Change Detection in Color Images

Abstract

One approach to change detection that is insensitive to illumination brightness and spectral changes uses properties of color imagery. This approach does not appear to have been previously investigated, and this paper asks: "What is the most appropriate metric for detecting changes in color imagery?". We define and compare six image difference functions, mainly based on RGB and HSV image representations. When the global illumination does not change radically, the normal Euclidean distance in the RGB or a modified HSV space works about equally well, when there is significant changes in image spectrum, then HSV works a bit better and if there is a large local change in the illumination, none of the investigated methods work well.

1 Introduction

Because illumination can vary over time in real world scenes, we would like change detection algorithms that are insensitive to the illumination change. In monochrome images, there is not much hope for pixel-based change detection algorithms that are invariant or insensitive to illumination changes, as the only signature of a scene change is an intensity change (which also occur as the illumination changes). Image content analysis is possible, as is the use of illumination invariant properties, such as monochrome image edge positions [1, 7]. On the other hand, real world scenes are generally colorful, and detection of scene changes based on changes in color imagery does not yet appear to have been investigated.

Jain and Chau [5] report results on using multisensor and multitime statistical data fusion for change detection, but did not exploit any special relationship between the difference sensors, which can be found when treating the different spectral bands in a color image as different sensors.

Because the color (albedo) of objects is largely separable from illumination and relative surface orientation (reflectance) effects (*e.g.* [4], pp 271), there is hope that we might be able to distinguish, in practice, image changes arising from scene changes (where the albedo would change) from those arising from illumination changes (where the albedo would not change). Research on color constancy obviously has a bearing on this problem, as elimination of all illumination effects would clearly leave only reflectance changes to consider. Unfortunately, this stream of research is not yet sufficiently numerically precise for change detection in complicated real-world scenes.

Here, we assume that we have a set of registered images, such as might arise from a single camera observing a potentially changing scene over time, perhaps for a security application. We will work at the pixel level, detecting pixels that have changed sufficiently. Recently, Rosin [6] has presented good results on the threshold selection process, concluding that one should model the spatial distribution of either the noise or signal. This leaves open the question of what should one threshold if one has color imagery, which is what is investigated here: **"What is the most appropriate metric for detecting changes in color imagery?"**.

We would like to have a change detection method that is insensitive to:

1. uniform illumination brightness changes - *e.g.* due to changes in light source brightness, camera aperture or sensitivity
2. uniform illumination spectral changes - *e.g.* due to changes in light source spectral composition or camera spectral sensitivity
3. local illumination brightness changes - *e.g.* due to shadows or nearby light sources.

The first two criteria motivate the spectral and intensity correction process given in the next section, and the local illumination change criteria motivate investigation of some distance functions that do not depend on the absolute pixel brightness, as discussed in Section 3. Comparing the results from different distance functions is never easy, but using Receiver-Operating-Characteristic curves, as shown in Section 4, makes it possible to avoid having to choose exactly comparable thresholds.

2 Pre-processing

Two global processes are applied before the image difference is calculated.

The first process identifies pixels where the distance function is probably invalid, and which are therefore not used in the distance calculation. The two cases are when the pixel is very bright or dark. In the bright case, the camera saturates, and so the proper relationship between the RGB components at a pixel cannot be determined. This often occurs at specularities (which can be detected [2]), but also in generally bright areas. In either case, the pixel values are unusable. So, we create a boolean mask: $U(i,j)$ having value 'true' when one or more of the RGB spectral values is greater than 250. In the dark case, the camera sensitivity is not good due to noise and quantization, and small changes in pixel RGB values can have exaggerated importance in the pixel distance result. So, we create second binary mask: $L(i,j)$ having value 'true' when the length of the RGB vector is smaller than a given threshold (50, chosen by trial and error). Together, these two masks are used to control when the pixel distances are calculated:

```
U(i,j) or U'(i,j) or [ L(i,j) and L'(i,j) ]
then distance = 0
else use distance function
```

In other words, don't calculate distance if one pixel has saturated or both pixels are dark.

The second process removes the global spectral and intensity change in illumination, which should correct for changes in the brightness or spectrum of the illumination (*e.g.* colored lighting). The technique works because surface brightness is proportional to the product of reflectance and illumination. Suppose that the initial image's pixels had RGB value (r_i, g_i, b_i). After a global change in illumination their values become $(\lambda_r r_i, \lambda_g g_i, \lambda_b b_i) = (r'_i, g'_i, b'_i)$. Then, we can estimate

$$\lambda_r = \frac{\sum_i r_i}{\sum_i r'_i}$$

and similarly for λ_g and λ_b. The mean function was used because it is assumed that the number of pixels changing is a small percentage of the total. (If an application did not have this property, another global illumination adjustment process could be found.) A transformed second image is calculated: $(r'_i/\lambda_r, g'_i/\lambda_g, b'_i/\lambda_b)$. This aims to produce corresponding pixels with similar intensity.

3 Distance Functions

We looked at two RGB space and three HSV space distance measures, which are described now. In all cases, we only look at pixel-wise differences.

3.1 Euclidean Distance of Raw RGB

We treat each pixel as a 3-vector $\vec{r}$ of the (red, green, blue) components on a [0..255] scale. The distance between pixels is $|| \vec{r} - \vec{r}' ||$.

3.2 Dot Product of Unit RGB Vectors

Each pixel is normalized $\vec{r}_u = \frac{\vec{r}}{||\vec{r}||}$ with the (red, green, blue) components on a [0..1] scale. The distance between pixels is: $1 - \vec{r}_u \cdot \vec{r}'_u$. Pixels with a similar color distribution will have a similar vector and thus a small distance.

Normalizing the raw RGB should remove residual uniform illumination intensity differences. We assume that any illumination-based color changes will be small.

3.3 Euclidean Distance of HSV

To make the change detection invariant to changes in illumination strength, we could use a representation in which the intensity of the light is explicit and separated from the color. One representation is HSV or hue, saturation, value ([3], pp 590).

Each raw RGB pixel is transformed into its corresponding HSV representation
$(h, s, v) = rgb2hsv(r, g, b)$, with $0 \leq h \leq 1$, $0 \leq s \leq 1$, $0 \leq v \leq 1$ ([3], pp 590). Note that h wraps around at 1 back to 0. Also, when $s = 0$, the h value is undefined, and, in practice, when s is small, then h becomes unstable.

Taking account of this instability and wrap-around problem, we transform the HSV encoding into a vector in the HSV color hexcone. A HSV vector (h, s, v) becomes $(vs\cos(2\pi h), vs\sin(2\pi h), v)$, so the distance between two HSV values (h, s, v) and (h', s', v') is: $|| (vs\cos(2\pi h), vs\sin(2\pi h), v) - (v's'\cos(2\pi h'), v's'\sin(2\pi h'), v') ||$.

3.4 Euclidean Distance of HS

A second HSV distance function uses just the H and S components, to make the distance independent of the brightness: $|| (s\cos(2\pi h), s\sin(2\pi h)) - (s'\cos(2\pi h'), s'\sin(2\pi h')) ||$.

3.5 Euclidean Distance of H

Another HSV distance function use s just the Hue component: $| h - h' |$.

3.6 Combined HS *versus* V

The final metric tries to select the best of either the V component when there is a big change in the pixel brightness or HS when there is a big change in value: $max(|| (s\cos(2\pi h), s\sin(2\pi h)) - (s'\cos(2\pi h'), s'\sin(2\pi h')) ||, | v - v' |)$.

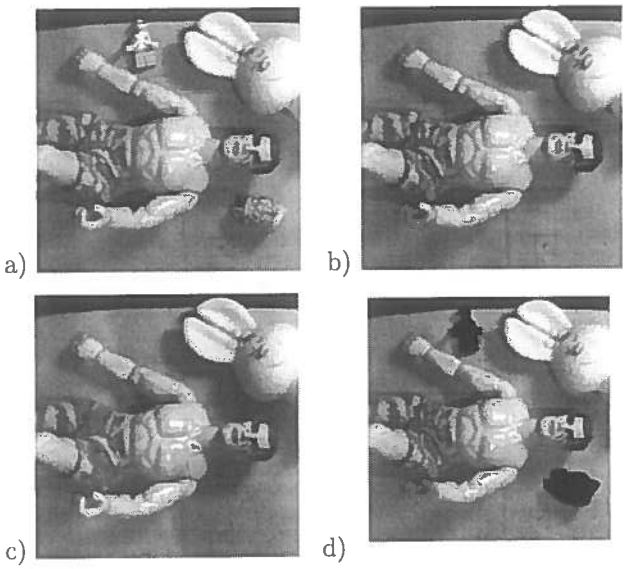

Figure 1: Some experiment 1 test images: a) original image (F), b) some objects removed (A), c) some objects removed plus some moved local light source and halogen light turned off (G), d) ground truth mask used in ROC calculation.

4 Experiments

To detect a change, most approaches threshold the difference measures, and the choice of threshold value is critical, because this affects how many changed pixels are missed and how many unchanged pixels are falsely detected. Comparing the image difference measures is difficult because each measure rates changes differently with changes in threshold. How this threshold should be chosen is a complex issue, although a recent paper [6] presents some good suggestions. Comparing results at a single threshold choice, even if a method of selecting comparable thresholds were possible, may also not give a good comparison (*e.g.* due to an unlucky threshold choice).

To avoid the issue of comparing thresholds, we compare the distance measures using a "Receiver-Operating-Characteristic" (ROC) methodology: for each threshold value in a range, calculate the percentage P_{fn} of changed pixels that were not detected and P_{fp} of unchanged pixel classified as changed. Then plot the curve of (P_{fp},P_{fn}) values as the threshold changes. This will show the range of performances possible, without having to directly select comparable thresholds. As we shall see below, this will give us a satisfactory tool for comparison.

We need ground truth to calculate P_{fn} and P_{fp}. The test images were generated under a variety of different illumination conditions, but with a known change to the scene - the removal of one or two objects. By hand, we generated a mask covering the changed region(s), with mask(i,j) = 1 if pixel (i,j) was changed. Let change(i,j) = 1 if the change detection algorithm decided that pixel (i,j) changed. Then:

$$P_{fn} = \frac{\sum_{(i,j)} mask(i,j)(1 - change(i,j))}{\sum_{(i,j)} mask(i,j)}$$

$$P_{fp} = \frac{\sum_{(i,j)} (1 - mask(i,j))change(i,j)}{\sum_{(i,j)} (1 - mask(i,j))}$$

To experiment with different issues, we set up a variety of scene changes:

- additional light sources
- moved light sources
- partial or complete shadows
- minor change to illumination spectrum

and, of course, removing the target objects. In all cases, the aperture and focus of the camera remained the same and the AGC was disabled.

We show here the results for 5 experiments each for two test scenes. The first scene (Figure 1a) has a flat ground plane with several colored 3D test objects, which causes shadow positions to move as the light source is moved. The second scene (Figure 3a) is largely all planar objects in a common plane.

The conditions varied in the first test scene are summarized in this table, with OBJ: extra object present, HFS: nearby halogen light fully shadowed with translucent sheet of white paper, HPS: nearby halogen light partly shadowed with translucent sheet of white paper, FIL: halogen light red filtered and TNL: nearby tungsten light moved to the left. The overhead and nearby halogen lights were always on and there was a nearby tungsten light initially at the right.

IMG	OBJ	HFS	HPS	FIL	TNL
1F	Y				
1A					
1C		Y			
1D			Y		
1E				Y	
1G					Y

Test images 1F, 1A and 1G are shown in Figure 1.

Three of the ROC curves are shown in Figure 4. In these curves, the smaller thresholds are at the bottom-right portion of the curves increasing to a maximum at the upper left. The percent of falsely detected pixels was only displayed to 10% because this was the most important region of the curve. The ideal curve would

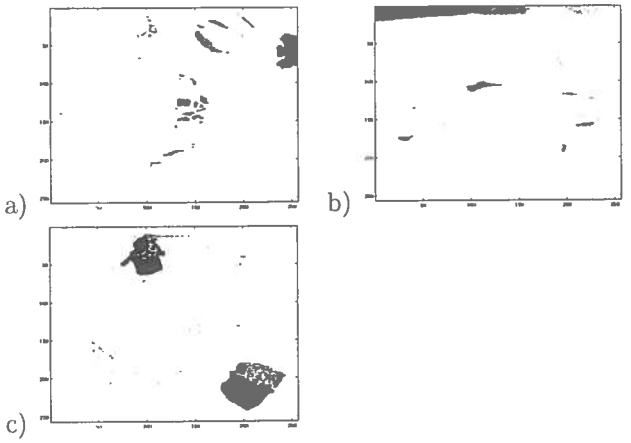

Figure 2: Experiment 1 results part 2: a) mask of too bright pixels, b) mask of too dark pixels, c) results of thresholding (at 32.4) RGB distance between images F and A.

pass very close to (0,0), meaning that it was possible to achieve very few false detections while simultaneously missing few object pixels. Figure 4a shows the curve for when only the target objects were removed. This plot shows that the RGB distance measure ("*") is comparable to the HSV measure ("x") at low false detection rates, with HS slightly better at very low rates. If you can tolerate about 1% falsely detected pixels (*e.g.* about 600), then you detect about 35% of the changed pixels. Further, as the RGB and HSV curves are generally closest to the origin, then these distance measures are better, and at higher percentages of falsely detected pixels, the RGB measure is more sensitive at detecting true changed pixels. The RGB distance between images 1F and 1A when thresholded at 32.4 is shown in Figure 2c. At this threshold, the $P_{fn} = 25\%$ and $P_{fp} = 2.6\%$.

In test images 1C, 1D and 1E, a similar relationship between the RGB and HSV measures holds, although they are nearly identical in performance, except in image 1D (which is shown in Figure 4b). Here, the halogen light source is partly shadowed and HSV is slightly better at higher P_{fa}.

In test image 1G, the nearby tungsten light source is placed at the left and the halogen light is turned off, causing considerable change to the shadow positions (Specularities are already masked out.) In this case, none of the algorithms perform well (see Figure 4c), with the HSV and Dot measures doing the least badly. We do not have a substantiated hypothesis about why the performance should be so bad, but some possibilities are: a) there is a strong illumination gradient here, so the global normalization described in Section 2 will not produce comparable RGB or V values, b) the increased brightness in image G was non-linearly compressed by the camera's gamma compression and c) the tungsten light source has a much redder color spectrum applied locally rather than globally because it is quite close to the scene.

Generally, the H and HS distance measures were worse than the HSV measure (except with the strong local illumination gradient), with H always worse than HS. The HSvsV measure was generally better than H and HS (except when the light spectrum changed), but was always worse than or equaled by RGB and HSV.

Figure 1d) shows the ground-truth mask used in the ROC curve calculation and parts b) and c) shows the pixels marked as too bright and too dark in the original image 1F.

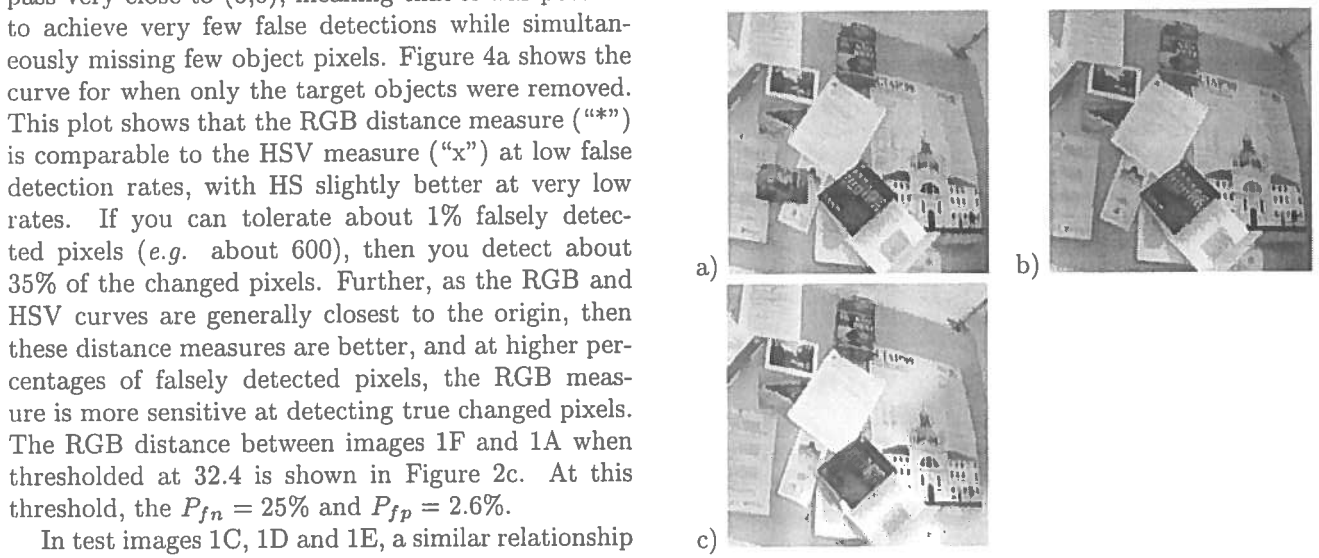

Figure 3: Some experiment 2 test images: a) original image 2A, b) image 2D with an object removed plus a red filter on the halogen light source and c) image 2G with an object removed plus table light on, halogen off and a strong shadow.

The conditions varied in the second test scene are summarized in this table, with OBJ: extra object present, OVH: overhead fluorescent lights on, HAL: nearby halogen light on, TAB: nearby tungsten incandescent light on, FIL: halogen light red filtered and SHA: partial shadow cast by table lamp.

IMG	OBJ	OVH	HAL	TAB	FIL	SHA
2A	Y	Y	Y			
2C		Y	Y			
2D		Y	Y		Y	
2E		Y	Y	Y		
2F			Y	Y		
2G		Y		Y		Y

Test images 2A, 2D and 2G are shown in Figure 3. In the results below, the H metric is always worst, the HSvsV metric is always worse than or equal to the RGB or HSV metric and the Dot metric is always worse than the HS metric. So, these metrics are not shown.

In test scene 2, when the test object is removed and there is a no change (image 2C) or only a slight change to the illumination spectrum (images 2D and 2E), the ROC curves for change detection are all quite similar. Figure 5a) shows the ROC curve for test image 2D where the red filter was used. At low P_{fp} the RGB and HSV measures are nearly identical, but at higher P_{fp} the HSV measure is better. In test image 2C, the ROC and HSV curves are nearly identical over the whole range.

Test image 2F is like test image 2E except that the overhead lights are turned off. Here, the ROC results (not shown) are poor, although slightly better than those of Scene 1 image 1G, with the best result by the Dot and HS distance measures (which is more predictable as these measures are independent of absolute intensity). In this case, the absolute brightness is quite different and the illumination gradient is also quite different, so presumably the explanations for poor results in Scene 1 image G also apply here.

Test image 2G (ROC curve shown in 5b) has some interesting behavior: at very low percentages of falsely detected changes, the RGB measure has the best performance, but at slightly larger values, the HS is the best. The hump in the HSV is due to the strong illumination gradient at the lower right. In the case of scene 2, the removed object has a higher contrast difference with the background and so, with the RGB measure, the contrast of the removed object is larger than the contrast with the illumination for some object pixels. This allows a threshold to select only object pixels.

5 Discussion and Conclusions

The main conclusion is that when the global illumination does not change radically, the normal Euclidean distance in the RGB or a modified HSV (Hue, Saturation, Value) space works about equally well. When there is significant changes in image spectrum, then HSV works a bit better. Finally, if there is a large local change in the illumination, none of the investigated methods work well, but using the H and S components tends to work slightly better.

It is interesting that the intensity independent distance measures (*e.g.* Dot, HS, H) did not work as well. Examining the data suggests that the reason for this is that brightness changes associated with a scene change are as important as spectral changes, and these should be exploited when possible.

The main problems arise where there is a significant change to local illumination spectrum, such as when a light source with a different spectral composition is nearby.

Acknowledgements

References

[1] M. Bichsel. Segmenting simply connected moving objects in a static scene. IEEE Trans PAMI 16, pp 1138–1142, 1994.

[2] G. Brelstaff and A. Blake. Detecting specular reflections using Lambertian constraints. Proc. ICCV, pp 297-302, 1988.

[3] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes. Computer graphics : principles and practice. (2nd ed. in C), Addison-Wesley, 1996.

[4] R. Jain, R. Kasturi, B. G. Schunck. Machine Vision. McGraw-Hill, 1995.

[5] Z-S. Jain, Y. A. Chau. Optimum Multisensor Data Fusion for Image Change Detection. IEEE Trans. Sys. Man. and Cyb, 25(9) pp 1340–1347, 1995.

[6] P. L. Rosin. Thresholding for Change Detection. Proc. 6th ICCV, Bombay, pp 274–279, 1998.

[7] K. Skifstad, R. Jain. Illumination independent change detection for real world image sequences. CVGIP, 46, pp 387–399, 1989.

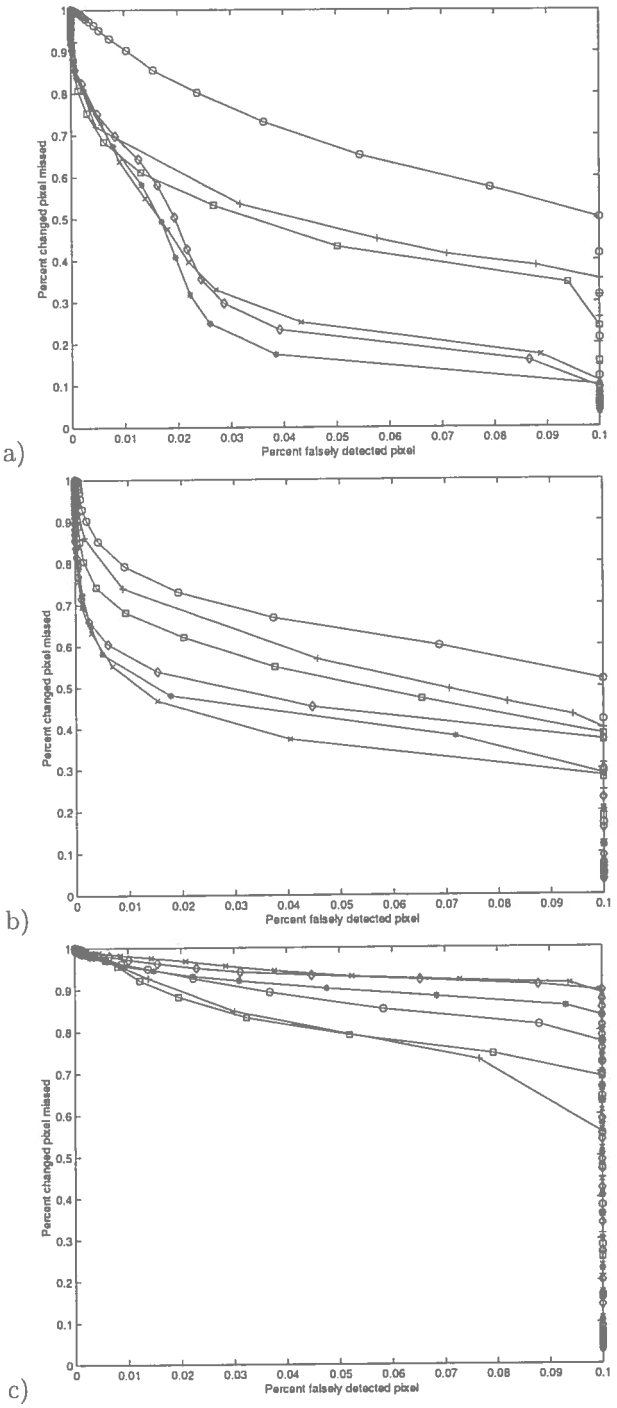

Figure 4: Experiment 1 results part 1: a) ROC curve of image 1A, b) ROC curve of image 1D and c) ROC curve of image 1G. The horizontal axis is the percent of false positives and the vertical axis is the percent of false negatives. The curves are labeled as: '+': dot product, '*': RGB distance, 'o': H distance, diamond: HSvsV distance, square: HS distance and 'x': HSV distance.

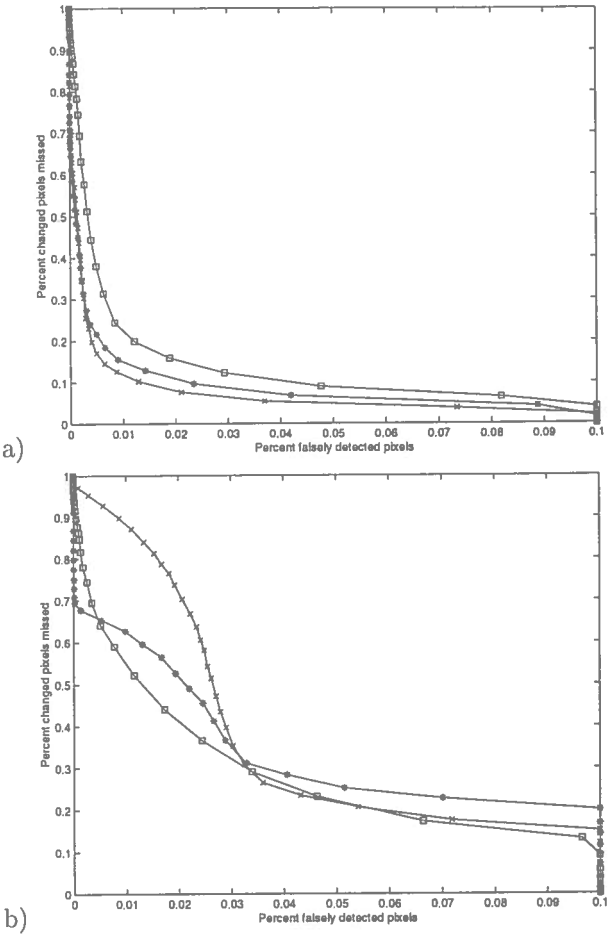

Figure 5: Experiment 2 results part 1: a) ROC curve of image D and b) ROC curve of image G. The horizontal axis is the percent of false positives and the vertical axis is the percent of false negatives. The curves are labeled as: '*': RGB distance, square: HS distance, and 'x': HSV distance.