

Channel Representation of Colour Images

Report LiTH-ISY-R-2418

Per-Erik Forssén, Gösta Granlund and Johan Wiklund

Computer Vision Laboratory, Department of Electrical Engineering

Linköping University, SE-581 83 Linköping, Sweden

March 26, 2002

Abstract

In this report we describe how an RGB component colour image may be expanded into a set of channel images, and how the original colour image may be reconstructed from these. We also demonstrate the effect of averaging on the channel images and how it differs from conventional averaging. Finally we demonstrate how boundaries can be detected as a change in the confidence of colour state.

1 Introduction

There are several advantages with separation of the brightness and colour information in image processing. The human visual system (HVS) also seems to have reached this conclusion since it has different pathways for colour and structural information [2]. Perceptual experiments indicate that the HVS has a much higher sensitivity to changes in brightness than in colour. This is exploited for instance in image transmission and compression, where colour information is stored and transmitted at half the resolution of the brightness [6].

There are many different ways to separate brightness and colour, here we will use a representation of *luminance* and *chrominance*. The luminance is a scalar value that corresponds to a weighted summation of the intensities in the colour bands. The weights are chosen to give an image which corresponds to the perceived brightness in the human visual system. The chrominance is a vector with the argument corresponding to the hue, and the magnitude corresponding to how confident we are in this colour statement. A high confidence should correspond to a high degree of both saturation and signal strength.

The idea of having the magnitude of the chrominance vector correspond to a certainty makes vector field averaging meaningful [4]. Colour averaging is necessary for instance in scale space decomposition [9]. A related approach is to perform a normalized averaging of colour, where the colour samples are weighted with a relevance statement. This is done in [9], with the luminance as a colour relevance. Note however that when the vector magnitude represents the relevance, no explicit weighting is necessary.

2 Luminance and Chrominance

The pixel value vector of a colour image I at position $\mathbf{p} = (p_x \ p_y)^T$ consists of red, green, and blue components:

$$\mathbf{I}(\mathbf{p}) = (I_r(\mathbf{p}) \ I_g(\mathbf{p}) \ I_b(\mathbf{p}))^T \quad (1)$$

We can compute a luminance image as a pointwise projection of the pixel value vector onto a vector \mathbf{w}_l :

$$I_l(\mathbf{p}) = \mathbf{w}_l^T \mathbf{I}(\mathbf{p}) \quad \text{where} \quad \mathbf{w}_l^T = (0.2989 \ 0.5866 \ 0.1145) \quad (2)$$

If the RGB image is gamma corrected, this projection gives us the luminance component, as defined by CIE in [1]. This luminance is used in the YCrCb colour space, and by NTSC and PAL television standards. A colour image, and the corresponding luminance image are shown in figure 1.



Figure 1: A colour image and its luminance component.

The chrominance is represented as a vector \mathbf{I}_c , which is computed using another projection:

$$\mathbf{I}_c(\mathbf{p}) = \mathbf{W}_c^T \mathbf{I}(\mathbf{p}) \quad \text{where} \quad \mathbf{W}_c^T = \begin{pmatrix} \cos(0) & \cos(2\pi/3) & \cos(4\pi/3) \\ \sin(0) & \sin(2\pi/3) & \sin(4\pi/3) \end{pmatrix} \quad (3)$$

The rows of the matrix \mathbf{W}_c correspond to three symmetrically arranged unit vectors in \mathbb{R}^2 as shown in figure 3. This makes our chrominance image different from the one in YCrCb, which has a slight asymmetry in the colour vector arrangement. Figure 2 shows a colour representation of the \mathbf{I}_c vectors, and an image of their magnitudes. In the colour image the intensity corresponds to the magnitude, and the hue corresponds to the argument of \mathbf{I}_c .

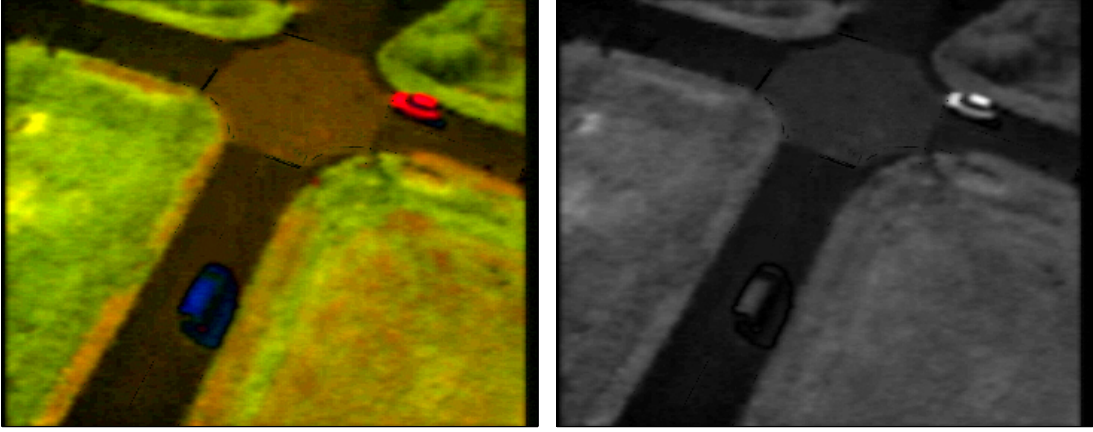


Figure 2: Visualisation of the chrominance component, and its magnitude.

Due to the symmetrical arrangement of the colour vectors, a scaling of the input image \mathbf{I} will be propagated to the chrominance vector \mathbf{I}_c , but an additive offset will not change the chrominance vector at all. This means that we can only reconstruct \mathbf{I} from \mathbf{I}_c , up to an additive offset in each point. This offset can later be restored using the luminance image I_l .

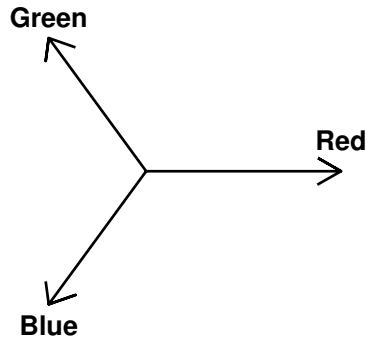


Figure 3: Modular representation of chrominance.

Since the vectors in \mathbf{W}_c are symmetrically arranged, they constitute a tight frame for the chrominance subspace. The inverse frame operator is thus a scaling of \mathbf{W}_c^T :

$$\mathbf{W}_c^\dagger = \frac{2}{3} \mathbf{W}_c^T \quad (4)$$

By \mathbf{I}_0 we denote the image obtained by applying \mathbf{W}_c^\dagger to \mathbf{I}_c . Figure 4 shows \mathbf{I}_0 with a constant offset added to avoid negative intensities.

As stated earlier the \mathbf{I}_0 image equals the original image \mathbf{I} plus an unknown offset $\epsilon(\mathbf{p})$:

$$\mathbf{I}_0(\mathbf{p}) = (\mathbf{W}_c^\dagger)^T \mathbf{I}_c(\mathbf{p}) = \mathbf{I}(\mathbf{p}) + (1 \ 1 \ 1)^T \epsilon(\mathbf{p}) \quad (5)$$

This means that the projection of \mathbf{I}_0 onto \mathbf{w}_l can be written as:

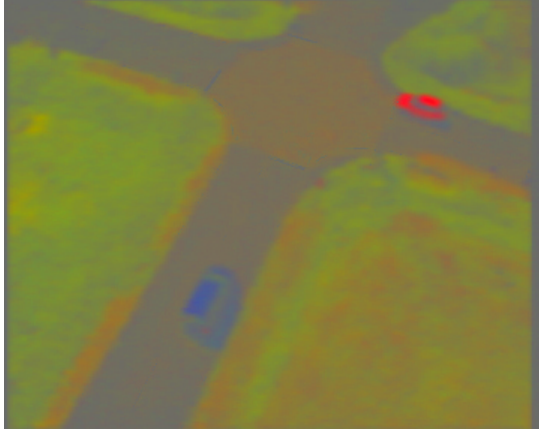


Figure 4: Image \mathbf{I}_0 before luminance and saturation adjustment.

$$\mathbf{w}_l^T \mathbf{I}_0(\mathbf{p}) = \mathbf{w}_l^T \mathbf{I}(\mathbf{p}) + \mathbf{w}_l^T \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T \epsilon(\mathbf{p}) = I_l(\mathbf{p}) + \epsilon(\mathbf{p}) \quad (6)$$

since the coefficients in \mathbf{w}_l sum to 1. We can now find the offset ϵ in each point, as:

$$\epsilon(\mathbf{p}) = \mathbf{w}_l^T \mathbf{I}_0(\mathbf{p}) - I_l(\mathbf{p}) \quad (7)$$

and reconstruct the image \mathbf{I} as:

$$\mathbf{I}(\mathbf{p}) = \mathbf{I}_0(\mathbf{p}) - \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T \epsilon(\mathbf{p}) \quad (8)$$

3 Channel Representation

The luminance and chrominance images obtained in section 2 can be expanded into the *channel representation* [5, 7]. For the luminance we will use a linear channel representation with padding channels placed outside the interval we want to represent, as is illustrated in figure 5.



Figure 5: Positions of channels representing an intensity interval.

For an encoding with K channels, using an overlap of $\omega = \pi/N$ we should map the desired interval $[r_l, r_h]$ into the interval $[1 + (N - 2)/2, K - (N - 2)/2]$. This can be done using a mapping:

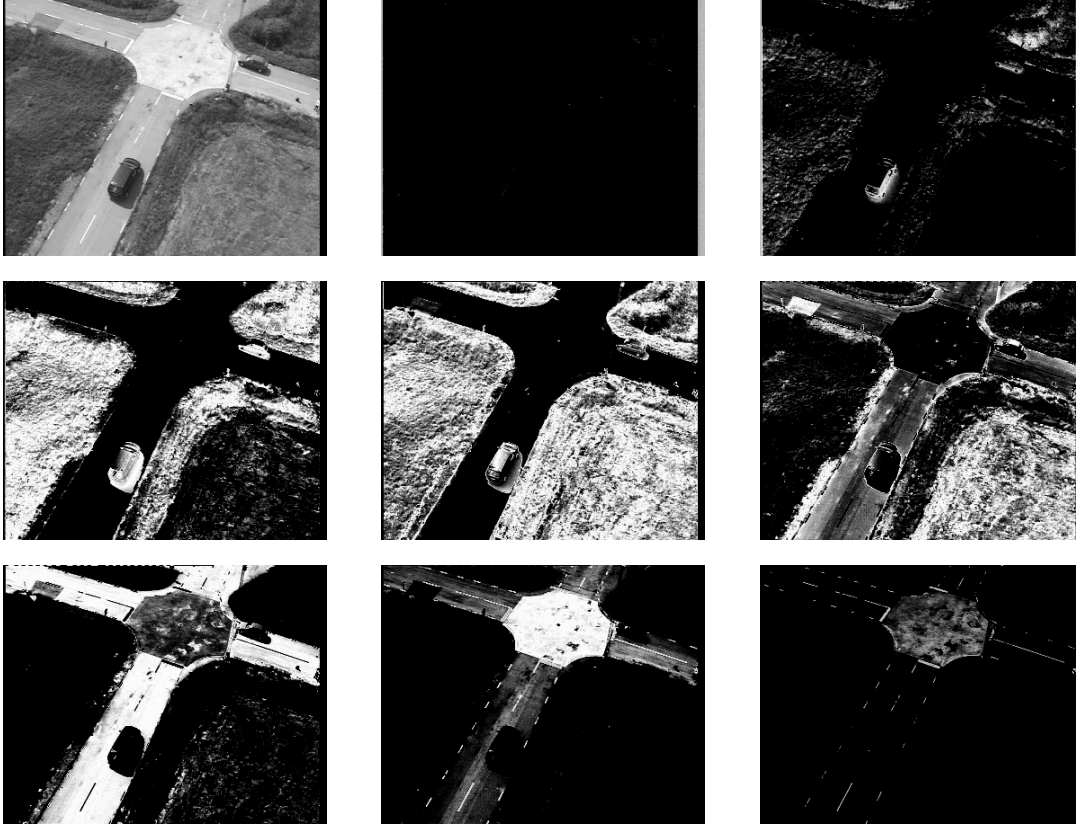


Figure 6: Luminance image (top left) and eight luminance channels.

$$x(\mathbf{p}) = sI_l(\mathbf{p}) - t \quad (9)$$

Where the scaling s , and translation t , are computed as:

$$s = \frac{K - N + 1}{r_h - r_l} \quad \text{and} \quad t = sr_l - N/2 \quad (10)$$

The channel vector in each position is now defined as:

$$\mathbf{x}(\mathbf{p}) = (\varphi_1(x(\mathbf{p}), 1) \quad \varphi_2(x(\mathbf{p}), 1) \quad \dots \quad \varphi_K(x(\mathbf{p}), 1))^T \quad (11)$$

where $\varphi_k(x, r)$ is the channel envelope function:

$$\varphi_k(x, r) = \begin{cases} r \cos^2(\omega(x - k)) & \text{when } |x - k| \leq \frac{\pi}{2\omega} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The parameter r in the channel encoding signifies the relevance of the current pixel. With this formulation, operations on irregularly sampled images, and images with missing

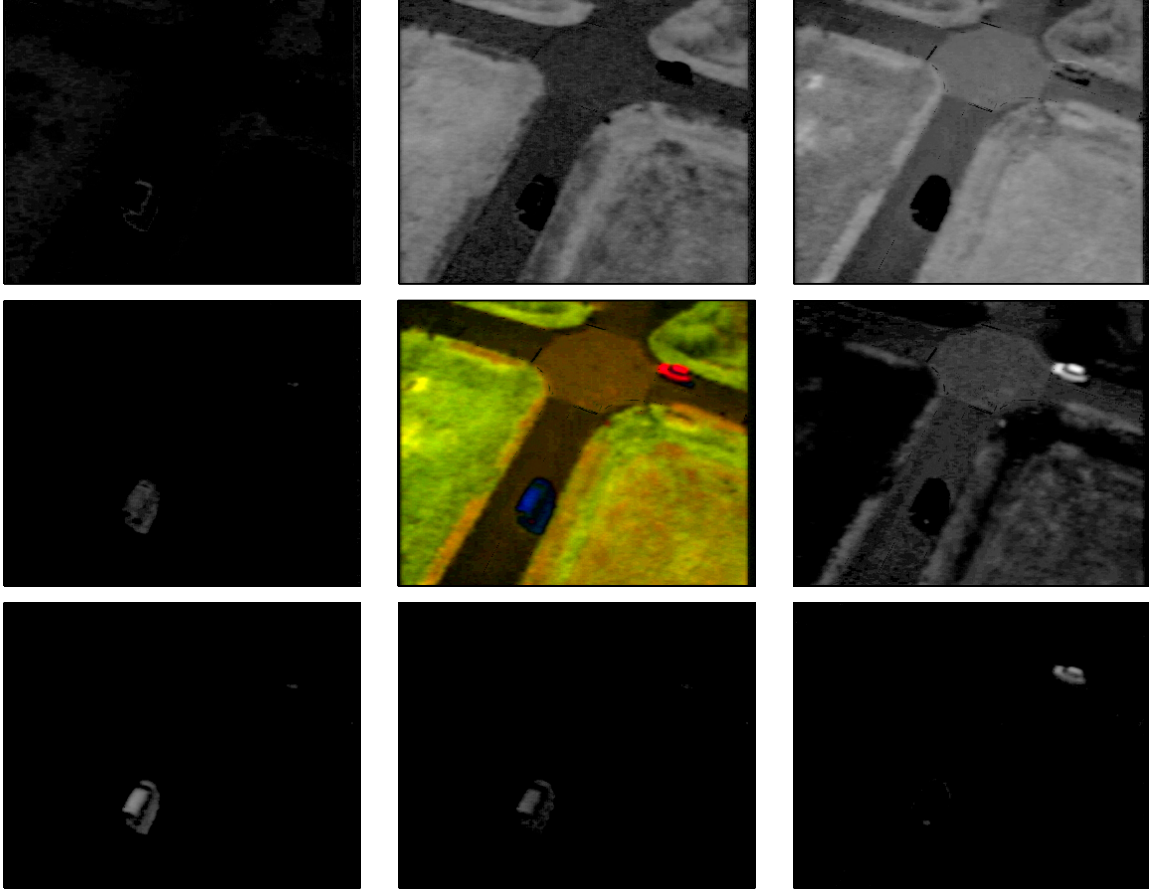


Figure 7: Chrominance image (centre) and eight chrominance channels.

data, can be performed in the same manner as when the sampling is dense (see section 5). A channel encoding of a luminance image into $K = 8$ channels is shown in figure 6.

The chrominance image is expanded into channels using a modular channel encoding:

$$\varphi_k(x, r) = \begin{cases} r \cos^2(\omega(x - k)) & \text{when } d_K(x, k) \leq \frac{\pi}{2\omega} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$\text{where } d_K(x, k) = \frac{K}{2} - \left| (x - k) \bmod K - \frac{K}{2} \right|$$

The chrominance image \mathbf{I}_c is mapped to x , and r using:

$$x = \arg \left[\begin{pmatrix} 1 & \mathbf{i} \end{pmatrix} \mathbf{I}_c \right] \frac{K}{2\pi} + 1 \quad \text{and} \quad r = (\mathbf{I}_c^T \mathbf{I}_c)^{\frac{1}{2}} \quad (14)$$

A channel encoding of a chrominance image into $K = 8$ channels is shown in figure 7. The position of the channels in the figure correspond to the hue representation in figure 3.

4 Reconstruction

To go back from the channels to the luminance and chrominance images, we will make use of the local reconstruction developed in [3]. We start by describing the reconstruction for the luminance image.

Each reconstruction hypothesis is computed from a consecutive group of channel values, for instance $\{x_1, x_2, x_3\}$. The index of the first channel in the group is denoted l . For overlaps of $\omega = \pi/N$ where $N = 3, 4, 5 \dots$, the number of channels related to a specific scalar becomes¹ N , and we can compute a reconstruction hypothesis \hat{x} as a local weighted summation of complex exponentials:

$$\hat{x}(\mathbf{p}) = l + \frac{1}{2\omega} \arg \left[\sum_{k=l}^{l+N-1} x_k(\mathbf{p}) e^{i2\omega(k-l)} \right] \quad (15)$$

The relevance measure embedded in the channel representation is retrieved as:

$$\hat{r}(\mathbf{p}) = \frac{2\omega}{\pi} \sum_{k=l}^{l+N-1} x_k(\mathbf{p}) \quad (16)$$

In each point \mathbf{p} , we try all consecutive groups of channels, i.e. we let l vary in $[1 \dots K - N + 1]$, and select the reconstruction which has the highest relevance measure $\hat{r}(\mathbf{p})$. We then map the result back to the correct luminance range, using the inverse of the mapping in equation 9:

$$\hat{I}_l(\mathbf{p}) = \frac{1}{s} \hat{x}(\mathbf{p}) + \frac{t}{s} \quad (17)$$

Where the coefficients s , and t are those defined in equation 10.

When reconstructing the chrominance, we will use the same approach to obtain hypotheses, but we will also consider groups of channels that are neighbours modulo K . For $\omega = \pi/3$ this means that we also have to consider the groups $\{x_K, x_1, x_2\}$ and $\{x_{K-1}, x_K, x_1\}$.

The chrominance vector $\hat{\mathbf{I}}_c(\mathbf{p})$ is then computed from the hypothesis $\hat{x}(\mathbf{p})$ with the highest relevance $\hat{r}(\mathbf{p})$ as follows:

$$\hat{\mathbf{I}}_c(\mathbf{p}) = \hat{r}(\mathbf{p}) \left(\cos\left((\hat{x}(\mathbf{p}) - 1)\frac{2\pi}{K}\right) \quad \sin\left((\hat{x}(\mathbf{p}) - 1)\frac{2\pi}{K}\right) \right)^T \quad (18)$$

The final reconstructed colour image $\hat{\mathbf{I}}(\mathbf{p})$ can now be computed using equations 5, 7, and 8.

¹Actually the number of related channels becomes $N - 1$ at regularly occurring locations, but this does not affect the resultant reconstruction.

5 Channel Averaging

A classical problem in image processing is that of image enhancement [4]. In image enhancement we want to suppress noise and at the same time enhance the image structure. Clearly this cannot be accomplished by plain averaging, since a high degree of noise reduction implies averaging over a large region, and this is guaranteed to remove the fine details of the image.

If we perform averaging of the channel decomposition presented in this report however, we are able to average homogeneous regions, and still keep edges. Figure 8 demonstrates how the number of channels controls how steep the edges have to be in order to be kept. For 3 channels the method is equivalent to plain averaging—it always performs smoothing. As we increase the number of channels however, increasingly small edges will be kept. Apparently the method has a behaviour similar to image enhancement—homogeneous regions are smoothed, but edges which are steeper than a given distance (the *metameric distance* [3]) are kept.

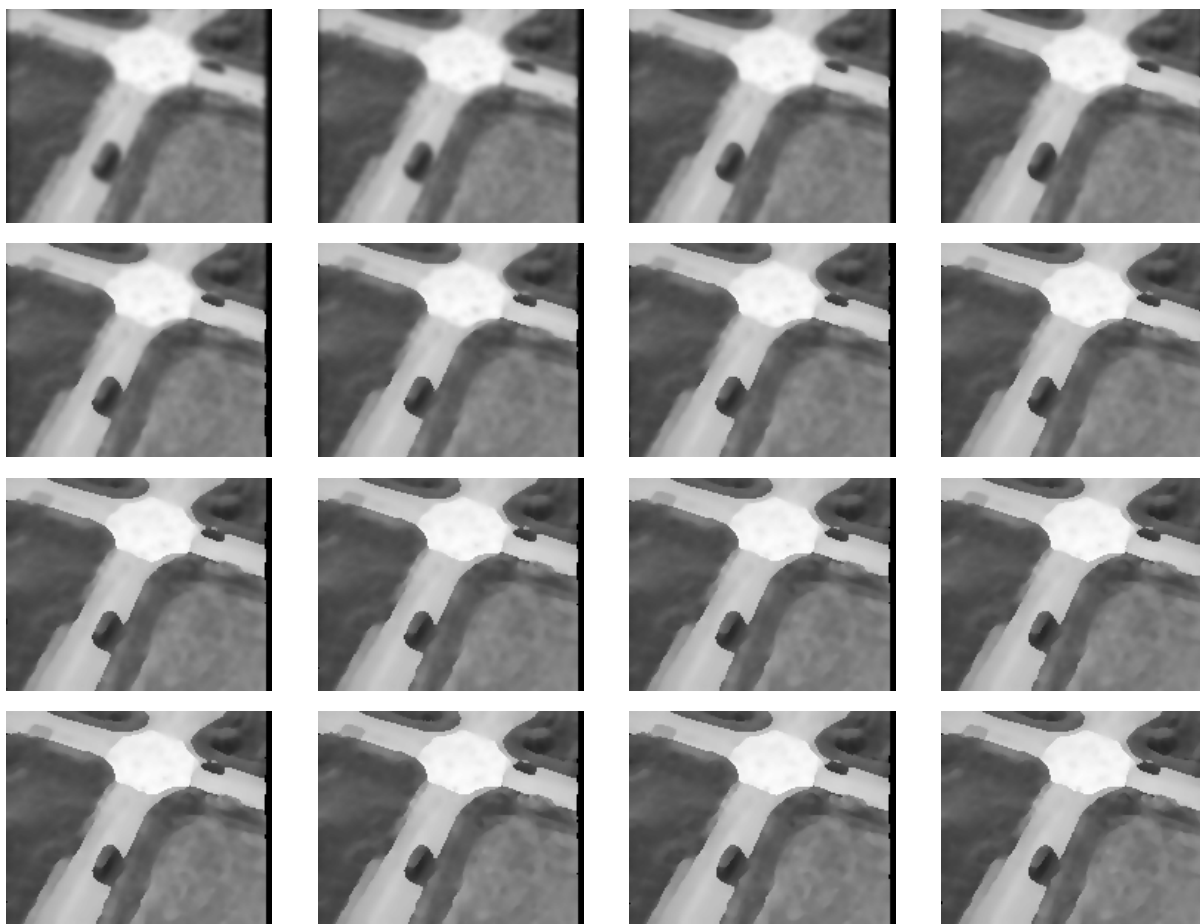


Figure 8: Varied number of channels.

left to right, top to bottom: Number of channels varied from 3 to 18. All images use $\sigma = 3.86$.

A side effect (which is not visible in these images) is that there is a slight dependency on alignment of the grey levels to the channel grid when the reconstruction decides whether to average or keep an edge, i.e. the metameric distance is intensity variant.

The effect of channel averaging is somewhat similar to non-linear Gaussian filtering [8, 11]. The channel decomposition can be seen as a means of making a filter, acting both spatially and intensity wise, separable. Differences between the approaches are that the intensity window here is a \cos^2 instead of a Gaussian, and that channel averaging appears to be less sensitive to outliers. However, the non-linear Gaussian filter does not have the channel alignment problem.

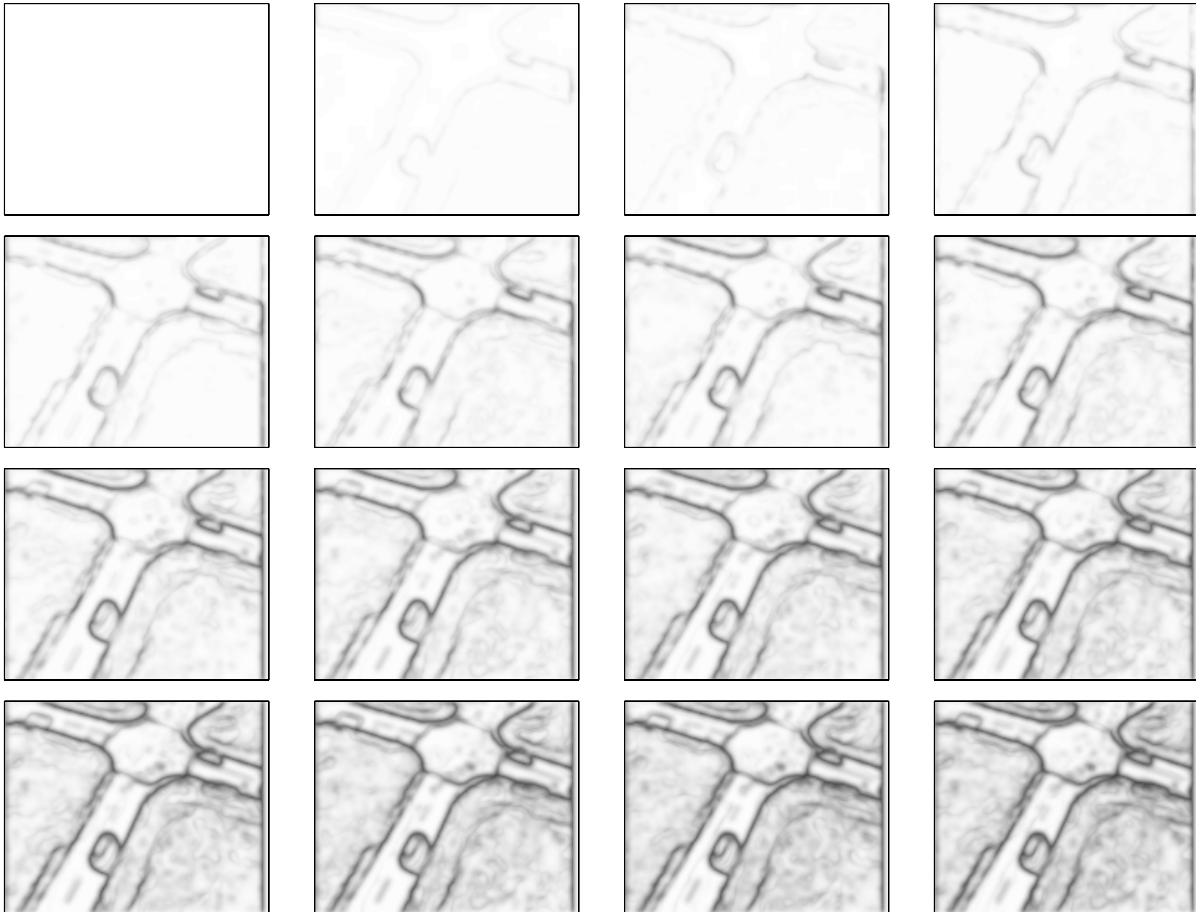


Figure 9: Boundary confidence measure under varied number of channels. *left to right, top to bottom: Number of channels varied from 3 to 18. White is zero, black is 1. All images use $\sigma = 3.86$.*

6 Boundary Detection

The regions where channel averaging differs from plain averaging can be identified by looking at how ambiguous the channel vector is in each point.

When no ambiguity in grey level is present, the channel value vector should contain only one valid reconstruction hypothesis $\hat{x}(\mathbf{p})$. In these situations, the channel vector sum should thus equal the local relevance measure sum in equation 16. Using this observation, we define a boundary confidence as the difference between the total channel value sum, and the largest local sum:

$$\hat{e}(\mathbf{p}) = \frac{2}{N} \left(\sum_{k=1}^K x_k(\mathbf{p}) - \max_l \left\{ \sum_{k=l}^{l+N-1} x_k(\mathbf{p}) \right\} \right) \quad (19)$$

Figure 9 shows this measure for the images in figure 8. By visual inspection, we can infer that, at least qualitatively, the boundary confidence measure indicates positions where edges are kept.

7 Dealing with Missing Data

In section 3 we mentioned that if we use the channel magnitude to represent a relevance, we can deal with missing data in a transparent manner. The classical example of operations on missing data is expansion of sparsely and irregularly sampled data to a dense grid. In the spirit of [4] we will now remove all but 10% of the pixels in the luminance image (see figure 10), and set the relevance of the remaining pixels to zero.

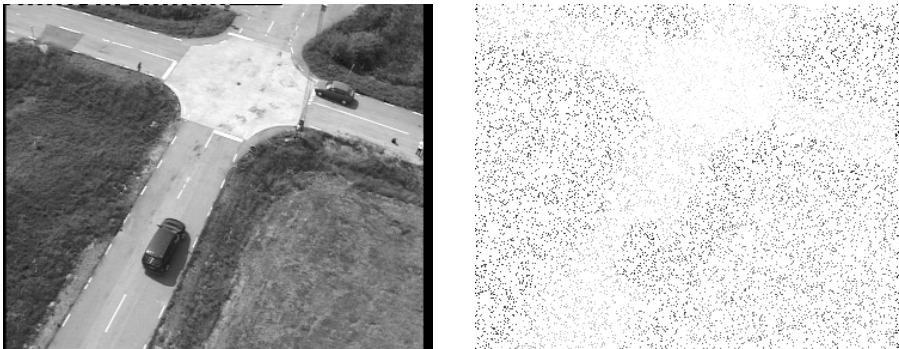


Figure 10: Sparsely sampled image.

Left: Original image. Right: Image with only 10% of pixels left.

We now compare averaging in channels followed by a reconstruction, with a method called *normalized averaging* [4, 10]. In normalized averaging, the resultant image is defined by the quotient:

$$\hat{I}_l(\mathbf{p}) = \frac{(I_l \cdot r * a)(\mathbf{p})}{(r * a)(\mathbf{p})} \quad (20)$$

Where $*$ denotes convolution, and a is an averaging kernel [4].

Figure 11 shows a comparison of normalized averaging and channel averaging with 14 channels. Both methods are applied to both the dense, and the sparse luminance images

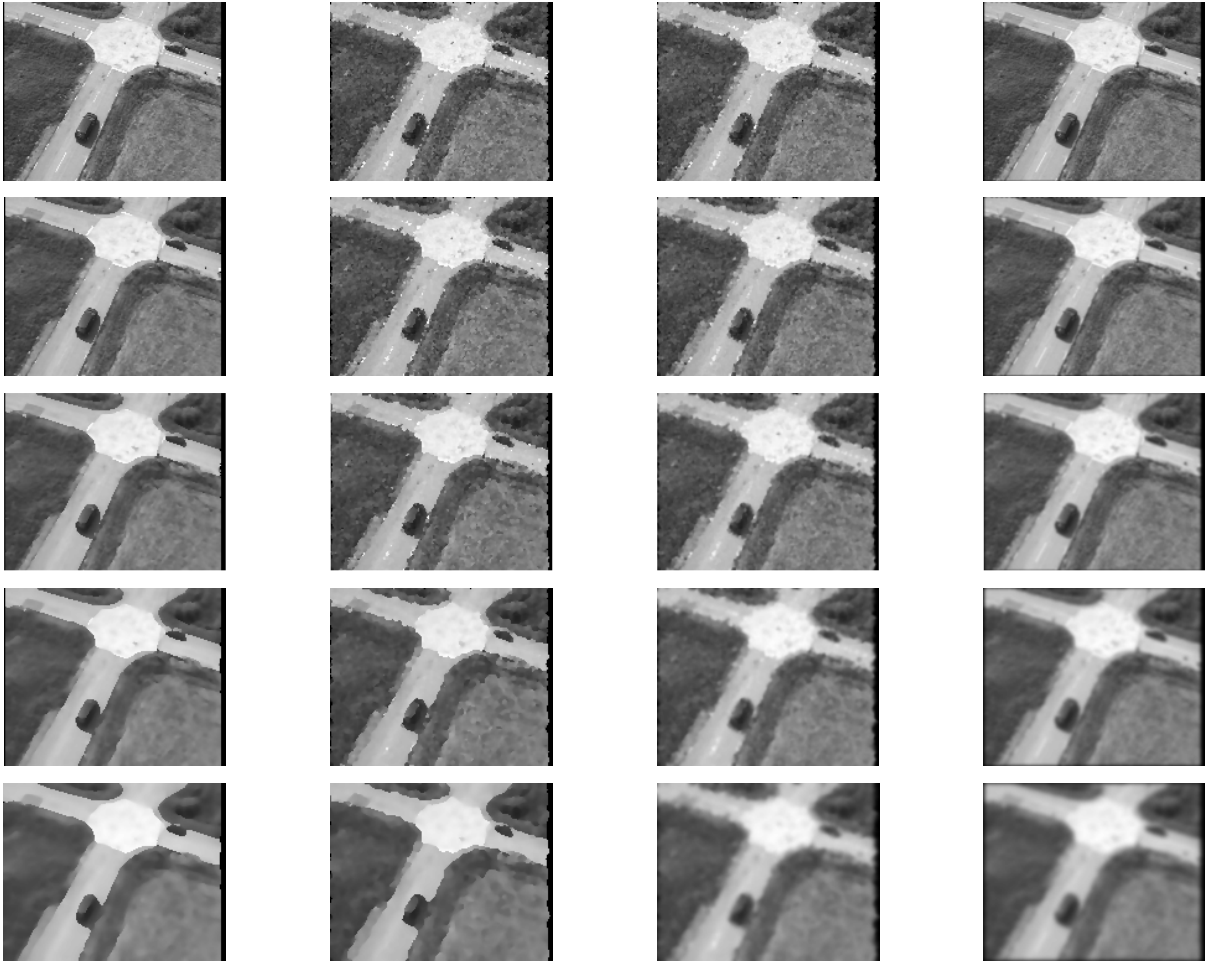


Figure 11: Comparison to normalized averaging.

Left to right: channel averaging on full image, channel averaging on sparsely sampled image, normalized averaging on sparsely sampled image, averaging on full image. Top to bottom shows varying degrees of blurring. A Gaussian with a support of 23×23 pixel has been used, with $\sigma = 0.69, 1.20, 1.83, 2.68, 3.86$. The channel averaging uses 14 channels.

in figure 10. The averaging is performed using a Gaussian kernel, with varying standard deviation σ . As can be seen, channel averaging and normalized averaging tend to have the same behaviour for small σ values. For higher degrees of smoothing however, normalized averaging tends to approximate blurring on the full image, whereas channel averaging on the sparse image tends to approximate channel averaging on the full image.

8 Concluding Notes

The goal of this report has been to demonstrate how colour images can be decomposed and reconstructed using the channel representation. An important aspect of the decomposition is the joint representation of statement and relevance, something which greatly simplifies meaningful averaging. Future research will attempt to make use of the presented decomposition for efficient image analysis.

Acknowledgements

The work presented in this report was supported by WITAS, the Wallenberg laboratory on Information Technology and Autonomous Systems, which is gratefully acknowledged.

References

- [1] Encoding parameters of digital television for studios. Technical Report CCIR Rec. 601-2, 1990.
- [2] M. F. Bear, B. W. Connors, and M. A. Paradiso. *Neuroscience: Exploring the Brain*. Williams & Wilkins, 1996. ISBN 0-683-00488-3.
- [3] P.-E. Forssén. Sparse Representations for Medium Level Vision. Lic. Thesis LiU-Tek-Lic-2001:06, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, February 2001. Thesis No. 869, ISBN 91-7219-951-2.
- [4] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.
- [5] G. Granlund, P.-E. Forssén, and B. Johansson. HiperLearn: A High Performance Learning Architecture. Technical Report LiTH-ISY-R-2409, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, January 2002.
- [6] K. N. Ngan, T. Meier, and D. Chai. *Advanced Video Coding: Principles and Techniques*. Elsevier Science B.V., 1999.
- [7] K. Nordberg, G. Granlund, and H. Knutsson. Representation and Learning of Invariance. In *Proceedings of IEEE International Conference on Image Processing*, Austin, Texas, November 1994. IEEE.
- [8] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
- [9] C.-J. Westelius and C.-F. Westin. A colour representation for scale-spaces. In *The 6th Scandinavian Conference on Image Analysis*, pages 890–893, Oulu, Finland, June 1989.
- [10] C.-F. Westin. *A Tensor Framework for Multidimensional Signal Processing*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1994. Dissertation No 348, ISBN 91-7871-421-4.
- [11] A. Wrangsjö, M. Borga, and H. Knutsson. Non-linear Gaussian Filtering for Image Resampling. In K. Åström, editor, *Proceedings SSAB02, Symposium on Image Analysis*, pages 65–68, Lund, March 2002.