

Learned Color Constancy from Local Invariant Regions

Tijmen Moerland, *unknown*, Frédéric Jurie, *unknown*

F. Jurie is with INRIA Rhône-Alpes.

Abstract

Human color constancy, the ability to correct for color deviation caused by illumination changes, is still unmatched by computer vision performances. The model used in this paper learns how illumination changes affect reference objects. It can then explain complex non-linear color transformations with only a few parameters. Global color changes can consequently be estimated from several observations of small reference regions.

Index Terms

Machine learning, color, photometry, object recognition, invariants, image models.

I. INTRODUCTION

Color is not an intrinsic property of objects. Illumination change can severely modify colors acquired by a camera and can make the use of color information for subsequent tasks difficult. The active work done by our brain to assign color to an observe scene is difficult to reproduce, even with complex computer vision techniques.

The color constancy problem can be expressed as that of deriving an image of the scene as it would appear under a canonical illuminant, given the image of this scene under an unknown illuminant. The mapping has to account for the change in relative spectral power distribution between the unknown and canonical illuminant.

Usually, two different ways to investigate color constancy are distinguished:

- the first approach is based on psychophysical investigations [1]. The problem with this kind of approach is that conclusions are hard to draw due to the complex experimental environment necessary to examine color constancy.
- the alternative approach tries to understand the principles involved with the spectral image formation [2], [3]. The knowledge of physical laws allows to build - for example - illumination independent color ratios [4], [5]. These color invariants unfortunately have several drawbacks, since these transformations can be singular for some sensor values or unstable at others [3].

A third - and new - way to address this problem is as a machine learning problem. By observing how colors are mapped between two images of the same scene under various "real world" lighting changes, it is possible to automatically build a model of possible color changes. First investigations in this area were proposed by Miller and Tieu [6]. They proposed the *color eigenflows model* that statistically learns joint color changes. However,

this promising approach suffers from a major drawback: color constancy is possible only if the image of the scene is aligned to the same scene viewed under the canonical illumination.

In this paper we try to go beyond the limitations of the Miller and Tieu's approach. The key idea is to evaluate color changes within a few reference regions of the scene (e.g. known objects) to compute the global color constancy. The presence of known objects in the image is clearly a restrictive assumption, which is however acceptable in numerous applications, including object recognition. We show that under this assumption much better results can be obtained.

The use of known objects through a matching process is not new in the color constancy area. Obdrzalek *et al.* [7] propose a similar approach, however they use a simple affine model for color change. Our approach is more powerful as our model is able to deal with complex non-linear transformations.

As it is often the case with many color constancy algorithms, our approach will only estimate the chromaticity of the color and will ignore the intensity component. There are many ways of normalizing RGB colors to eliminate the effect of the intensity. For this purpose we will use the standard HSV representation for colors and ignore the V component.

The rest of the paper is organized as follows. In the next section we will introduce the color flow model, as proposed by Miller and Tieu [6]. We will then proceed in section III by detailing improvements we made, in particular regarding photometric invariance and the choice of number of parameters. In section IV we detail the framework of reference regions and discuss how color flow parameters can be estimated from these regions. Experimental results are presented in section V and we conclude in section VI.

II. COLOR FLOW MODEL

In order to model changes in color, we adopt the *color flow* model [6], [8]. It is used to model *joint color changes* between images depicting the same scene. While the model itself is linear, the basis vectors used to describe the joint color changes are completely arbitrary and are not at all restricted to represent linear color changes. This provides great modeling power while keeping parameter estimation simple.

A color flow defines how colors jointly change between two aligned images. For *starting image* I_1 and *target image* I_2 , both containing N pixels, we define p_1^k and p_2^k to be corresponding pixels, $1 \leq k \leq N$, and their colors are given by the 3-vectors $c(p_1^k)$ and $c(p_2^k)$. As a color space C the RGB-cube is used where *rgb* components are real values within range

$0 \leq r, g, b \leq 255$. The color change $\mathbf{d}(p_1^k, p_2^k)$ from pixel p_1^k to p_2^k is then defined to be the difference vector:

$$\mathbf{d}(p_1^k, p_2^k) = \mathbf{c}(p_2^k) - \mathbf{c}(p_1^k). \quad (1)$$

A vector field $\phi' : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ can now be defined over C as follows:

$$\phi'(\mathbf{c}(p_1^k)) = \mathbf{d}(p_1^k, p_2^k), \quad 1 \leq k \leq N \quad (2)$$

which can be interpreted as a group of vectors defined for each pixel in the starting image, indicating its color change. The endpoint of the vector corresponds to the pixel's color in the target image. Since the vector field might have multiple definitions at the same point and none at other points, we refer to ϕ' as the *partially observed color flow*. Interpolation is performed by a 3-D Gaussian kernel to obtain the *full color flow* $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, defining a color change for every color in the color space C :

$$\phi(r, g, b) = \frac{\sum_{k=1}^N e^{-\|\mathbf{c}(p_1^k) - (r, g, b)^T\|^2 / 2\sigma^2} \phi'(\mathbf{c}(p_1^k))}{\sum_{k=1}^N e^{-\|\mathbf{c}(p_1^k) - (r, g, b)^T\|^2 / 2\sigma^2}} \quad (3)$$

where σ defines the size of the kernel and thus the amount of 'smoothing' on ϕ' . The partially observed color flow should be locally smooth for this interpolation to work. The full color flow is quantized by determining its value at specific points in the color space C . In particular, C is divided into 16 bins in each dimension to obtain a total of $16^3 = 4096$ bins, and the full color flow is calculated at the bin centers. The thus obtained 3-vectors are stacked in one column vector of $D_c = 4096 \times 3 = 12288$ rows which can be used to represent the full color flow.

A. Learning

Although it is theoretically possible for color changes to be independent of each other, in practice similar colors exhibit similar color changes, ensuring the smoothness of ϕ' required for (3). Furthermore, a small set of particular color flows can be used to model color changes under wide illumination variety. These color flows have been learned from a large set of training images in the way described below.

A total of 1646 images were taken from a poster displaying a wide range of colors. The poster was located in an office and illumination was changed by controlling blinds, turning fluorescent light on and off, closing and opening the office door and people walking in and

out. From this set, 800 pairs were randomly selected and the quantized full color flow was calculated and stored as a sample in a column of data matrix \mathbf{X} . Now each column represents a point in the \mathbb{R}^{D_c} space and we can model the distribution of the samples by performing a Principal Component Analysis. PCA finds the directions ('principle components') in order of decreasing variance. It can be performed by determining the row-mean vector \mathbf{m} of \mathbf{X} and applying (economy) Singular Value Decomposition on the matrix $\bar{\mathbf{X}} = \mathbf{X} - \mathbf{m}$. In the context of color flows, the principal components are dubbed *eigenflows*. A subset of the eigenflows is used as a model for color changes. A suitable size for the subset can be determined by considering the eigenvalues associated with the eigenflows, as these are a measure of the variance in the original data matrix in the direction of the eigenflow. We will come back to this issue in section III-B, after modifications to the method have been proposed.

B. Estimation

It is now possible to estimate a color flow observed between two test images in terms of a number of eigenflows. We write for the i th eigenflow $\psi_i(\mathbf{c}(p))$ and define a $D_c \times P$ model matrix \mathbf{M} to have the first P eigenflows in its columns. The issue here is that we have a partially observed flow from the test images, but a set of full color flows that are the eigenflows and it is necessary to transform one into the other. Straightforward calculation of the quantized full color flow for the test images by application of (3) will in general not be possible due to image size, as calculation for one 320×240 training image already takes minutes. In their article, Miller and Tieu take the reverse approach by converting the eigenflows into *flow images*. Each flow image F_i represents the effect of the i th eigenflow on the starting image of the test pair. They are obtained by placing at each location (x, y) in F_i the value of $\psi_i(\hat{\mathbf{c}}(p_i^k))$, where $\hat{\mathbf{c}}(p_i^k)$ is the color closest to $\mathbf{c}(p_i^k)$ for which ψ_i has been calculated. Defining the matrix \mathbf{F} to have F_i as its i th column, a least squares estimation can then be performed to compute *flow parameters* γ in the linear system

$$\mathbf{F}\gamma = I_2 - I_1 \quad (4)$$

Estimation can be performed by obtaining the pseudo inverse of \mathbf{F} . The number of equations in this system is equal to the number of pixel components $3N$ in the starting image. For images larger than about 800×600 the estimation becomes infeasible.

We have developed a mathematically equivalent procedure that transforms the partially observed color flow between the test images to a full color flow by taking averages per bin.



(a) RGB

(b) Hue, saturation

Fig. 1. Effects of shadow

The number of equations is then independent of image size and has an upper limit equal to the dimension of (full) color flows, $D_c = 12288$. Although this method is useful for processing large aligned images, we will be needing a more robust form of estimation than least squares and therefore skip detailed discussion.

III. PHOTOMETRIC INVARIANCE

Local effects such as shading, shadows and highlights cannot be explained in terms of a global color flow model. However they also disturb parameter estimation. Since shadows and highlights can very well move from one image to the next due to changes in illumination position, they will cause high magnitudes in the color difference between two images. The estimation of the color flow can thus be severely affected, resulting in a non-coherent set of flow parameters. Tieu and Miller [8] suggest to solve for this by allowing the flow parameters to vary linearly or quadratically over the image. Parameters are calculated individually and independently for small image patches of fixed size, after which the optimal linear or quadratic fit is found.

In figure 1(a) parameters have been calculated independently on 128×128 image patches, without applying the linear or quadratic constraint. We see that even without the constraint estimation is severely affected. The effect is also present when estimating on smaller image patches. The sensitivity to shadows and highlights is particularly worrisome for our estimation

from local reference regions as detected points are likely to lie on and around objects, precisely the location where those effects can be found. Note that a global robust estimation is not feasible for larger images, since individual outlying pixels have to be detected and the efficient estimation described at the end of II-B cannot be used.

In the proposed approach a way to describe the color of pixels invariant of shading, shadows and highlights is used to make the estimation robust to those effects. *Photometric invariant* color spaces like normalized *rgb*, hue and saturation, $l_1l_2l_3$ and $c_1c_2c_3$ [5] are used to this end, for example for shadow segmentation [9]. In the HSV color space, hue and saturation represent the two chromaticity components, while the third component, value, represents brightness. Since global brightness normalization is easily performed by for example mean and variation normalization, we adopt HSV as a new color space to estimate color flow unaffected by shadows and highlights.

The model is retrained on hue and saturation components only and expresses changes in hue and saturation between two images in terms of the new model. The vector field ϕ in (3) is now $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ and we only work with color components. We have experimented with estimation of the full trichromatic color space HSV based on values for hue and saturation. The vector field stored as a third component the relative change in the brightness component. Results however were not very stable. Results for the chromaticity estimation on 128×128 image patches are displayed in figure 1(b). We clearly obtain good estimation results, even in presence of disturbing shadows. Note that the size of eigenflows is now reduced from 12288 to $16^2 \times 2 = 512$ rows.

A. HSV issues

Two issues appear with the use of HSV. First of all, hue is defined as an angle and its space is thus circular. In particular, there always exist two difference measures between two hue values since we can go either clockwise or counterclockwise around the color circle. The hue circle is designed so that similar angles represent similar colors, so it makes sense to define the difference in hue space to be one with shortest absolute value. Differences are signed and have range $[-0.5, 0.5]$. We have to account for this effect during model training, in the calculation of the full color flow in (3). But also for parameters estimation when calculating the difference between starting and target image. Finally, when evaluating results by means of error calculation we have to take shortest distances in the hue components.

Secondly, singularities in the HSV color space have to be considered. Hue is undefined on

the achromatic axis, where saturation is zero. Both hue and saturation are undefined when brightness is zero. As has been investigated by Kender [10], the components are unstable close to those singularities. Gevers *et al.* [3] investigated noise propagation through the calculation of photometric invariant color spaces, among others hue and saturation, and proposes to suppress the effect of it by applying a variable kernel density estimator.

In training and estimation we are only concerned with instabilities in hue, as brightness is not being considered and the HS space is a solid circle instead of a cone. The smoothing effect of (3) ensures stable hue values near zero saturation in the eigenflows as also color changes with higher saturation contribute to such values. However there might be some effect in estimation and evaluation. A common approach is to simply ignore values close to singularities. We tested this technique to error calculation, with several thresholds on the distance to the achromatic axis, but found results to differ only marginally. Thus the instability of the color space is not an issue here.

B. Dimension selection

We will now come back to the issue of choosing a suitable number of eigenflows. As discussed in section II-A, the eigenvalues associated with the eigenflows are a measure for the variance in that direction. The total variance in the distribution is obtained by summing up the eigenvalues λ_i for all eigenflows, $1 \leq i \leq M$ with M the number of eigenflows. In our case the dimension of the eigenflows is much larger than the number of samples and M is equal to this number of samples (provided the samples are mutually orthogonal). Now the *explained variance* $V(p)$ indicates the fraction of the total variance we retain by projecting the data on the subspace spanned by the first p principal components:

$$V(p) = \frac{\sum_{i=1}^p \gamma_i}{\sum_{i=1}^M \gamma_i}, \quad p \leq M. \quad (5)$$

Figure 2(a) shows the explained variance using the RGB model, figure 2(b) the equivalent with the HS model. The vertical axis has range $[0.8, 1]$. We immediately notice the much faster total ascent of the graph to 100% variance in the HS model. For the RGB color space the ascent is initially fast too, but quickly changes to marginal ascent without reaching 100% variance within 100 eigenflows.

These results are confirmed by considering the (root mean squared) error in estimation for both color spaces as a function of the number of eigenflows used, as displayed in figure 3.

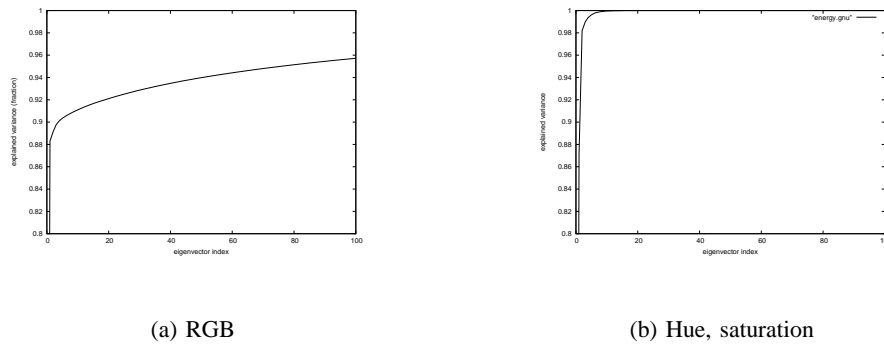


Fig. 2. Explained variance

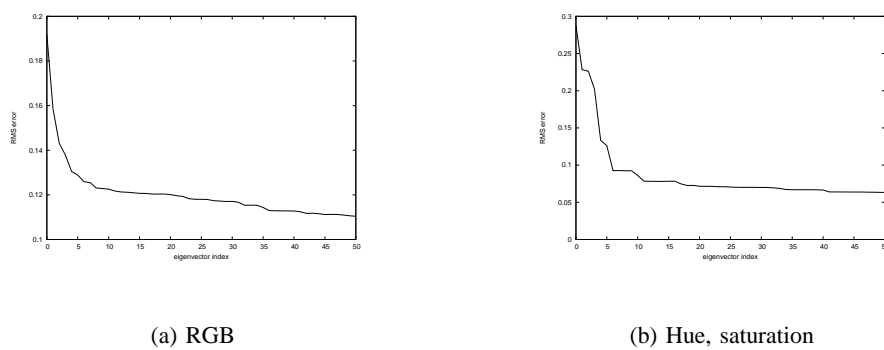


Fig. 3. Error as a function of number of eigenflows used

The pair of test images contained more than ten toys of very different colors, both saturated and unsaturated, and included numerous shadows and highlight effects. Note that the error is measured in the same space as estimation. In the HS color space the graph remains relatively horizontal after an initial fast decrease in error during the first 6 eigenflows. The RGB graph on the contrary keeps descending slowly and it remains useful to add more eigenflows to the estimation process. From both HS graphs we conclude that 6 eigenflows suffice to deal with virtually all possible variance. The color flow model in the RGB color space needs a higher dimension to explain the same fraction of variance and also lacks a clear point on the curve after which additional eigenflows become useless. We understand that the comparison is not entirely fair as the HS space does not perform brightness estimation. Nevertheless, results are clearly much better in the HS case as can be seen from figure 1.

IV. REFERENCE REGIONS

We propose to combine the color flow algorithm with techniques for achieving viewpoint invariancy. It will also allow for physical movement of objects and, to a certain extent, the (dis-)appearance of objects in the scene. The color flow method itself will remain largely unchanged, the input will be different and a robust estimation is used instead of least squares. Instead of providing it with two aligned images, corresponding pairs of normalized regions around detected keypoints are used as input. Below we will first describe the process of extracting corresponding regions and then discuss how flow parameters can be estimated robustly from this data.

A. Preprocessing

We are not interested in the structure of objects present in a scene but more in the changes in their color, so that we favor blob-like regions to corners. Those regions are provided by the Difference-of-Gaussian (DoG) detector [11]. Keypoints are detected in both color images using a single intensity band, which is a linear combination of the three RGB color bands. Descriptions for the detected keypoints are calculated in the same intensity band by SIFT [11], using the full 128 dimensions. Correspondences between keypoints in both images are obtained by finding *mutual* nearest neighbors in SIFT space. That is if keypoint $P_{1,j}$ in the first image has keypoint $P_{2,k}$ in the second image as its nearest neighbor, we demand that this also holds *vice-versa*: keypoint $P_{2,k}$ has $P_{1,j}$ as its nearest neighbor. The distance metric used can be the Euclidean distance in SIFT space. However the Mahalanobis distance is also used [12] and provided higher true matching rates in our case. The covariance matrix is experimentally determined from the descriptions of all detected keypoints.

B. Using color to improve matching accuracy

The SIFT descriptor was chosen because of its best performance under rotation, scale and affine transformations and second-best ranking for illumination changes [13]. Our tests on color images under illumination changes showed that performance was not sufficient, often not surpassing true positive rates of 50% on keypoint matching. This would clearly lead to too much false data for the flow parameters to be estimated correctly. Since we are using color images, it seems wise to use this valuable source of information to increase true positive rates (TPR). A secondary goal is to keep the absolute number of true positives high, in order

to retain as much information about color changes as possible for an accurate estimation of flow parameters.

The DoG detector provides us with a keypoint location, orientation and scale allowing us to define a support region centered on the keypoint. This support region can be transformed into a normalized region by appropriate rotation and rescaling. The size of these normalized regions is predetermined and set to 128×128 in our experiments. Note that rescaling only performs 'zoom-in' and excludes 'zoom-out'. The smoothing required for 'zoom-out' introduces new colors and we have found the color flow algorithm to be very sensitive to this. For the same reason, no interpolation is performed for 'zoom-in' and color from the nearest neighbor is copied. This effectively forces an upper limit on keypoint scales and we also set a lower limit so that rescaling is never more than a factor 10. These thresholds ranges are wide enough to reject only a small number of keypoints.

We can now perform color histogram matching [14] on pairs of normalized regions as found by the mutual SIFT match. Since we are dealing with illumination changes, we will use color ratio histograms [4] where the ratios of neighbor pixels are histogrammed instead of their absolute value. This provides some robustness to slowly varying illumination.

The specific application to normalized regions requires two adaptations. As we know from [4], the distribution of color ratios is not linear and histogram bin boundaries have to be adapted accordingly. Since normalized regions are acquired from keypoints with differing scales, the range of color ratios will differ considerable from region to region as keypoints with small scales are less likely to display strong changes in color. The maximum color ratio, and thus the bin boundaries, is therefore determined dynamically for every pair of normalized regions. Ratio histogramming is performed in the two dimensional chromaticity space and separate scales are used for the dimensions. Secondly, since there is no interpolation in the rescaling in normalization process, many color ratios will have a value of 1. This is of course especially true for the smallest scales with the highest zoom factor. Common zoom factors of 5 to 10 result in 80% to 90% of ratios with value 1. Those values thus have a severe weight in the histogram, making the 'true' color ratios virtually irrelevant. We solve for this by bluntly removing all values of 1, as only very few 'true' ratios have exactly value 1. Both adaptations strongly improve performance in obtaining high true positive matching rates.

Once two histograms have thus been created, their normalized intersection provides us with a match probability. This probability can be either thresholded, or used to rank the pairs of normalized regions after which the best N pairs are selected for further processing. The

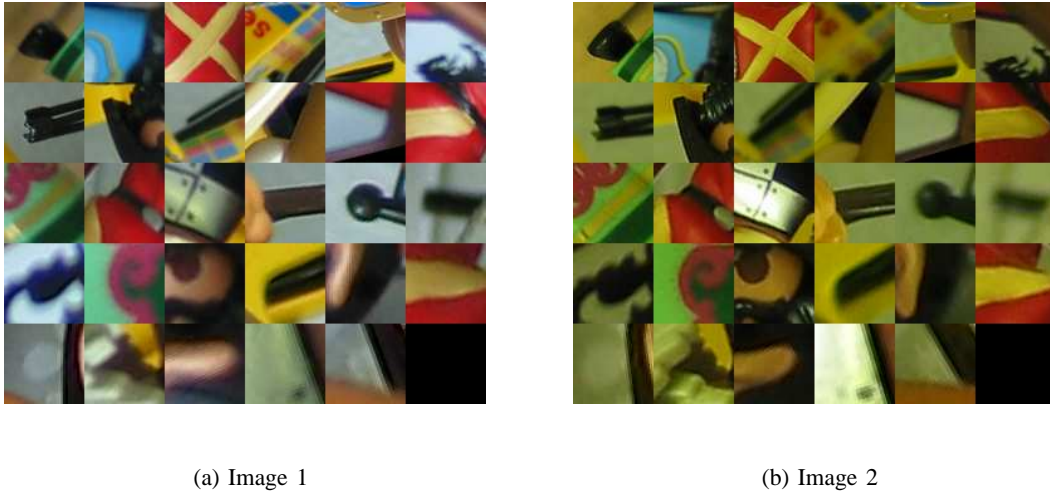


Fig. 4. Corresponding normalized regions

higher the threshold, the higher the true positive rate but also the lower the absolute number of true matches. Furthermore, two remarks are of importance. Firstly, true corresponding keypoints will still have deviations in their parameters location, rotation and scale. This leads to misalignment in the normalized regions and these are propagated into the histogram and match probability. This effect is likely to be the reason why experiments with the χ^2 -test failed, although this test performs reportedly better [15]. Secondly, the color ratio histogram method assumes a diagonal model for color change, where each color band is scaled by a factor. The color flow model is clearly much more powerful and care should be taken not too strictly remove matches that do not comply with the simplistic diagonal model.

C. Robust estimation

Figure 4 displays corresponding normalized regions extracted from a certain test pair. We intend to estimate the color flow observed from the ensemble of these normalized regions in terms of our model.

A closer scrutiny of figure 4 reveals the misalignments in corresponding normalized regions due to deviations in keypoint parameters. Due to the blob-like nature of regions, they contain extensive subregions of similar colors. This ensures that the effect of misalignments is limited, an important reason to prefer the DoG detector. Nevertheless, the deviations come up as high magnitudes in the difference image and have, similar to moving shadows, important effect on the estimation process. Furthermore, some false positive matches might remain after



Fig. 5. Images and result from the Newspaper set.

preprocessing. To reliably estimate flow parameters, a robust estimation method is required that is able to detect false color changes (outliers) and eliminate their influence on parameter estimation.

We have tested RANSAC, MDPE and M-estimation [16]. Best results were obtained by using an M-estimator with the biweight loss-function. Initialization was done by non-robust least squares as performance was on average similar but more stable to initialization by RANSAC or MDPE. Inlier scale estimation was performed during the first 3 iterations by the Median Absolute Deviation, a much more robust estimation of scale than the standard deviation.

Experiments on a synthetically generated target image have shown that while outlier fraction rises to high levels with (synthetic) increasing noise in keypoint location, orientation and scale, the M-estimator performed remarkably robust with estimating errors up to 0.02 for very large deviations in keypoint parameters. Results on real images are presented in section V.

1) Memory concerns: Robust estimation on multiple corresponding normalized regions takes a lot of memory and time, making the calculation not feasible for more than 30-50 regions. To be able to use all matched regions, we have investigated an approach in which parameters are estimated from individual pairs. Viewing these solutions as points in the parameter space, we expect good estimations to lie close together, while bad estimation or estimations based on false positive matches lie arbitrarily far away. A Mean Shift [17] approach is then employed to find the center of highest density, corresponding to the cluster of good estimations.

We have found however that solutions to (true positive) matches can differ widely. This is

furthermore aggravated by regions lacking enough variation in colors to define all parameters used in estimation, *degenerate* regions. In these cases outliers might be used to 'break' the degeneracy and help define parameters, giving them arbitrary values. It is clear that under these circumstances the Mean Shift approach will not work.

V. RESULTS

The experimental results presented in this section substantiate two key points. First, the color flow method is superior to standard color change models. Secondly, estimating global color change from reference regions as opposed to full aligned images does not decrease results. The local method is robust enough and does not invoke a penalty on the estimation results so that we obtain viewpoint invariance for free.

A. Training and test sets

For the experiments shown in this article, we have created two pairs of training and test images, one, called 'Tapis', based on a children's book containing many colors in complicated structures, another of newspapers with color images spread out in a random fashion on the floor, called 'Newspapers'. For each pair, two sets (of similar but different scenes) of 10 images were taken under different illumination conditions. Images were taken in a normal office, where the amount of daylight could be adjusted by controlling blinds and fluorescent lights could be turned on and off. Figure 5 shows examples of the Newspaper set.

B. Experiments

The color flow model was trained twice separately on 4 aligned image pairs taken from the first sequence of both sets. From this we obtain 4 eigenflows, which were all used for estimation and the model has 4 parameters. The estimation was done on six pairs from the second sequence. For comparison we ran three other color change models. The Gray World model matches the separate color band means from the first image to the second image and derives the three scaling factors from it (3 parameters). The Diagonal model estimates these scaling parameters directly (also 3 parameters), while the affine model additionally estimates offsets in each color band (6 parameters). The last two models can be estimated robustly in the same way as the color flow model. In these cases 30 reference regions are extracted from the training images.

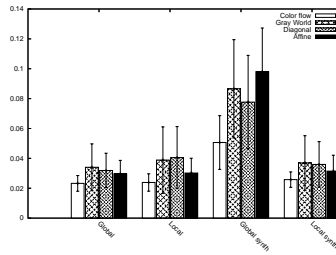


Fig. 6. Experimental results: comparing different methods applied on different datasets (see text for details).

As an error measurement we consider the error in each component of the full image and calculate the Root Mean Squared error in HS space. This measures the model's performance on all relevant colors, as opposed to the method employed by [7] that only calculates the error in the chromatic components of a white patch. The first two histograms in figure 6 show mean RMS error and the standard deviation for each of the four methods. The cluster on the left is obtained by estimating directly from the full image, while the second cluster is the results of estimation from 30 local corresponding regions (the same robust estimation is used for each method). Comparing these two histograms we find that results remain the same for the color flow and affine model, but get somewhat worse for the gray world and diagonal model. Note that the gray world model can not be estimated robustly. The color flow model outperforms all others in both cases.

To show the invariance to changes in the scene composition, we added in another experiment a randomly colored block to the target image and estimated the parameters. The block had a size of 40% the image size in both dimensions, such that it covered a surface of only 16%. For each pair, 20 such synthetic target images were generated. Errors are calculated in respect to the original, block-less target image and results are displayed in the two histogram clusters on the right of figure 6. Even though the block is not too large, estimation on the global image is severely disturbed by the block's presence. However results of the local method are unaffected.

Figure 5(c) visualizes the performance of the method when the scene changes. The scenes are different due to the reshuffling of newspapers, but even so our approach is able to catch the color change.

VI. CONCLUSION

In this article we demonstrate how the assumption of having a few known reference elements in the scene allows to compensate easily and efficiently for illuminant changes. In our approach the effect of illumination changes on the reference objects is modeled in a training stage. Subsequently, a few correspondences between these reference objects and some canonical views of these objects allow to estimate accurately the dense set of color changes. Experimental results on real images show the efficiency of this method.

ACKNOWLEDGMENT

The authors would like to thank... This work was supported by the IEEE.

REFERENCES

- [1] E. Day, R. Berns, L. Taplin, and F. Imai, "A psychophysical experiment evaluating the color and spatial image quality of several multispectral image capture techniques," *J. Imaging Science and Technology*, vol. 48, no. 2, pp. 93–104, 2004.
- [2] J. Geusebroek, R. van den Boomgaard, A. Smeulders, and H. Geerts, "Color invariance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 12, pp. 1338–1350, December 2001.
- [3] T. Gevers and H. Stokman, "Robust histogram construction from color invariants for object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 1, pp. 113–118, 2004.
- [4] B. Funt and G. Finlayson, "Color constant color indexing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 5, pp. 522–529, 1995.
- [5] T. Gevers and A. W. M. Smeulders, "Color based object recognition," *Patt. Rec.*, vol. 32, pp. 453–464, Mar. 1999.
- [6] E. Miller and K. Tieu, "Color eigenfbws: Statistical modeling of joint color changes," in *Proc. 8th Int'l Conf. Comp. Vision*, vol. 1, 2001, pp. 607–614.
- [7] S. Obdrzalek, M. Matas, and O. Chum, "On the interaction between object recognition and colour constancy," in *IEEE Workshop on Color and Photometric Methods in Computer Vision*, 2003.
- [8] K. Tieu and E. Miller, "Unsupervised color constancy," *Advances in Neural Information Processing Systems 15.*, 2003.
- [9] E. Salvador, A. Cavallaro, and T. Ebrahimi, "Cast shadow segmentation using invariant color features," *Comp. Vision and Image Understanding*, vol. 95, pp. 238–259, 2004.
- [10] J. Kender, "Saturation, hue, and normalized color: calculation, digitization effects, and use," Master's Thesis, 1976.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. Comp. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] T. Tuytelaars and L. V. Gool, "Matching widely separated views based on affine invariant regions," *Int'l J. Comp. Vision*, vol. 59, pp. 61–65, Aug. 2004.
- [13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *Proc. Conf. Comp. Vision and Patt. Rec.*, vol. 2, June 2003, pp. 257–263.
- [14] M. Swain and D. Ballard, "Color indexing," *Int'l J. Comp. Vision*, vol. 7, pp. 11–32, 1991.
- [15] B. Schiele and J. Crowley, "Object recognition using multidimensional receptive field histograms," in *Proc. 4th Eur. Conf. Comp. Vision*, 1996, pp. 610–619.

- [16] R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*. Academic Press, 2004.
- [17] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, May 2002.