Color Cluster Rotation

D. Paulus[1] and L. Csink[2] and H. Niemann[1]

Accepted for ICIP 98

Contents

File	`/home/paulus/text/papers/KEPAF-97/RCS/icip98.tex,v`
Version	`1.35` vom `14. August 1998`
LaTeX	`14th August 1998`
Pages	5 (max: 5)

Color Cluster Rotation

D. Paulus[1] *and L. Csink*[2] *and H. Niemann*[1]

[1]Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, D–91058 Erlangen, Germany
Tel.: +49–9131–85-7775, FAX: +49–9131–303811
`paulus@informatik.uni-erlangen.de`
`http://www5.informatik.uni-erlangen.de/Persons/pa`

[2]Kandó Polytechnic of Technology
Institute of Informatics
POB 112, H-1431 Budapest, Hungary
Tel.: +36–1–3684610, FAX: +36–1–3689632
`csink@novserv.obuda.kando.hu`
`http://www.kando.hu`

ABSTRACT

The distribution of color values in color images depends on the illumination which varies widely under real–world conditions. We present a new approach for color normalization which adjusts the statistical properties of the distribution to predefined values. We introduce two algorithms based on geometric manipulations of the color cluster. Our new color rotation algorithm is tested on some natural and synthetic images.

1. INTRODUCTION

The importance of color for computer vision is currently increasing [2, 4, 9, 10]. Although illumination of a scene may change, the human observer perceives the color of the objects in the scene almost independently from the illumination variations.

Many color spaces exist and are used in different applications. For computer vision, mostly RGB is used since it is directly technically available and most quality cameras supply RGB signals.

In this contribution we present new results of our approach to color normalization [7], whose results are similar to those of [9], but no neural algorithm is used and all computations are done in RGB rather than in some other color space.

In Sect. 2 we review color constancy algorithms. Such algorithms can play an important role for image retrival from image databases. Model based computer vision using color images also depends on standardized data. In Sect. 3 we describe cluster analysis which is applied in Sect. 4 for geometric transformations and in Sect. 5 for statistical analysis. In Sect. 6 we investigate the effect of our normalization algorithms on natural and synthetic images. First results of ongoing research on object localization using histogram backprojection [10] in combination with color normalization are presented as well.

2. COLOR NORMALIZATION ALGORITHMS

One of the most frequently cited papers in the area of color algorithms is [10]; in order to identify color objects in a scene color histograms are used. However, a disadvantage of this so called color indexing method is its sensitivity to illumination changes. The so called comprehensive color image normalization presented in [2] is shown to increase localization and object classification results in combination with color indexing. The idea of this iterative two–stage algorithm is to normalize each color pixel first, using the rgb color space in which the intensity is normalized; second, each color channel is normalized separately so that the sum of the color components is equal to one third of the number of pixels. These two steps are repeated, until no more changes occur. Since global intensity changes are eliminated from the normalized images, and since the number of color values may be reduced considerably, the images may look less natural to the observer, although the recognition rates using these images is higher.

Pomierski and Gross [9] propose to use an artificial neural network (ANN) to compute principal components of color clusters with a technique described in [6]. The color space used in this work is (RG, BY, WB) (red–green, blue–yellow, white–black) which is motivated by neuro–physiology. The color space transformations from a color vector $\boldsymbol{f}$ in RGB to a vector $\widetilde{\boldsymbol{f}} = \boldsymbol{A}\boldsymbol{f}$ in RG, BY, WB use

[1]This work was funded partially by the Deutsche Forschungsgemeinschaft (DFG) under grant number SFB 182. Only the author is responsible for the contents.

[2]The work of the second author was funded by the Hungarian State Eötvös Fellowship, grant no. 59.079/1996 and by DAAD, Germany, in 1998.

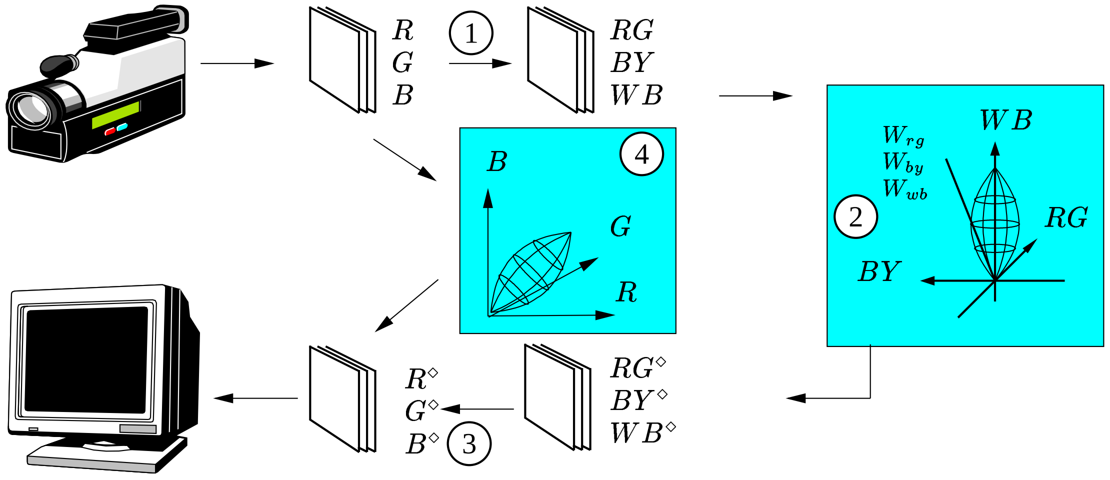

Fig. 1: Conversion of Pomierski (partially from [9])

the matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{pmatrix} 6.9012 & -13.9416 & 7.0404 \\ -12.4116 & .0048 & 12.4068 \\ 20.9968 & 21.1423 & 20.8609 \end{pmatrix} \qquad (1)$$

After finding the principal component of the color cluster in (RG, BY, WB), i.e. the direction of the eigenvector belonging to the greatest eigenvalue, the cluster is rotated so that this vector points to the WB direction of the $(RG, BY, WB)-$cube. The last step is a nonlinear stretching so that the cluster is distributed along this axis. As in the comprehensive color normalization, no reference image or calibration is required in order to transform an arbitrary image to normalized colors. In many cases, the normalized image looks “better” to the human observer.

One question now is whether we can yield similar results for computer vision, as [9] demonstrates for human vision, without an explicit transformation to another color space. This is depicted as ④ in Fig. 1.

3. COLOR CLUSTER ANALYSIS

Our new approach starts with color cluster analysis of a color image $[\boldsymbol{f}_{ij}]_{1\leq i\leq N, 1<\leq j<M}$ in the following steps which are common to the two algorithms described in Sect. 4 and Sect. 5 (we write $\boldsymbol{f}$ for pixels $\boldsymbol{f}_{ij}$ in the following):

1. Compute the cluster center of all pixels $\boldsymbol{f}$ by $\boldsymbol{m} = E[\boldsymbol{f}]$ which is the vector pointing to the center of gravity.
2. Let $\boldsymbol{C}$ be the (3×3)–matrix defined by

$$\boldsymbol{C} = E\left[(\boldsymbol{f} - \boldsymbol{m})(\boldsymbol{f} - \boldsymbol{m})^{\mathrm{T}}\right]$$

whose eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors are simply computed directly.

3. Denote the eigenvector belonging to the largest eigenvalue by $\boldsymbol{v} = (a, b, c)^{\mathrm{T}}$.

The vector denoting the principal orientation $\tilde{v}$ of the cluster in (RG, BY, WB) is computed in a similar way from $\widetilde{\boldsymbol{C}} = \boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^{\mathrm{T}}$.

Now two approaches have been tried, an idea which we called *color rotation in RGB* (Sect. 4) and the adaptation of the *whitening transform* (Sect. 5, [3]). Both are based on the same idea as [9], namely that in technical environments most of the objects are gray. The mean of the color vectors is expected on the gray–axis in the color system. If it is not there it will be rotated to this position.

4. COLOR ROTATION IN RGB–SPACE

From geometric considerations we proceed as follows in order to rotate the cluster to the main diagonal (Fig. 2):

4. Find the normal $\boldsymbol{n}'$ through the origin on the plane defined by the main diagonal in the RGB–cube and the principal component of the cluster: $\boldsymbol{n}' = (a, b, c)^{\mathrm{T}} \times \frac{1}{\sqrt{3}}(1, 1, 1)^{\mathrm{T}}$, where $\times$ denotes the vectorial product in $\mathbb{R}^3$. The rotation angle ϕ' is computed from the dot product of eigenvector and the diagonal: $\cos\phi' = (a, b, c)^{\mathrm{T}} \cdot \frac{1}{\sqrt{3}}(1, 1, 1)^{\mathrm{T}}$.
In order to rotate with ϕ, we use the Rodrigues formula [1, p. 150] for the rotation by an angle ϕ around an axis expressed as a vector $\boldsymbol{n}$: $\boldsymbol{R}_3(\phi, \boldsymbol{n}) = \boldsymbol{Id}_3 - \sin\phi\, \boldsymbol{U}(\boldsymbol{n}) + (1 - \cos\phi)\boldsymbol{U}^2(\boldsymbol{n})$ where $\boldsymbol{U}^2(\boldsymbol{n}) = \boldsymbol{n}\boldsymbol{n}^{\mathrm{T}} - \boldsymbol{Id}_3$ and $||\boldsymbol{n}|| = 1$.

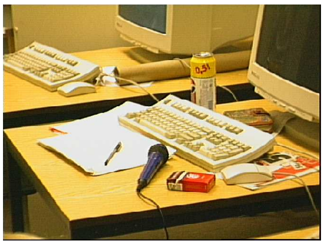

Fig. 3: Input image[3]

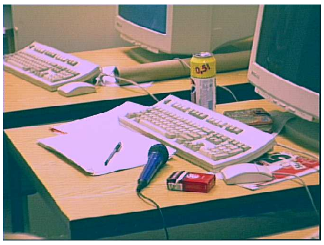

Fig. 4: Algorithm of Sect. 4

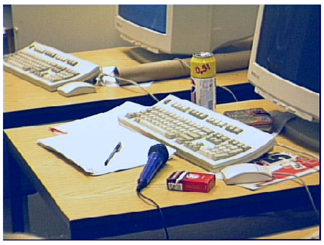

Fig. 5: Algorithm of Sect. 5

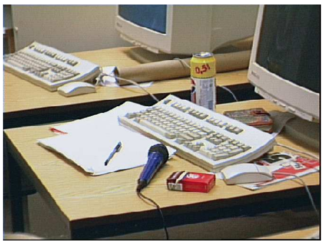

Fig. 6: Algorithm of Pomierski

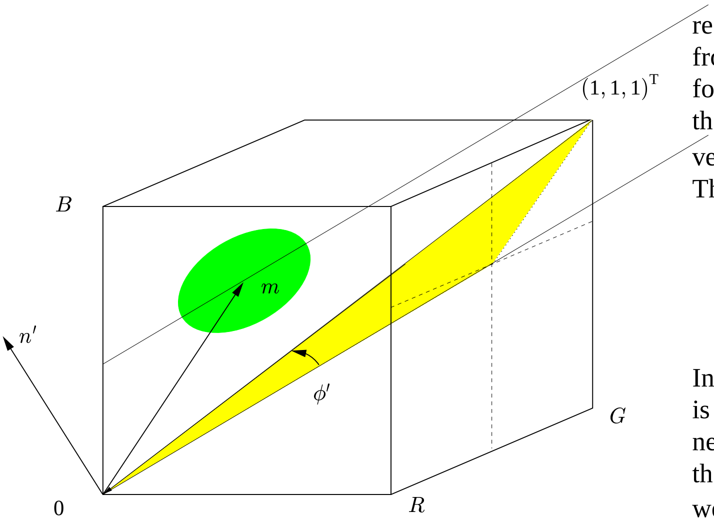

Fig. 2: Color rotation in RGB

The matrix $\boldsymbol{Id}_3$ is the identity matrix. The matrix $\boldsymbol{U}$ for an axis $\boldsymbol{n} = \left(n_x, n_y, n_z\right)^{\mathrm{T}}$ is in this case: $\boldsymbol{U}(\boldsymbol{n}') = \frac{\sqrt{3}}{3}\begin{pmatrix} 0 & b-a & c-a \\ a-b & 0 & c-b \\ a-c & b-c & 0 \end{pmatrix}$

5. Let
$$\boldsymbol{m}' = \frac{\|\boldsymbol{m}\|}{\cos\phi'}(1,1,1)^{\mathrm{T}} \quad .$$
Convert each pixel $\boldsymbol{f}$ to normalized $\boldsymbol{f}'$ by $\boldsymbol{f}' = \boldsymbol{U}(\boldsymbol{n}')(\boldsymbol{f} - \boldsymbol{m}) + \boldsymbol{m}'$.
6. Scaling by a variable factor is allowed (default is no scaling). The overflows above 255 and the underflows under 0 are clipped to 255 and 0, respectively.

The result is a color image which has a normalized color distribution; the mean of the color vectors is on the main diagonal of the RGB–cube; the first principal component of the cluster is on the same diagonal.

As the conversion matrix (1) is not orthogonal, the results of this rotation are not the same, as those resulting from the algorithm in [9]. We can, however, apply the same formulas to create a rotation matrix $\tilde{\boldsymbol{R}}$ which rotates $\tilde{\boldsymbol{v}}$ to the WB–axis in (RG, BY, WB). The normalized color vectors $\boldsymbol{f}'$ in RGB are then computed by $\boldsymbol{f}' = \boldsymbol{A}^{-1}\tilde{\boldsymbol{R}}\tilde{\boldsymbol{f}}'$. The normalization can thus be done entirely in RGB by

$$\boldsymbol{f}' = \boldsymbol{A}^{-1}\tilde{\boldsymbol{R}}\boldsymbol{A}\,(\boldsymbol{f} - \boldsymbol{m}) + \boldsymbol{A}^{-1}\tilde{\boldsymbol{R}}\boldsymbol{A}\boldsymbol{m} \quad . \qquad (2)$$

5. WHITENING TRANSFORM

In Fukunaga [3] the *whitening transform* is introduced, which is an orthonormal transform mapping the principal components of a cluster into the (orthogonal) eigenvectors, and at the same time a scaling is done with $\frac{1}{\sqrt{\lambda_i}}$. In this section we examine whether the above transform can be used for image normalization and we compare the results with those of section 4. We first perform steps 1–3 as described in Sect. 3 and then proceed as follows:

4. Compute the eigenvector matrix $\boldsymbol{V}$ of $\boldsymbol{C}$, and denote
$$\boldsymbol{\Lambda} = \begin{pmatrix} \frac{255}{\sqrt{\lambda_1}} & 0 & 0 \\ 0 & \frac{255}{\sqrt{\lambda_1}} & 0 \\ 0 & 0 & \frac{255}{\sqrt{\lambda_1}} \end{pmatrix}$$
where λ_1 is the greatest eigenvalue of $\boldsymbol{C}$. We note that 255 appears in the dominator instead of 1, since 255 is the scale in which R, G and B may vary. We also note that here we modified the original transform not wanting to scale each principal component with the corresponding fraction involving its eigenvalue, as this would change the shape of the cluster more than it is desirable.
5. For each pixel $\boldsymbol{f}$, let us form $\boldsymbol{f}' = \boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{m}) \quad .$
6. Rotate the cluster along the R axis by 45 degrees in the positive direction, and then rotate the image along the B axis with 45 degrees again and shift the image along the main axis of the RGB-cube by $(128, 128, 128)^{\mathrm{T}}$. After clipping the values by 255 we get the result.

[3] Color images are included on the CD–ROM

The result again is a color image which has a normalized color distribution; the mean of the color vectors is on the main diagonal of the RGB–cube; the first principal component of the cluster is on the same diagonal. In addition, the second axis of the cluster is rotated to the diagonal $(0, 1, 1)^{\mathrm{T}}$ in the RGB–cube.

6. EXPERIMENTS

We integrated both algorithms in our image analysis system [8] and made experiments with both synthetic and real images. Fig. 3 shows one scene as captured from the camera. The results of a conversion with our first algorithm (Sect. 4) is shown in Fig. 4.[4] Fig. 5 illustrates the results of the modified whitening transform (Sect. 5). The result of the algorithm in [9] is shown in Fig. 6.[5]

Experiments on synthetic images show that the algorithms work even for unusual directions of the principal components.

A red object (Fig. 7 (left)) and a blue object (Fig. 7 (right)) are both captured with a high focal length setting for a zoom camera. These objects are present in two scenes (Fig. 8 (top) and Fig. 8 (bottom)) captured with different settings of the zoom lens and different lighting conditions.

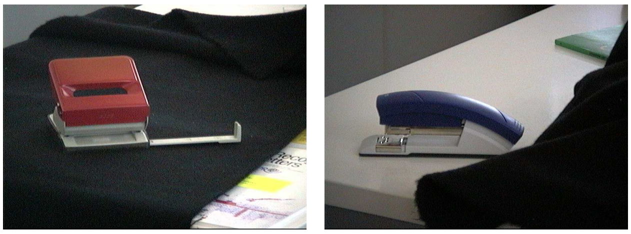

Fig. 7: Two objects captured from the camera with high focal length.

The effects of color normalization on object localization based on color backprojection [10] are presented in Fig. 9 and Fig. 10; the images show backprojected values, filtered by a median filter of the approximate size of the object. No advantage of color normalization can be seen for the first object Fig. 7 (left), since the general shift of colors to red increases the number of red pixels and thus deteriorates the backprojection of a red object. The results for the second object (Fig. 7 (right)) are shown in Fig. 10. Here, the modified whitening transform improves the results of backprojection, since the rotation of the second principal component helps discriminating blue color from the others. The results of the comprehensive color normalization look better than those of the color rotation, although the red object in Fig. 8 (bottom) is not found.

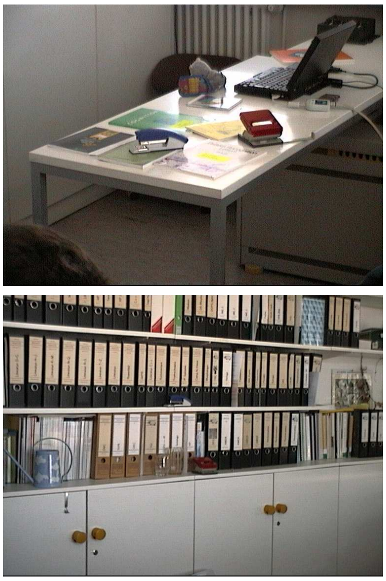

Fig. 8: Two scenes containing the objects shown in Fig. 7

7. CONCLUSION

We presented two established algorithms and two new approaches to color normalization: one was inspired by [9]; the other is based on an extension of the whitening transform [3]. We claim that color normalization can facilitate more reliable object localization under changing lighting conditions. The best choice for the proposed normalization algorithms, however, depends on the object to be localized, but not on the scene. Further investigations will be done in order to optimize object localization using backprojection as in [10] in combination with our algorithms for color normalization and with other correction algorithms and strategies which compensate for color changes, such as [4, 2].

8. REFERENCES

[1] O. Faugeras. *Three–Dimensional Computer Vision – A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[2] G.D. Finlayson, B. Schiele, and J.L. Crowley. Comprehensive colour image normalization. In H. Burkhard and B. Neumann, editors, *Computer Vision — ECCV ’98*, pages I/475–490, Heidelberg, 1998. Springer.

[3] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[4] For the printout, all color images are vector quantized to 32 colors using the median cut algorithm [5].

[5] Thanks to T. Pomierski for providing this image.

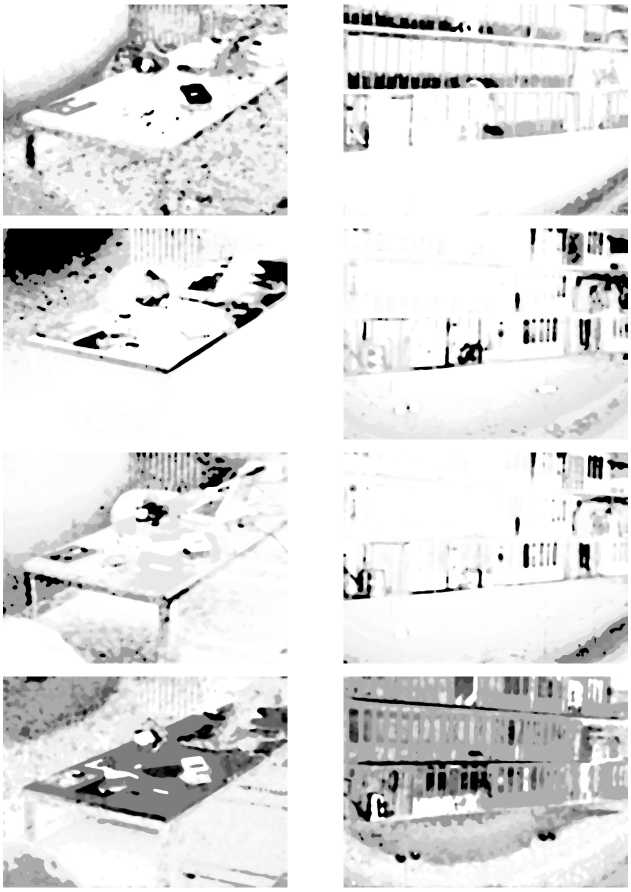

Fig. 9: Backprojection of the object in Fig. 7 (left) to the scenes in Fig. 8. Line 1: without normalization; line 2: using the comprehensive color normalization [2] line 3: with color rotatation (Sect. 4); line 4: using the modified whitening transform (Sect. 5)

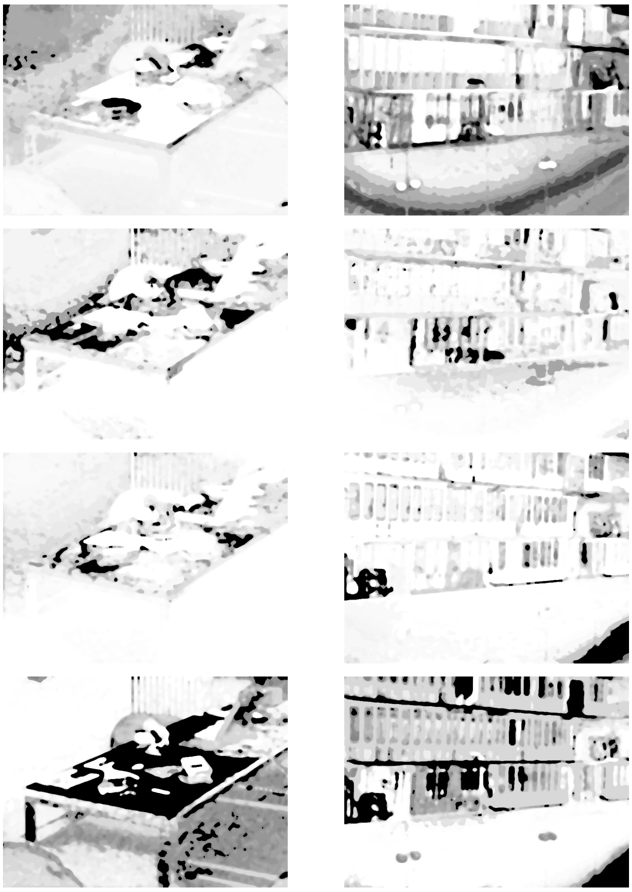

Fig. 10: Backprojection of the object in Fig. 7 (right) to the scenes in Fig. 8. Line 1: without normalization; line 2: using the comprehensive color normalization [2] line 3: with color rotatation (Sect. 4); line 4: using the modified whitening transform (Sect. 5)

[4] B. V. Funt and G. D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(5):522–529, 1995.

[5] P. Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 16(3):297–307, July 1982.

[6] E. Oja and J. Parkkinen. On Subspace Clustering. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 692–695. San Diego, 1984.

[7] D. Paulus and L. Csink. On color normalization. In S. Tanás and B. Jósef, editors, *Magyar Képfeldolgozók és Alakfelismerõk Orszßgos Konferencißja, Konferenciakiadvßny*, pages 222–229, Budapest, 1997.

[8] D. Paulus and J. Hornegger. *Pattern Recognition of Images and Speech in C++*. Vieweg, Braunschweig, 1997.

[9] T. Pomierski and H.M. Groß. Verfahren zur empfindungsgemäßen Farbumstimmung. In G. Sagerer, S. Posch, and F. Kummert, editors, *Mustererkennung 1995*, pages 473–480, Berlin, September 1995. Springer.

[10] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.