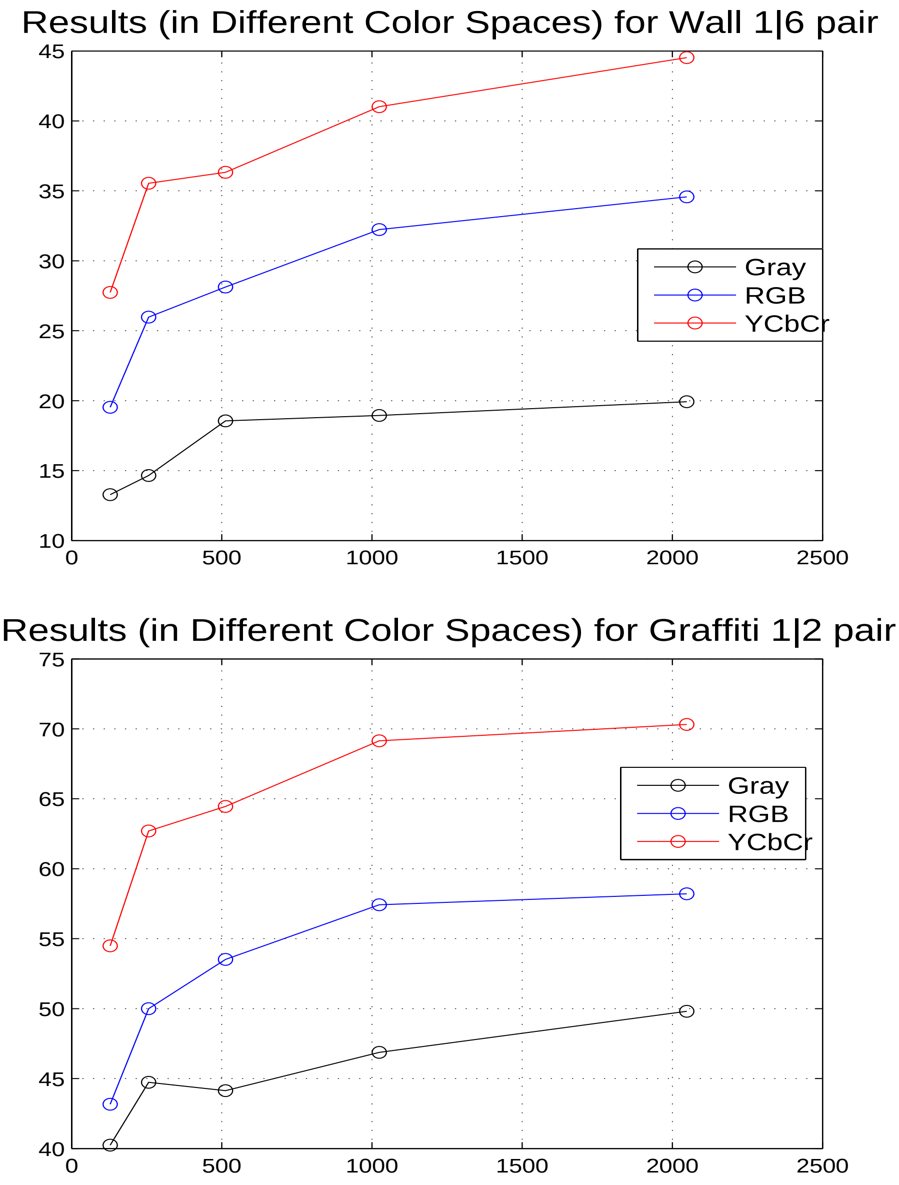


Figure 5: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

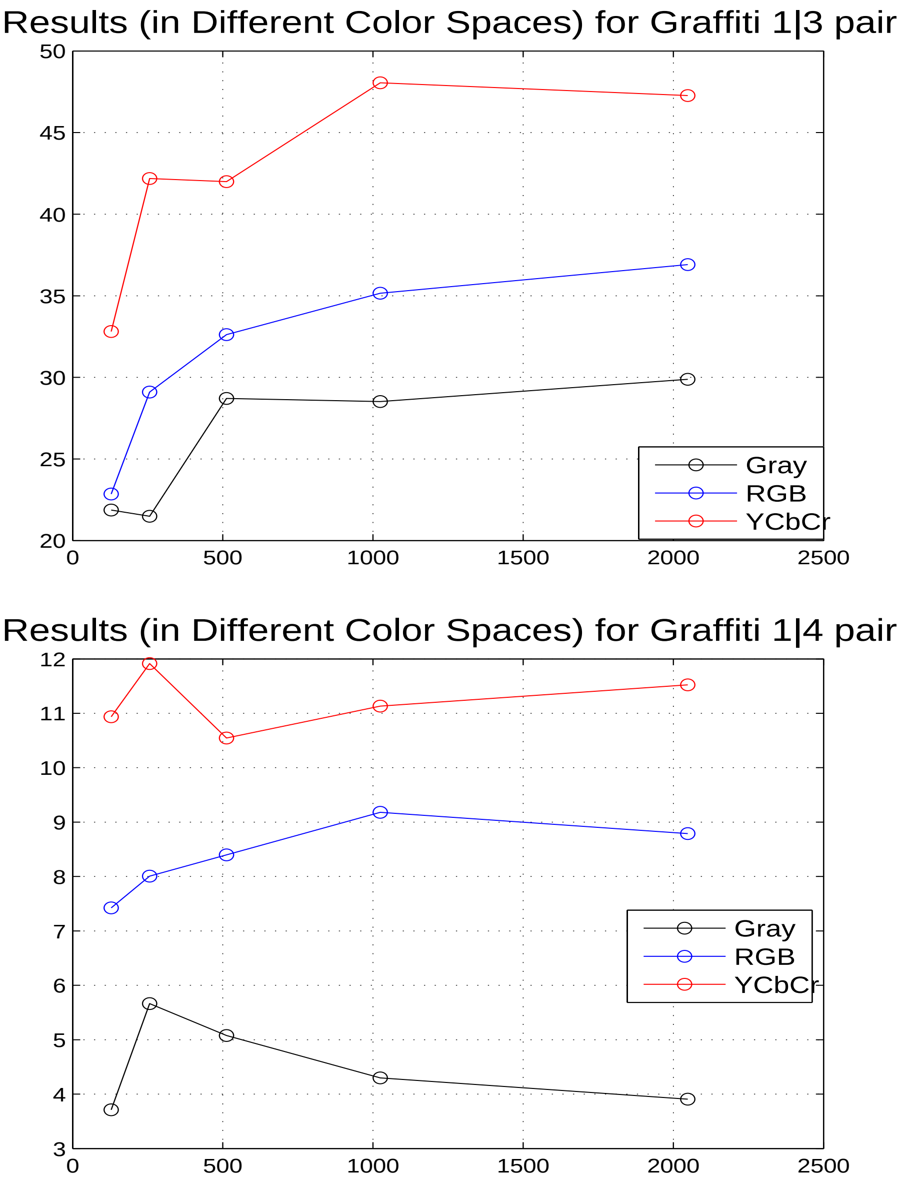
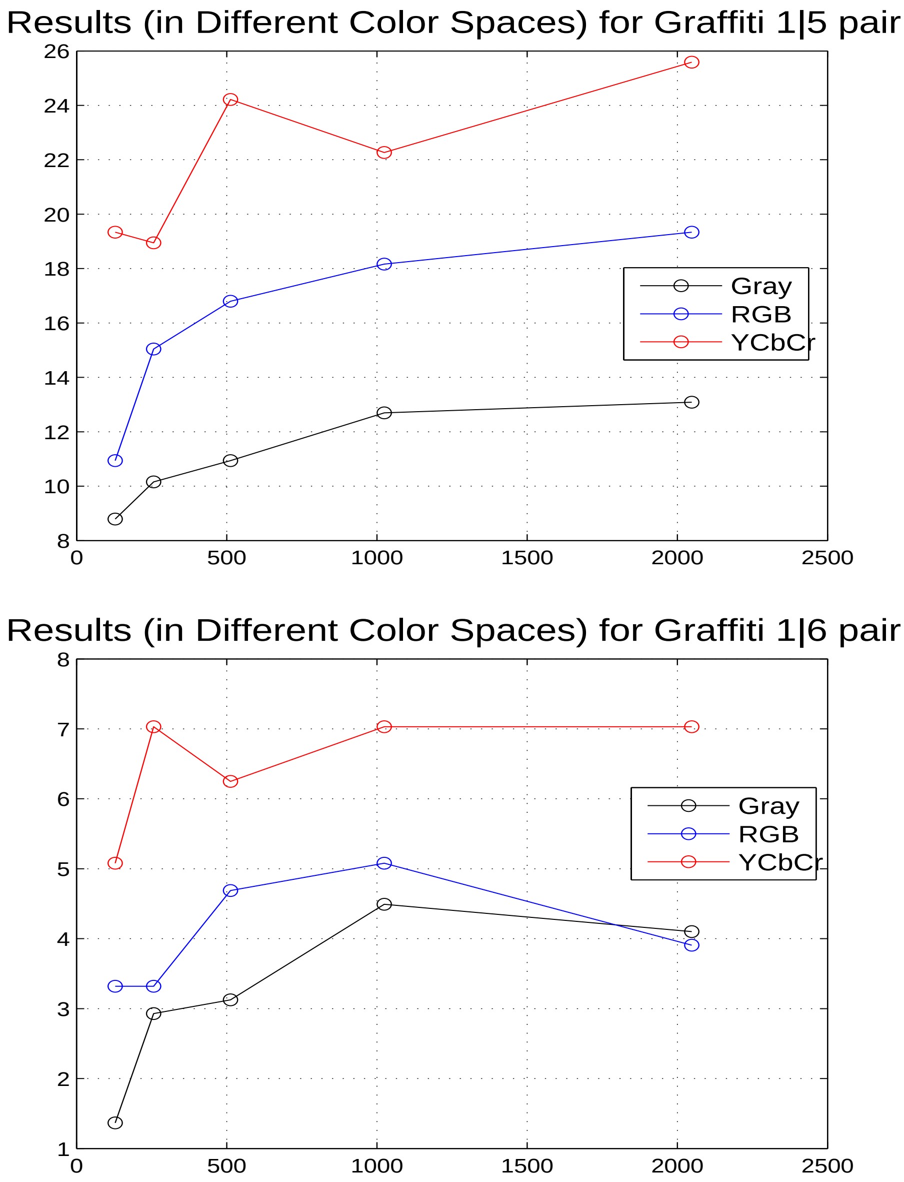


Figure 6: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.


Results (in Different Color Spaces) for Graffiti 1|5 pair

Results (in Different Color Spaces) for Graffiti 1|6 pair

Figure 7: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

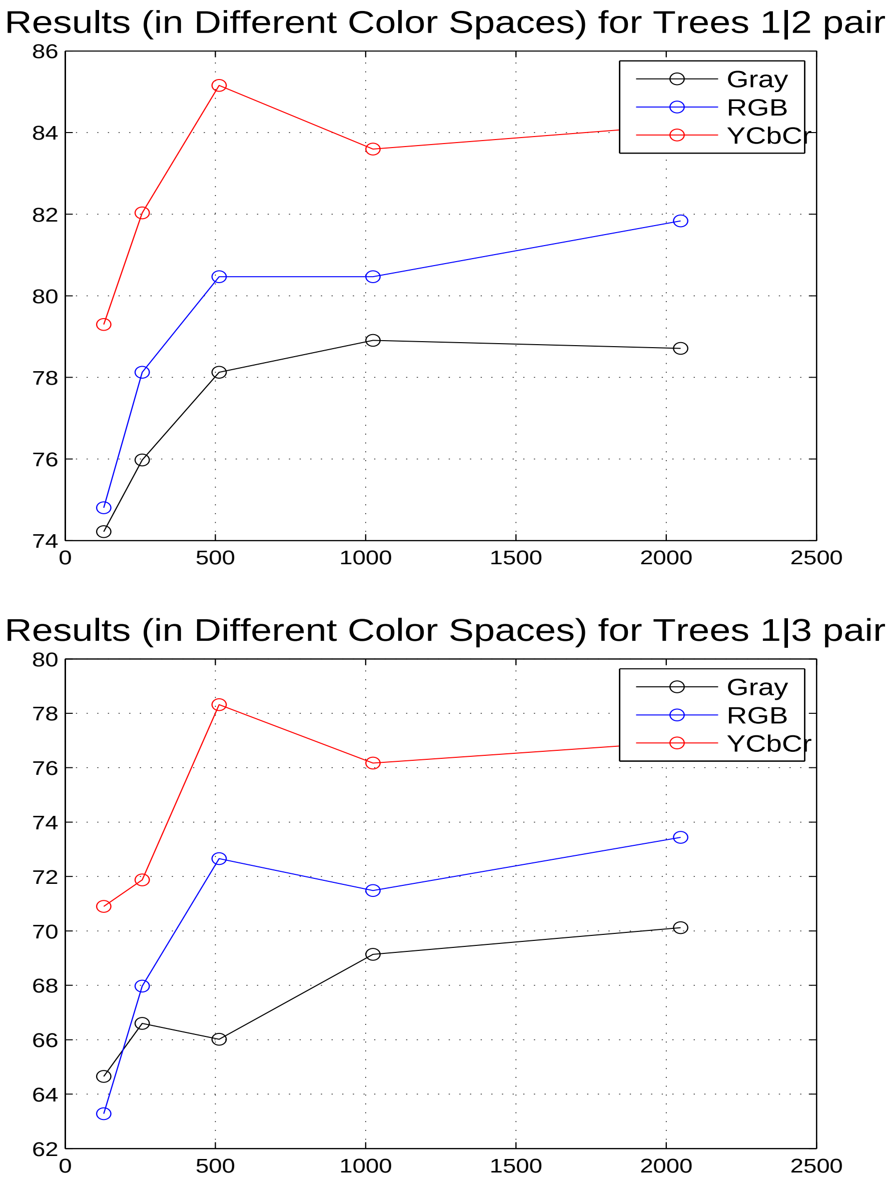


Figure 8: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

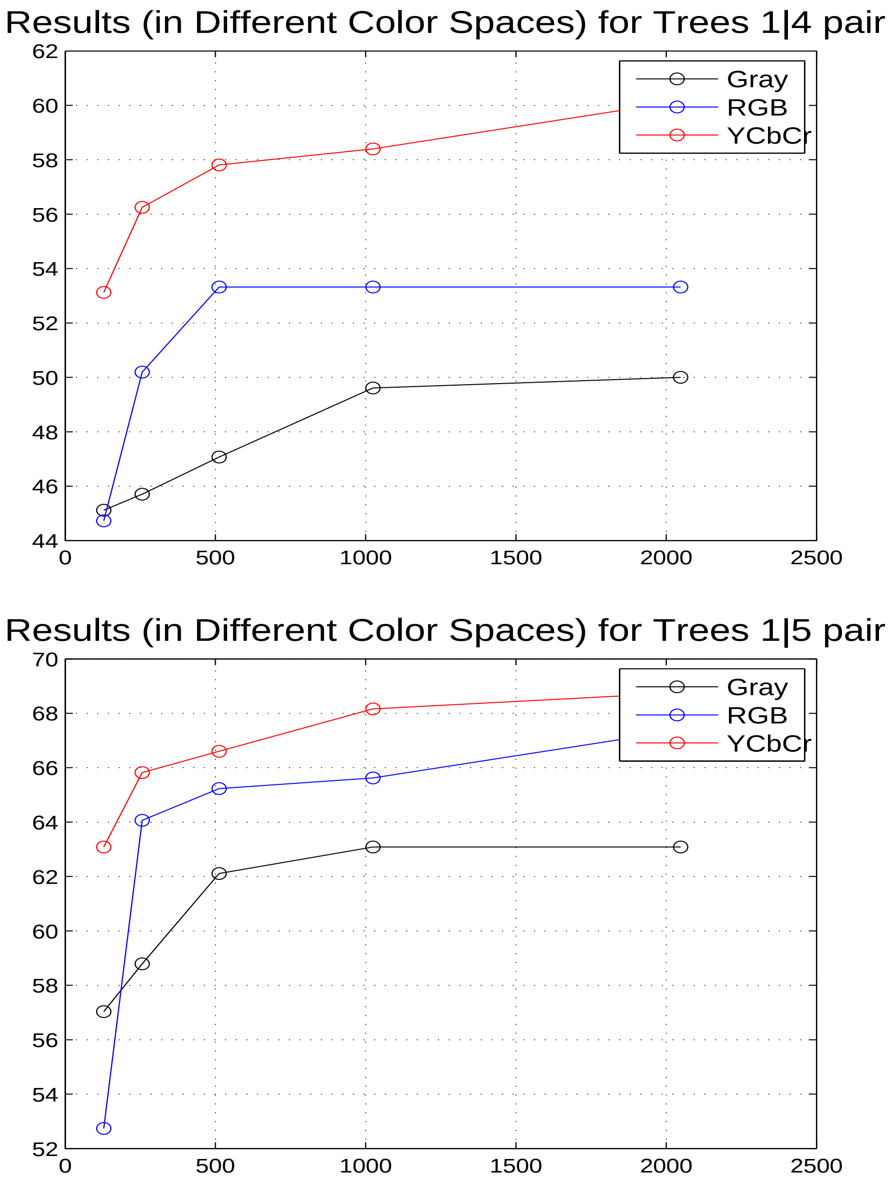


Figure 9: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

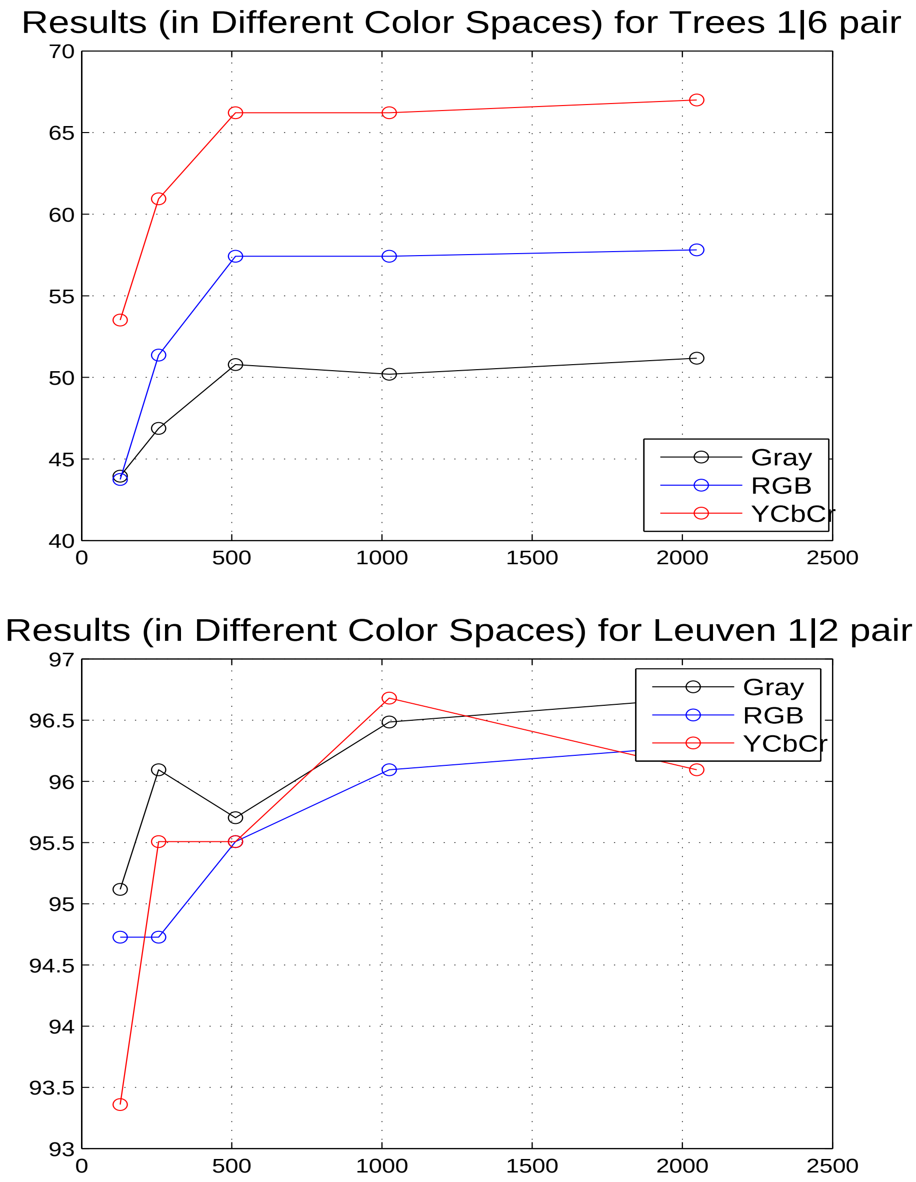


Figure 10: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

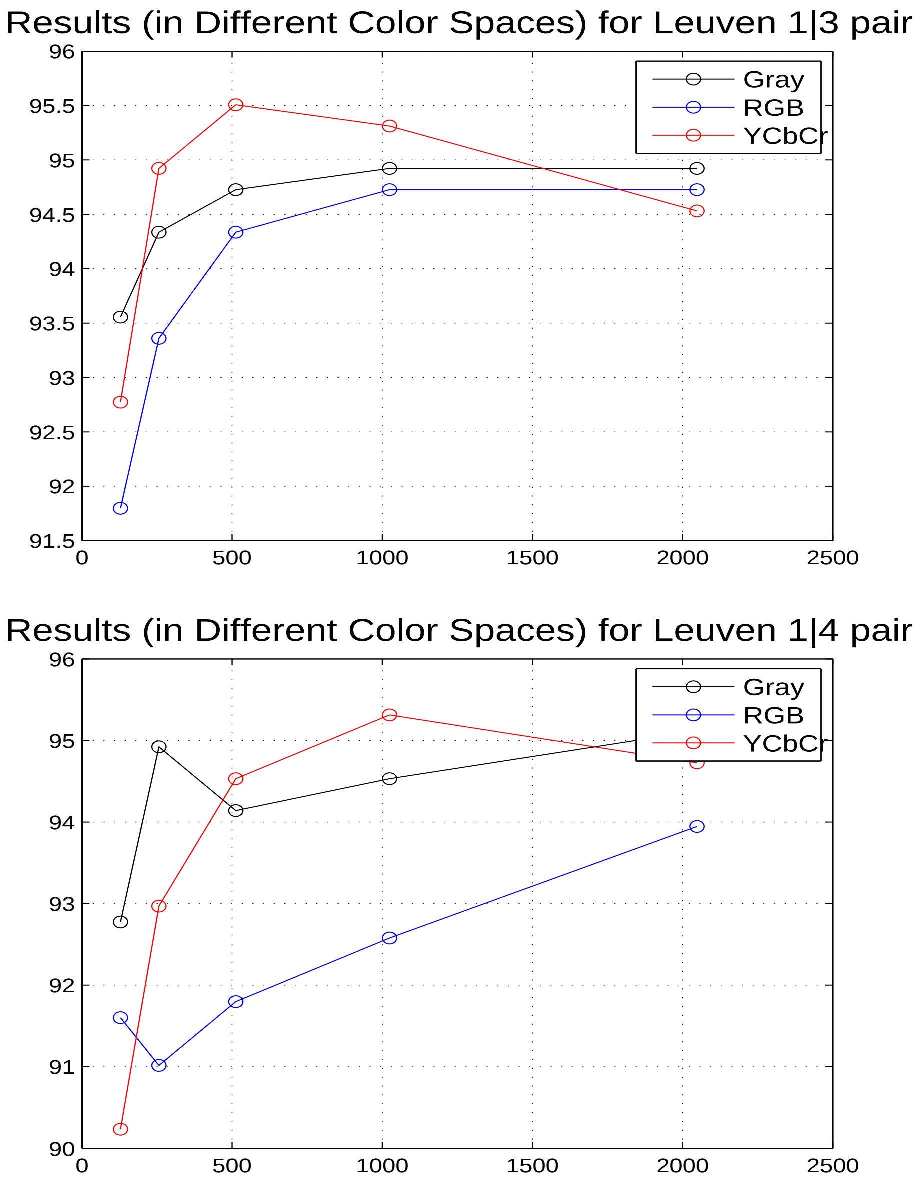
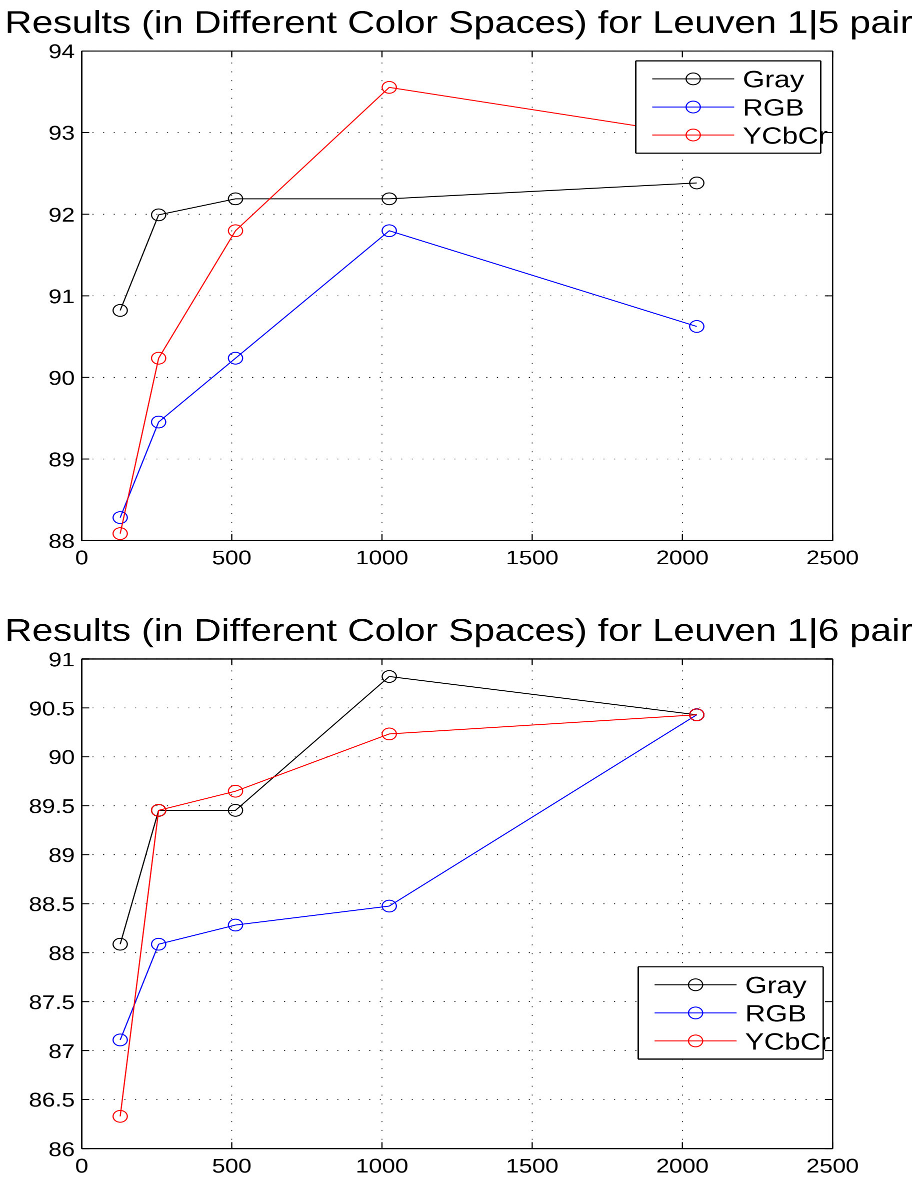


Figure 11: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.



Results (in Different Color Spaces) for Leuven 1|5 pair

Results (in Different Color Spaces) for Leuven 1|6 pair

Figure 12: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

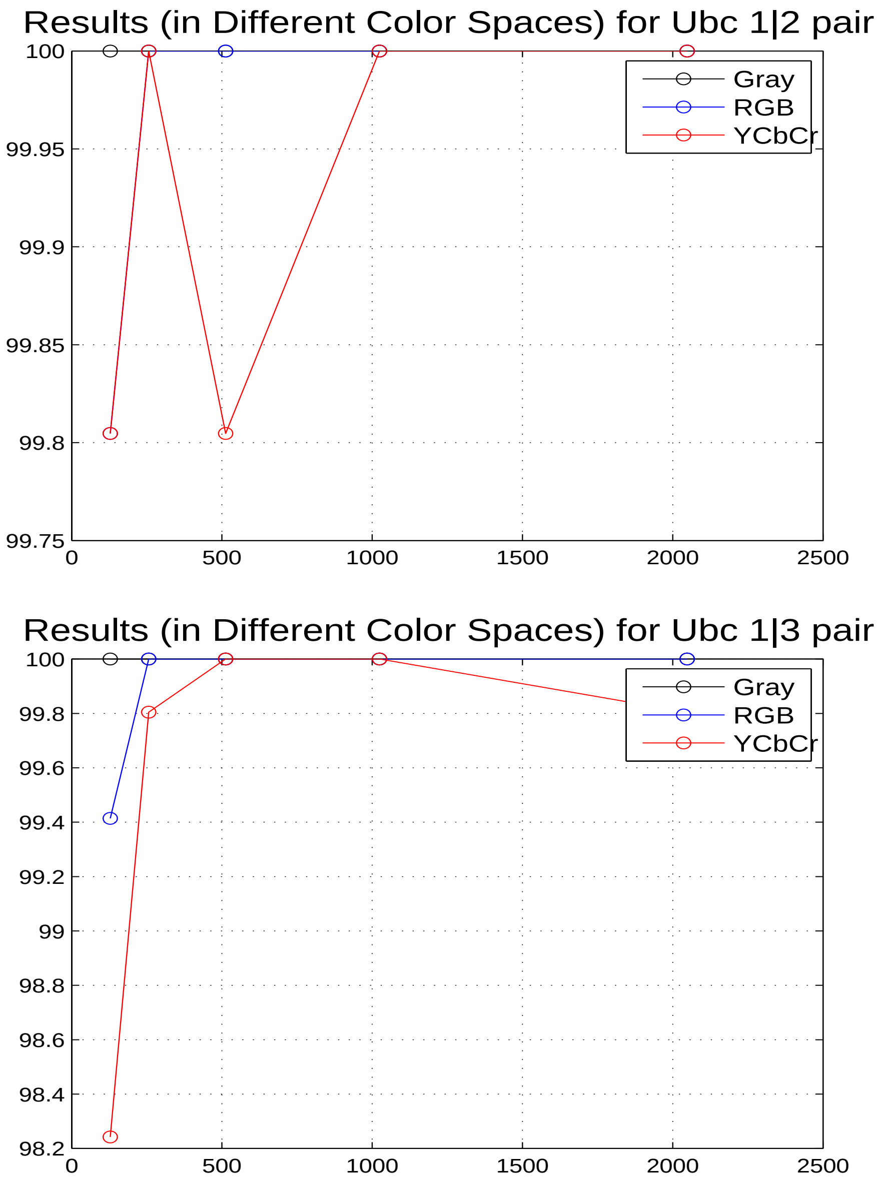


Figure 13: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

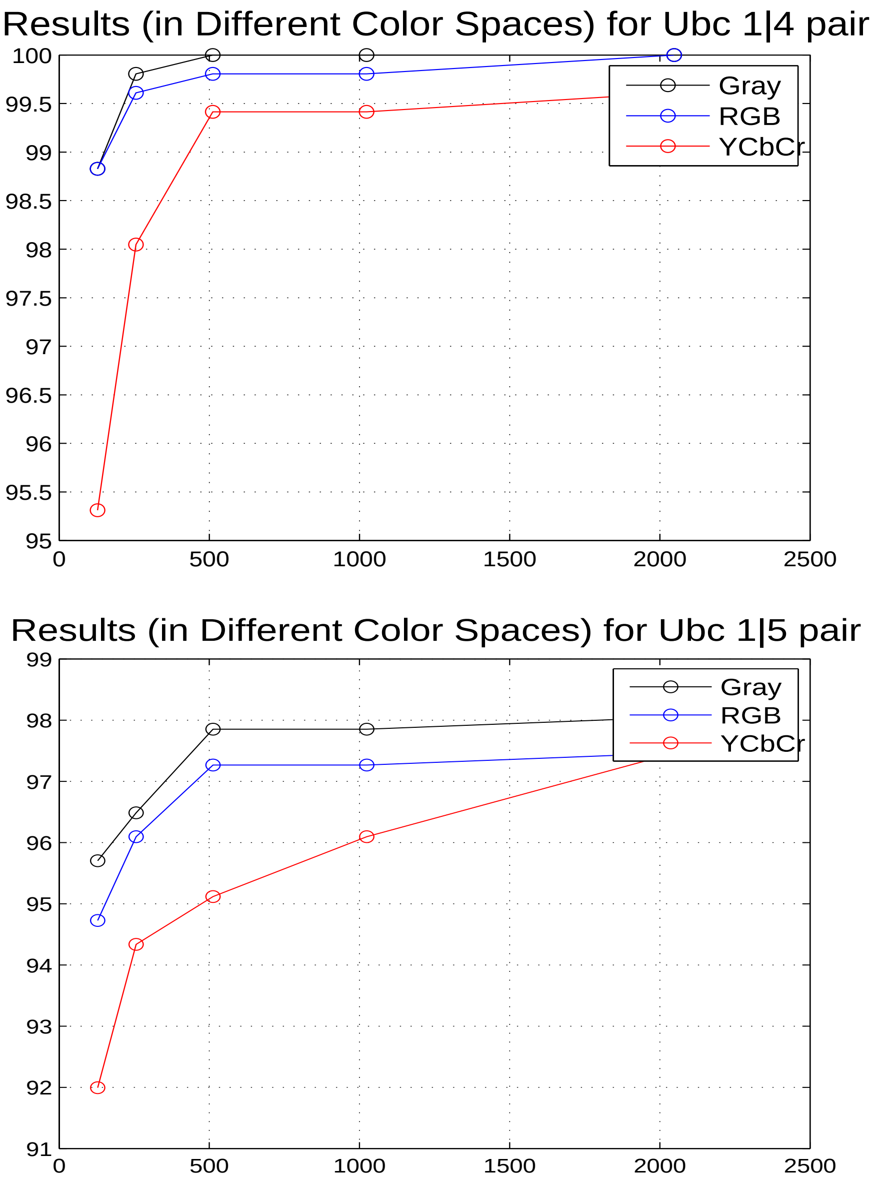


Figure 14: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.

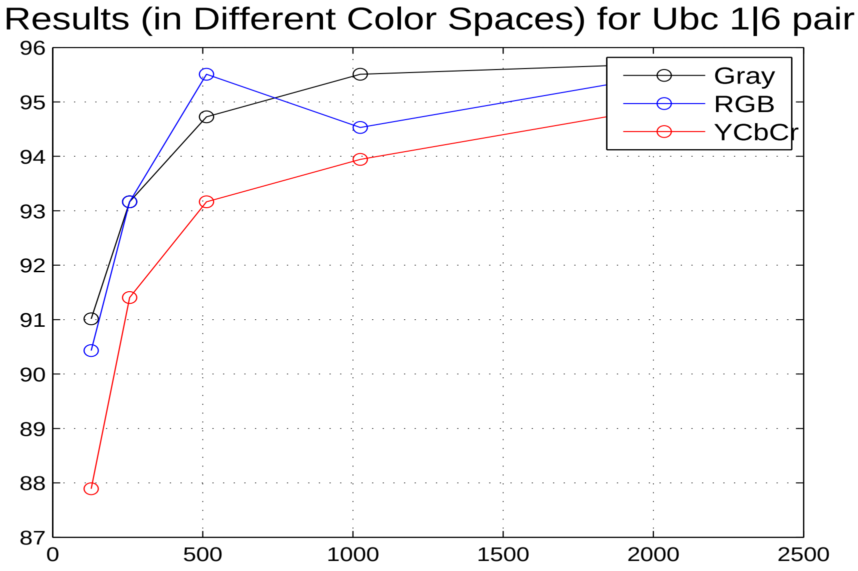
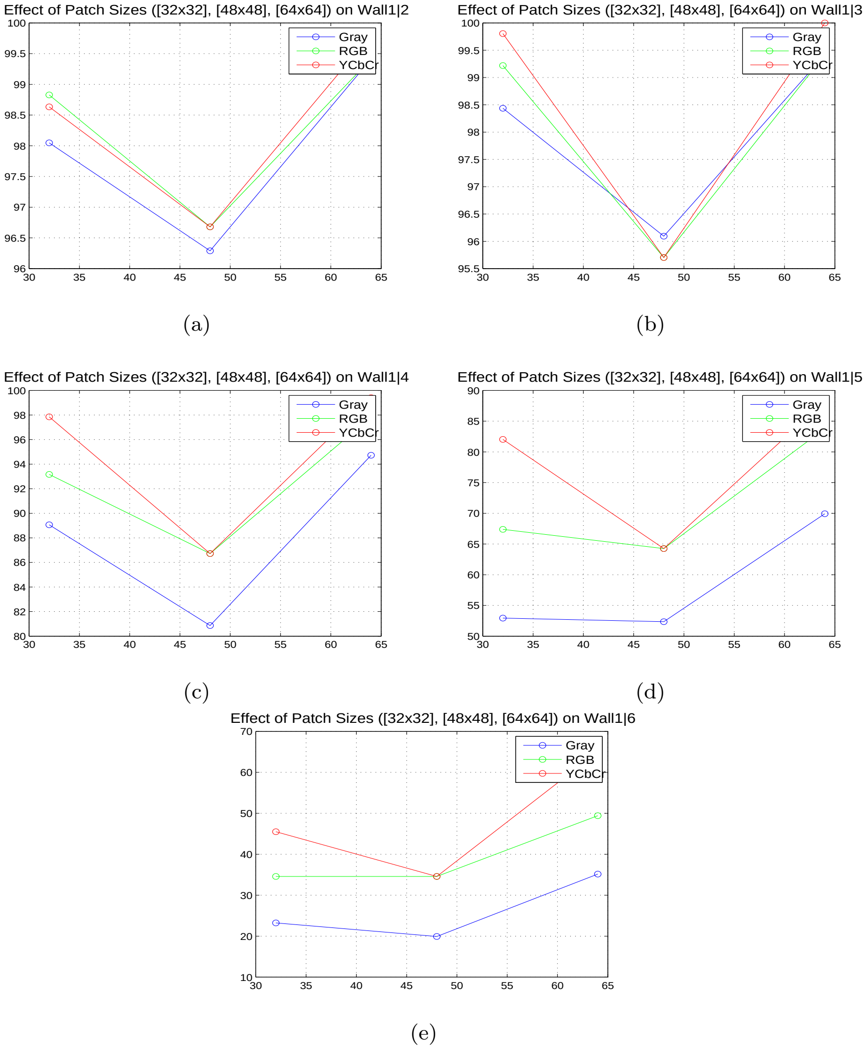


Figure 15: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.



(a)

(b)

(c)

(d)

(e)

Figure 16: Performances of Gray, RGB and YCbCr Samplings vs. Number of Samples.