On color normalization

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

Kandó Polytechnic of Technology
Institute of Informatics
POB 112, H-1431 Budapest, Hungary
Tel.: +36–1–3684610, FAX: +36–1–1689632
csink@novserv.obuda.kando.hu
http://www.kando.hu

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

D. Paulus, L. Csink: *On Color Normalization*, in S. Tanás, B. Jósef (Hrsg.): *Magyar Képfeldolgozók és Alakfelismerõk Orszßgos Konferencißja, Konferenciakiadvßny*, Budapest, 1997, S. 222–229.

On color normalization

A színnormalizációról

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

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

Kandó Polytechnic of Technology
Institute of Informatics
POB 112, H-1431 Budapest, Hungary
Tel.: +36–1–3684610, FAX: +36–1–1689632
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 or color constancy which adjusts the statistical properties of the distribution to predefined values. Such algorithms play an important role for image retrival from image databases. Model based computer vision using color images also depends on standardized data.

Our new method differs from existing neural-based approaches used for color constancy and also from the whitening transform which is introduced to normalize distributions for numerical classification. The new color rotation algorithm is tested on some natural and synthetic images.

Kivonat

Színes képeken a színértékek eloszlása a megvilágítástól függ, amely valós körülmények között erősen ingadozhat. A színnormalizációra vagy színállandóságra egy új megközelítést mutatunk be, amely az eloszlás statisztikai tulajdonságait előre meghatározott értékekre állítja be. Az ilyen algoritmusoknak fontos szerepe van akkor, amikor képi adatbázisokban folytatunk kereséseket. A modell alapú színes számítógépes látás szintén standardizált adatoktól függ. Az új módszerünk különbözik a színállandósággal foglalkozó neurális elvű megközelítésektől, valamint az ún. whitening transzformációtól is, melyet numerikus klasszifikációra szolgáló eloszlások normalizációjára vezettek be. Az új színforgatási algoritmust egyaránt teszteltük színes és szintetikus képeken.

1 Motivation

The importance of color for computer vision is currently increasing, as can be seen from the contributions in [8] or from [4]. 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. The study of such kind of adaptation is an important topic of color machine vision [9].

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 cameras supply RGB signals.

In this contribution we present a new approach, 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. 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 [11] in combination with color normalization are presented as well.

[1]This work of the first author was funded partially by the German Research Foundation (DFG) under grant number SFB 182.

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

2 Color Normalization Algorithms

One of the most frequently cited papers in the area of color normalization algorithms is [11] *to develop visual skills for robots that allow them to interact with a dynamic, realistic environment.* In order to identify color objects in a scene color histograms are used. However, a disadvantage of their color indexing method is the sensitivity to illumination changes. This can be helped by preprocessing with a color constancy algorithm (e.g. [5]).

Pomierski and Gross [9] propose to use an artificial neural network (ANN) to compute principal components of color cluster 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.. After finding the principal component, i.e. the direction of the eigenvector belonging to the greatest eigenvalue, the cluster is rotated such that this vector points to the WB direction of the $(RG, BY, WB)-$cube. The last step is a nonlinear streching so that the cluster is distributed along this axis. The major advantage of this idea is that no reference image or calibration is required in order to transform an arbitrary image to normalized colors.

The color space transformations from a color vector $\boldsymbol{f}$ in RGB to a vector $\widetilde{\boldsymbol{f}}$ in RG, BY, WB are:

$$\widetilde{\boldsymbol{f}} = \begin{pmatrix} RG \\ BY \\ WB \end{pmatrix} = \boldsymbol{T} \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \boldsymbol{T}\boldsymbol{f}$$

where

$$\boldsymbol{T} = \begin{pmatrix} 6.9012 & -13.9416 & 7.0404 \\ -12.4116 & .0048 & 12.4068 \\ 20.9968 & 21.1423 & 20.8609 \end{pmatrix}$$

The steps marked with 1 and 3 in Fig. 1 are the color space transformations by $\boldsymbol{T}$ and $\boldsymbol{T}^{-1}$, respectively. 2 stands for the search of the principal component and the neural-based rotation to the WB–axis of the (RG, BY, WB)–space. The basic question now is whether we can yield similar results and effects for computer vision, as [9] demonstrates for human vision, without an explicit transformation to (RG, BY, WB). This is depicted as 4 in Fig. 1.

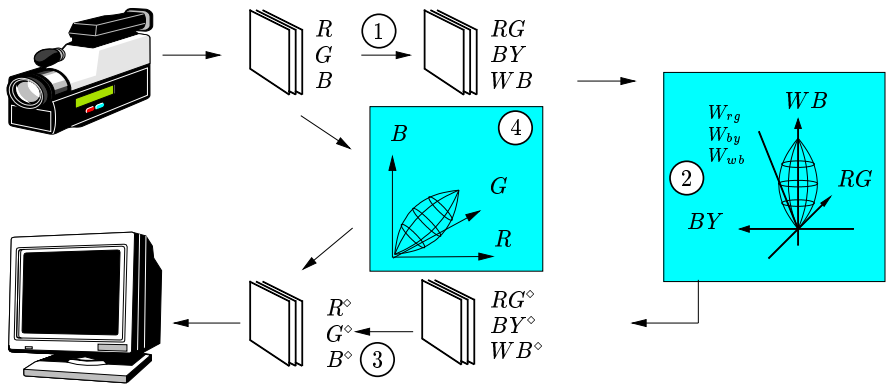

Fig. 1: Conversion of Pomierski (partially from [9])

3 Color Cluster Analysis

Our approach starts with color cluster analysis of a color image $[\boldsymbol{f}_{ij}]$ in the following steps which are common to the two algorithms described in Sect. 4 and Sect. 5:

1. Assume that $\boldsymbol{m} = E\{\boldsymbol{f}_{ij}\}$ is the vector pointing to the centre of gravity. Translate each color vector $\boldsymbol{f}_{ij} \leftarrow \boldsymbol{f}_{ij} - \boldsymbol{m}$.
2. Let $\boldsymbol{C}$ be the (3×3)–matrix defined by

$$\boldsymbol{C} = E\{\boldsymbol{f}_{ij}\boldsymbol{f}_{ij}{}^{\mathrm{T}}\}$$

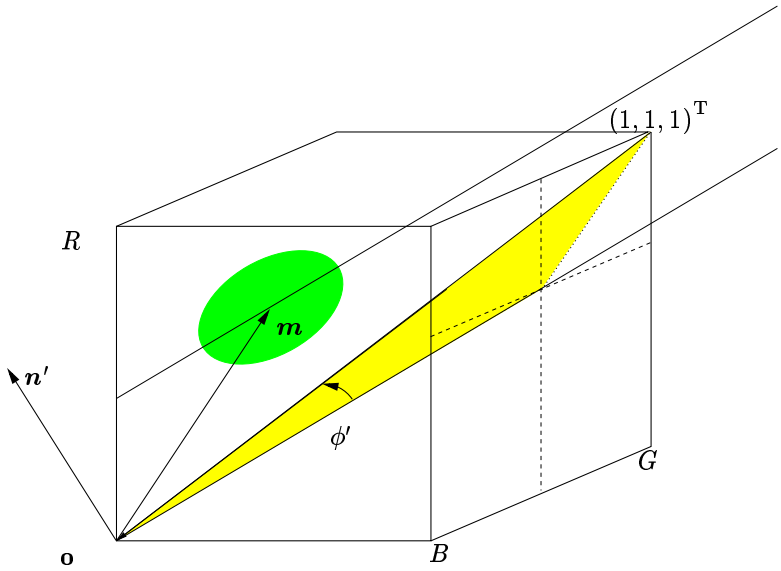

Fig. 2: Color rotation in RGB

whose eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and eigenvectors are simply computed directly (e.g. by the Jacobi method).

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

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, [1]).

4 Color Rotation in RGB

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: $\boldsymbol{n}' = (a, b, c)^{\mathrm{T}} \times \frac{1}{\sqrt{3}}(1, 1, 1)^{\mathrm{T}}$, i.e., $\boldsymbol{n}' = \frac{1}{\sqrt{3}}(b - c, c - a, a - b)^{\mathrm{T}}$, where $\times$ denotes the vectorial product in $\mathbb{R}^3$. The rotation angle $\cos\phi'$ 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 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 $\|\boldsymbol{n}\| = 1$.

   The matrix $\boldsymbol{Id}_3$ is the identity matrix. The matrix $\boldsymbol{U}$ for an axis $\boldsymbol{n} = (n_x, n_y, n_z)^{\mathrm{T}}$ is

$$\boldsymbol{U}(\boldsymbol{n}) = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix} .$$

   Here:
$$\boldsymbol{U}(\boldsymbol{n}') = \frac{\sqrt{3}}{3} \begin{pmatrix} 0 & b-a & c-a \\ a-b & 0 & c-b \\ a-c & b-c & 0 \end{pmatrix}$$
   The rotation matrix $\boldsymbol{R}_3(\phi', \boldsymbol{n}')$ resulting from this formula for this particular case is given in Appendix A.

5. Translate each pixel in the rotated image with a parameter along the axis $(1,1,1)^{\mathrm{T}}$

$$\boldsymbol{m}' = \frac{\|\boldsymbol{m}\|}{\cos\phi'}(1,1,1)^{\mathrm{T}}$$

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.

5 Whitening transform

In Fukunaga [1] the *whitening transform* is introduced, which is an orthonormal transform mapping the principal components of the color 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 nominator of the above fractions 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. Let us form

$$\boldsymbol{f}'_{ij} = \boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{f}_{ij}$$

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 (so that they should not point outside the RGB-cube) we get the result.

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 [7] and made experiments with both synthetic and real images. Fig. 3 shows one scene as captured from the camera.[1]

The results of a conversion with our first algorithm (Sect. 4) is shown in Fig. 4.[2] Fig. 5 illustrates the results of the modified whitening transform (Sect. 5). The corresponding color clusters are visualized in Fig. 6, 7, and 8.

Two experiments on synthetic images are shown in Fig. 9, proving that the algorithms work on principal components which are furthest away from the main diagonal in the RGB cube, and for eigenvalues which are zero.

A red object (Fig. 10 (left)) and a blue object (Fig. 10 (right)) are both captured with a high focal length setting for a zoom camera. These objects are present in two scences (Fig. 11 (left) and Fig. 11 (right)) captured with different settings of the zoom lens and different lighting conditions.

Fig. 3: Input image

Fig. 4: Algorithm of Sect. 4

Fig. 5: Algorithm of Sect. 5

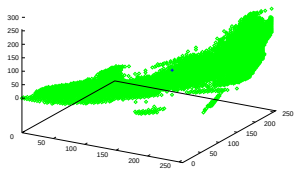

Fig. 6: Cluster of Fig. 3

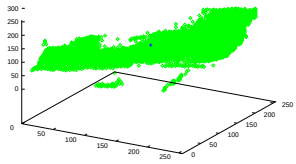

Fig. 7: Cluster of Fig. 4

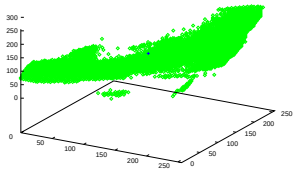

Fig. 8: Cluster of Fig. 5

The results of the two proposed color normalization algorithms on the scenes in Fig. 11 are shown in Fig. 12. Even in the gray–level print, the changes are visible in comparison to the original images. The mean intensity of the images is higher than the original. For the whitening transformed images, the *white* was shifted in the *red* direction, such that the table in the center of the image now is light pink. This effect is due to the normalization of the second principal component and could be observed in most of our experiments.

The effects of color normalization on object localization based on color backprojection [11] are presented in Fig. 13 and Fig. 14. The results are rather disappointing for the first object (Fig. 10 (left)). No advantage of color normalization can be seen for the backprojection algorithm, 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. 10 (right) are shown in Fig. 14. 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.

7 Conclusion

We presented two new approaches to color normalization. One is based on an extensions of the whitening transform [1]. The other was inspired by [9]. Normalization does not make images look better. 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. Further investigations will be done here in order to optimize object localization using backprojection as in [11, 10] in combination with our algorithms for color normalization and with other correction algorithms and strategies which compensate for color changes, such as [2].

References

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

[2] B. V. Funt, G. D. Finlayson: *Color Constant Color Indexing*, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Bd. 17, Nr. 5, 1995, S. 522–529.

[1] The color images are available in PostScript version of the paper which can be found in the world wide web in the section *Publications* in `http://www5.informatik.uni-erlangen.de`.

[2] All color images are vector quantized to 32 colors using the median cut algorithm [3].

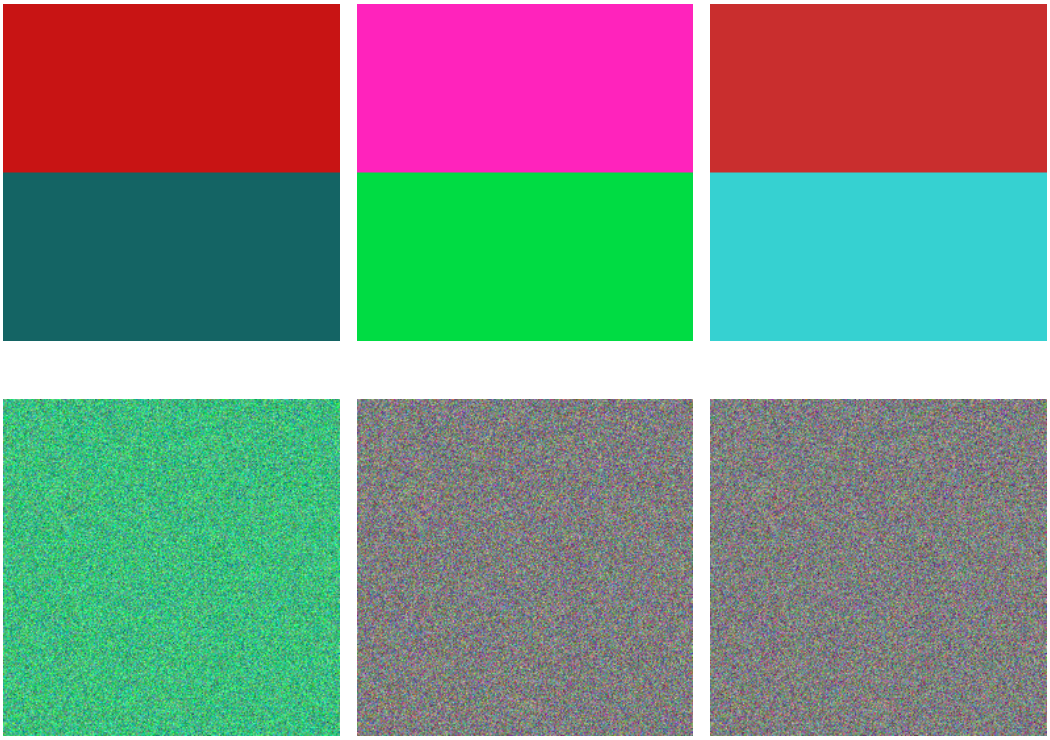

Fig. 9: Color normalization on synthetic images: Input image (left), color rotation (middle), whitening (right). First row: Two constant color values. Second row: Gaussian color noise

Fig. 10: Two objects captured from the camera with high focal length.

Fig. 11: Two scenes containing the objects shown in Fig. 10

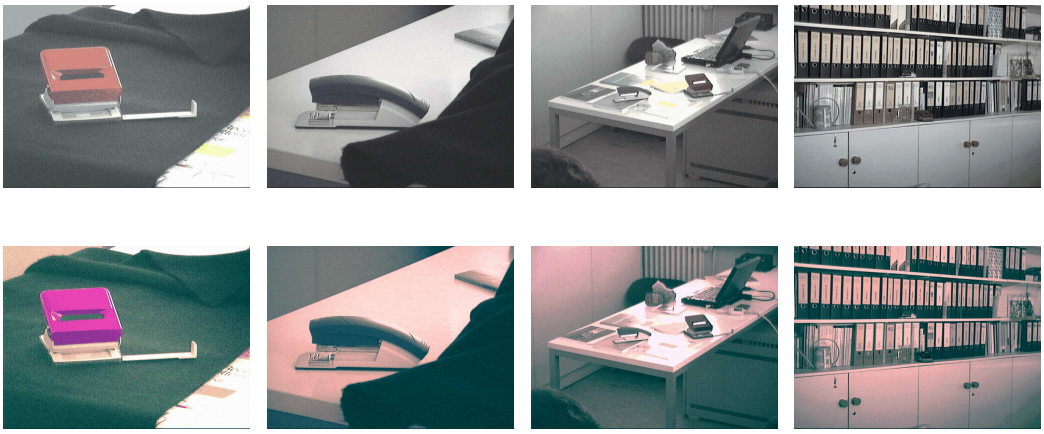

Fig. 12: Normalization of the images shown in Fig. 10 and Fig. 11 with color rotatation (Sect. 4) (top), and with the modified whitening transform (Sect. 5) (bottom)

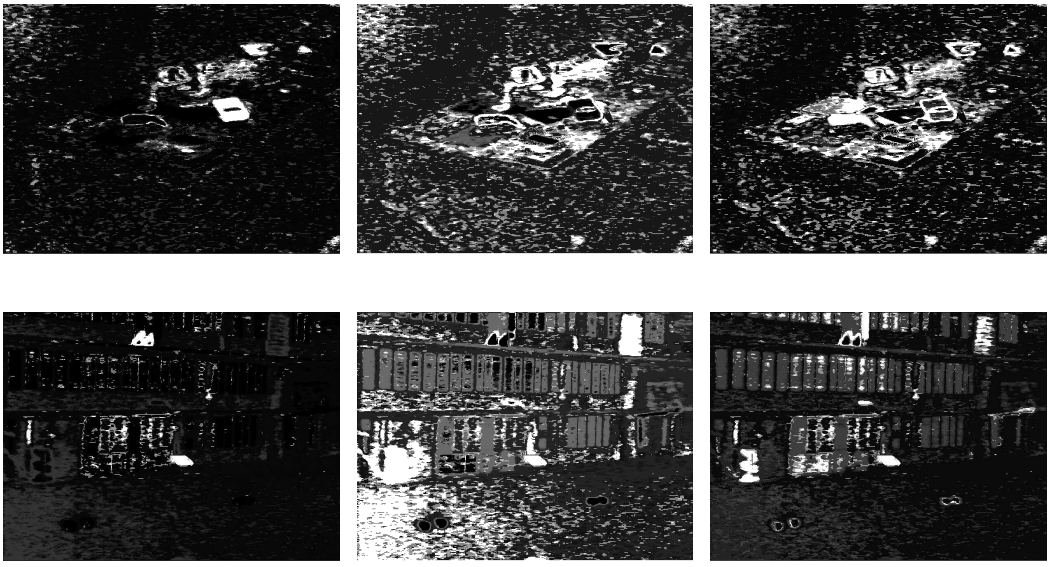

Fig. 13: Backprojection of the object in Fig. 10 (left) to the scenes in Fig. 11. Left: without normalization; middle: with color rotatation (Sect. 4); right: using the modified whitening transform (Sect. 5) (bottom)

[3] P. Heckbert: *Color Image Quantization for Frame Buffer Display*, *Computer Graphics*, Bd. 16, Nr. 3, July 1982, S. 297–307.

[4] Q.-T. Luong: *Color in Computer Vision*, in C. H. Chen, L. F. Pau, P. S. P. Wang (Hrsg.): *Handbook of Pattern Recognition & Computer Vision*, World Scientific Publishing, Singapore, 1993, S. 311–369.

[5] U. Mahlmeister, H. Pahl, G. Sommer: *Color–Orientation Indexing*, in B. Jähne, P. Geißler, H. Haußecker, F. Hering (Hrsg.): *Mustererkennung 1996*, Springer, Heidelberg, September 1996, S. 3–10.

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

[7] D. Paulus, J. Hornegger: *Pattern Recognition of Images and Speech in C++*, Advanced Studies in Computer Science, Vieweg, Braunschweig, 1997.

[8] D. Paulus, T. Wagner (Hrsg.): *Dritter Workshop Farbbildverarbeitung*, IRB-Verlag,

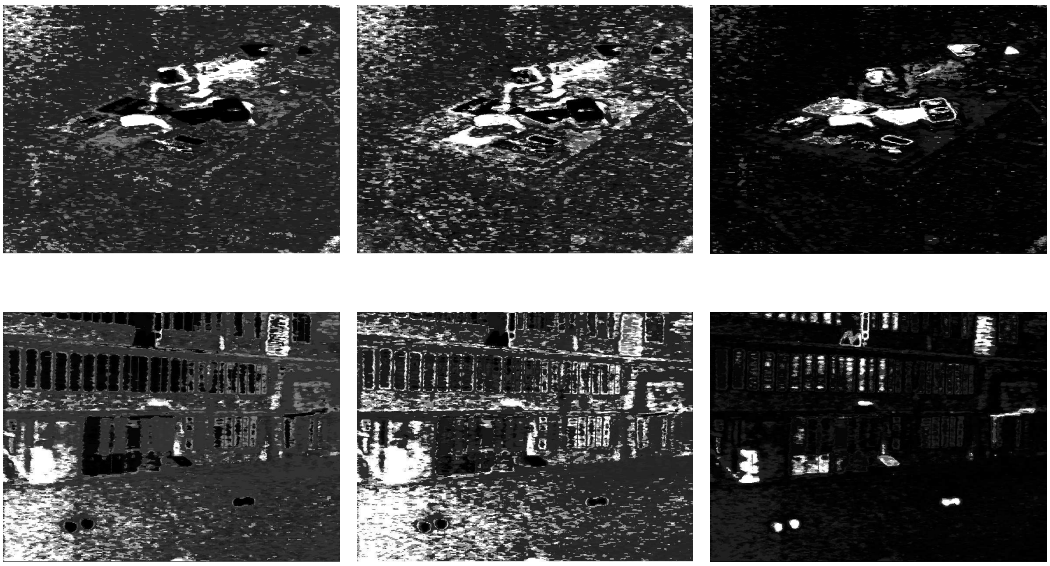

Fig. 14: Backprojection of the object in Fig. 10 (right) to the scenes in Fig. 11. Left: without normalization; middle: with color rotatation (Sect. 4); right: using the modified whitening transform (Sect. 5)

Stuttgart, 1997.

[9] T. Pomierski, H. Groß: *Verfahren zur empfindungsgemäßen Farbumstimmung*, in G. Sagerer, S. Posch, F. Kummert (Hrsg.): *Mustererkennung 1995*, Springer, Berlin, September 1995, S. 473–480.

[10] H. S. Sawhney, J. L. Hafner: 1995.

[11] M. J. Swain, D. H. Ballard: *Color Indexing*, *International Journal of Computer Vision*, Bd. 7, Nr. 1, November 1991, S. 11–32.

A Rotation Matrix

$$
\begin{aligned}
\boldsymbol{R}_3(\phi', \boldsymbol{n}') &= [\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{m,n} \; (m, n \in \{1, 2, 3\}) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{1,1} &= 1 - \mathrm{H}(-2a^2 + 2ab - b^2 - c^2 + 2ca) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{1,2} &= -\sqrt{\mathrm{G}}(b - a) + \mathrm{H}(a - c)(b - c) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{1,3} &= -\sqrt{\mathrm{G}}(c - a) - \mathrm{H}(a - b)(b - c) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{2,1} &= -\sqrt{\mathrm{G}}(a - b) + \mathrm{H}(a - c)(b - c) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{2,2} &= 1 - \mathrm{H}(-a^2 + 2ab - 2b^2 + 2bc - c^2) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{2,3} &= -\sqrt{\mathrm{G}}(c - b) + \mathrm{H}(a - b)(a - c) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{3,1} &= -\sqrt{\mathrm{G}}(a - c) - \mathrm{H}(a - b)(b - c), \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{3,2} &= -\sqrt{\mathrm{G}}(b - c) + \mathrm{H}(a - b)(a - c) \\
[\boldsymbol{R}_3(\phi', \boldsymbol{n}')]_{3,3} &= 1 - \mathrm{H}(-2c^2 + 2ca - a^2 - b^2 + 2bc) \\
\mathrm{H} &:= 1 - \frac{\sqrt{3}}{3}(a + b + c) \\
\mathrm{G} &:= \frac{1}{3}\left((b - c)^2 + (c - a)^2 + (a - b)^2\right)
\end{aligned}
$$