

Automated Colour Grading using Colour Distribution Transfer

F. Pitié*, A. C. Kokaram, R. Dahyot

Trinity College Dublin, 2 College Green, Dublin, Ireland

Abstract

This article proposes an original method for grading the colours between different images or shots. The first stage of the method is to find a one-to-one colour mapping that transfers the palette of an example target picture to the original picture. This is performed using an original and parameter free algorithm that is able to transform any N -dimensional probability density function into another one. The proposed algorithm is iterative, non-linear and has a low computational cost. Applying the colour mapping on the original picture allows reproducing the same ‘feel’ as the target picture, but can also increase the graininess of the original picture, especially if the colour dynamic of the two pictures is very different. The second stage of the method is to reduce this grain artefact through an efficient post-processing algorithm that intends to preserve the gradient field of the original picture.

Key words: Colour Grading, Colour Transfer, re-Colouring, distribution transfer

* Corresponding author.

Email addresses: fpitie@mee.tcd.ie (F. Pitié), anil.kokaram@tcd.ie (A. C. Kokaram), dahyotr@tcd.ie (R. Dahyot).

1. Introduction

A major problem in the post production industry is matching the colour between different shots possibly taken at different times in the day. This process is part of the large activity of film grading in which the film material is digitally manipulated to have consistent grain and colour. The term colour grading will be used specifically to refer to the matching of colour. Colour grading is important because shots taken at different times under natural light can have a substantially different *feel* due to even slight changes in lighting.

Currently in the industry, colour balancing (as it is called) is achieved by experienced artists who use edit hardware and software to manually match the colour between frames by tuning parameters. For instance, in an effort to balance the red colour, the digital samples in the red channel in one frame may be multiplied by some factor and the output image viewed and compared to the colour of some other (a target) frame. The factor is then adjusted if the match in colour is not quite right. The amount of adjustment and whether it is an increase or decrease depends crucially on the experience of the artist. This is a delicate task since the change in lighting conditions induces a very complex change of illumination. It would be beneficial to automate this task in some way.

The technique proposed in this paper is an example-based re-colouring method. The idea, which has raised a lot of interest recently [1,2,3,4,5], is illustrated on Figure 1. The original picture is transformed so that its colours match the palette of the small image, regardless of the content of the pictures. Consider the two pictures as two sets of three dimensional colour pixels. A way of treating the re-colouring problem would be to find a one-to-one colour mapping that is applied for every pixel in the original image. For example in the figure, every blue pixel is re-coloured in gray. The new picture is identical in every aspects to the original picture, except that the picture now exhibits the same colour statistics, or palette, as the target picture.

The first stage of colour grading is thus to find a colour mapping that transfers the colour statistics of a picture example to the original picture. The transfer of statistics encompasses an entire range of possibilities from the simple match of the mean, variances [1], covariance [2,6] to the exact transfer of the whole probability density function (pdf) of the samples [7,3,5]. This paper considers the latter general problem of finding a one-to-one mapping function that transforms the source samples into a new sample set that exhibits the same pdf as the target samples. Finding a one-to-one mapping for one dimensional (1D) samples is a simple task that can be solved in the same way as grayscale picture equalisation [8]. The difficulty resides in extending the method to higher dimensions. This paper proposes a simple method that performs an exact colour pdf transfer, based on the iterative use of the one-dimensional pdf transfer. The advantage of the method is that the pdf transfer operates in 1D which means that the overall algorithm has a linear computational complexity of $\mathcal{O}(M)$, where M is the number of samples or pixels processed. Moreover working in 1D considerably reduces the memory load required by the algorithm.

Applying the colour mapping is enough to transfer the colour ‘feel’ to the original picture. However if the colour palette is very different, the resulting mapping can be stretched and the transformation produces some grain artefacts on parts of the picture. This can be understood if the mapping is done from a low dynamic range to a high dy-



Fig. 1. Colour transfer example. The original picture is transformed using our method so that its colours match the palette of the vignette.

namic range. The resulting mapping is stretched and thus enhances the noise level of the picture, which appears as grainy. Although the noise could be attenuated by employing colour filtering techniques [9,10], these might degrade the picture itself. Instead, we propose here a post-process which aims at protecting the original picture by preserving its original gradient field while still applying the colour transfer. This balance is done using a variational approach inspired by Poisson editing techniques [11]. Preserving the gradient of the original picture especially protects the flat areas and more generally results in the exact aspect of film grain/noise than the original image.

Organisation of the Article. After an overview of existing techniques in section 2, the mechanism of the proposed iterative pdf transfer technique is detailed in section 3. The re-graining stage is exposed in section 4 and comparative results are discussed in section 5.

2. Related Works

Re-colouring a picture using another has many applications in computer vision. In digital restoration [12] for instance, the idea is to recolour paintings that have been faded by smoke or dust. The process can also be used for colour image equalisation for scientific data visualisation [13] or simply useful for non-realistic rendering. There are thus a few articles which disclose ideas that could be used. These are discussed next.

Transfer of Colour Statistics. The first popular method proposed by Reinhard [1] matches the mean and variance of the target image to the source image. The transfer of statistics is performed separately on each channel. Since the RGB colour space is highly correlated, the transfer is done in another colour space $l\alpha\beta$ that has uncorrelated components and accounts for human-perception of colour [14]. The method has been extended to sequences in [4]. A better approach is to consider possible dependence between channels and transferring the covariance information by aligning the principal component axes [2,6]. These colour transfer techniques are limited to affine transformations of the colour space. The motion picture industry employs however routinely non-linear colour grading techniques. Hence, in a practical situation, some example-based recolouring scenarios actually require non-linear colour mapping.

A simple approach to the non-linear mapping is to perform a histogram matching [8] (or histogram specification as it is sometimes called) on the three colour channels independently [15]. A more complex mapping is considered in [13] for colour equalisation (c.f. grayscale histogram equalisation). That work proposes to deform tessellation meshes in the colour space to fit to the 3D histogram of a uniform distribution. This method can be seen as being related to warping theory which is explicitly used in [16] where the transfer of the 2D chromatic space is performed directly by using a 2D-bi-quadratic warping. An interesting approach proposed by Morovic [7] is to extend naturally the 1D case by treating the mapping via linear programming and the popular Earth-Mover distance (EMD). In that way the colour transfer is exact and minimises the amount of colour changes. The major disadvantage of using the EMD is that pixels of same colour are sometimes assigned to different colours. For instance some red pixels could be assigned to blue whereas others to green, and it is proposed in [7] to assign the colour randomly. Also the computational cost becomes intractable if a very fine clustering of the colour space is desired. Another, and computationally simpler, exact colour histogram matching technique has also been proposed by Neumann [3].

Note that, aside from these methods based on colour statistic transfer, Shen and Xin [17] have recently developed a technique that synthesises the color of 3D objects. The method is derived from a dichromatic reflection model and produces realistic recolouring under different illuminant. The methods requires however the use of image segmentation and is thus not yet well suited to generic automated tasks.

Dealing with Content Variations. One important aspect of the colour transfer problem is the change of content between the two pictures. Consider a pair of landscape images where the sky in one picture covers a larger area than in the other. When transferring the colour from one picture to the other, the excess of sky colour may be used in parts of the scenery on the ground in the other. This is because all colour transfer algorithms are sensitive to variations in the areas of the image occupied by the same colour. They risk

overstretching the colour mappings and thus producing unbelievable renderings. To deal with this issue a simple solution [1,2,17] is to manually select swatches in both pictures and thus associate colour clusters corresponding to the same content. This is tantamount to performing manual image segmentation, and is simply impractical for a large variety of images, and certainly for sequences of images.

One automated solution is to invoke the spatial information of the images to constrain the colour mapping [18,19,20]. In an extreme situation, colour from a coloured image may be required to be transferred to a grayscale image. Hence similarities between spatial neighbourhoods of the two pictures is then the only way to create a colour transfer operation automatically. This is a computationally demanding solution. Another automated solution is to restrict the variability on the colour mapping. For example in [21,22], the pixels of both images are classified in a restricted set of basic colour categories, derived from psycho-physiological studies (red, blue, pink, etc.). The colour transfer ensures for instance that blue-ish pixels remain blue-ish pixels. This gives a more natural transformation. The disadvantage is that it limits the range of possible colour transfers.

Drawing from these remarks, the method that we propose is based on an exact colour transfer technique but has the advantage of being computationally attractive. The handling of content variations is implicitly dealt with in the re-graining process, which protects the content of the original image.

3. Colour Distribution Transfer

Consider that the original and example images can be represented as two sets of M and M' colour samples $\{\mathbf{u}_i\}_{i \leq M}$ and $\{\mathbf{v}_i\}_{i \leq M'}$ respectively. The colour samples are of dimension $N = 3$ and can be represented for instance by the RGB component: $\mathbf{u}_i = (R_i, G_i, B_i)$. Other colour spaces like $L\alpha\beta$, Lab , Luv , could be considered but since the complete statistics are transferred, this has no consequences. To simplify the study, it is considered that the samples are issued from a continuous colour pdf f for the original samples, and g for the example colour samples¹. The problem is to find a differentiable mapping t that transforms the original colour pdf f into a new colour pdf that matches the target pdf g . This latter problem will be referred to as the Distribution Transfer problem.

3.1. The 1-D Case

For the 1-dimensional case, the distribution transfer problem has a very simple solution. The differentiable mapping yields the following constraint which simply corresponds to a change of variables:

$$f(u)du = g(v)dv \quad \text{with } t(u) = v \tag{1}$$

Integrating both sides of the equality yields

$$\int^u f(u)du = \int^{t(u)} g(v)dv \tag{2}$$

¹ Note that the colour pdfs can be numerically estimated using kernel density approximations [23]

Using cumulative pdf notations F and G for f and g yields the expression for the mapping t ,

$$\forall u \in \mathbb{R}, t(u) = G^{-1}(F(u)) \quad (3)$$

where $G^{-1}(\alpha) = \inf \{u | G(u) \geq \alpha\}$. The mapping can then easily be solved by using discrete look-up tables.

3.2. The N -D Case

Extending the 1-dimensional case to higher dimensions is sadly not trivial. As in [7] and [3], the method considers an exact pdf colour matching. The idea proposed here still preserve some monotonous and smoothness properties but avoids the direct manipulation of the voluminous 3D colour pdf as in [7,3]. The idea is to break down the problem into a succession of 1-Dimensional distribution transfer problems. Consider the use of the N -dimensional Radon Transform. It is widely acknowledged that via the Radon Transform, any N -dimensional function can be uniquely described as a series of projections onto 1-dimensional axes [24]. In this case, the function considered is a N -dimensional pdf, hence the Radon Transform projections result in a series of 1-dimensional marginal pdfs. Intuitively then, operations on the N -dimensional pdf should be possible through manipulations of the 1-dimensional marginals. Consider that after some sequence of such manipulations, all 1-dimensional marginals match the corresponding marginals of the target distribution. It then follows that, by nature of the Radon Transform, the transformed f , corresponding to the transformed 1-dimensional marginals, now matches g .

There are now several questions to answer. How to manipulate the 1-dimensional pdfs? What guarantee is there of eventual convergence? How many axes are needed? Is there an optimal sequence of axes? The following paragraphs provide the answers to these questions.

3.3. Description of the Manipulation

The operation applied to the projected marginal distributions is thus similar to that used in 1-dimension. Consider a particular axis denoted by its vector direction $\mathbf{e} \in \mathbb{R}^N$. The projection of both pdfs f and g onto the axis \mathbf{e} results in two 1-dimensional marginal pdfs $f_{\mathbf{e}}$ and $g_{\mathbf{e}}$. Using the 1-dimensional pdf transfer mapping of the equation (3) yields a 1-dimensional mapping $t_{\mathbf{e}}$ along this axis:

$$\forall u \in \mathbb{R}, t_{\mathbf{e}}(u) = G_{\mathbf{e}}^{-1}(F_{\mathbf{e}}(u)) \quad (4)$$

For a N -dimensional sample $\mathbf{u} = [u_1, \dots, u_N]^T$, the projection of the sample on the axis is given by the scalar product $\mathbf{e}^T \mathbf{u} = \sum_i e_i u_i$, and the corresponding displacement along the axis is

$$\mathbf{u} \rightarrow \mathbf{u} + (t_{\mathbf{e}}(\mathbf{e}^T \mathbf{u}) - \mathbf{e}^T \mathbf{u}) \mathbf{e} \quad (5)$$

After transformation, the projection $f'_{\mathbf{e}}$ of the new distribution f' is now identical to $g_{\mathbf{e}}$. The manipulation is explained in figure 2.

Considering that the operation can be done independently on orthogonal axes, the proposed manipulation consists in choosing an orthogonal basis $\mathbf{R} = (\mathbf{e}_1, \dots, \mathbf{e}_N)$ and then applying the following mapping τ :

Algorithm 1 IDT method

- 1: **Initialisation** of the data set source \mathbf{u}
 $k \leftarrow 0$, $\mathbf{u}^{(0)} \leftarrow \mathbf{u}$
- 2: **repeat**
- 3: take a rotation matrix $R = [\mathbf{e}_1, \dots, \mathbf{e}_N]$
- 4: for every rotated axis i of the rotation, get the projections f_i and g_i
- 5: find the 1D transformation t_i that matches the marginals f_i into g_i
- 6: remap the samples \mathbf{u} according to the 1D transformations:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{R} \begin{pmatrix} t_1(\mathbf{e}_1^T \mathbf{u}^{(k)}) - \mathbf{e}_1^T \mathbf{u}^{(k)} \\ \vdots \\ t_N(\mathbf{e}_N^T \mathbf{u}^{(k)}) - \mathbf{e}_N^T \mathbf{u}^{(k)} \end{pmatrix}$$

- 7: $k \leftarrow k + 1$
 - 8: **until** convergence on all marginals for every possible rotation
 - 9: The final one-to-one mapping T is given by: $\forall j, \mathbf{u}_j \mapsto t(\mathbf{u}_j) = \mathbf{u}_j^{(\infty)}$
-

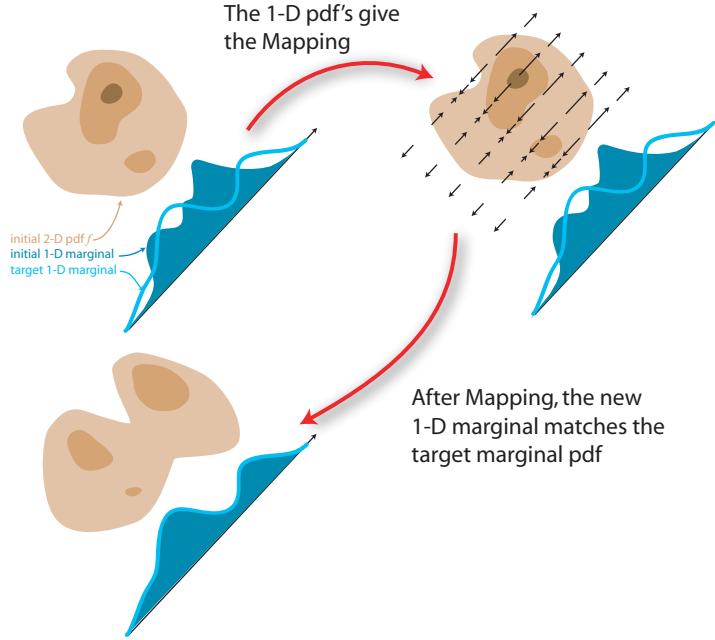


Fig. 2. Illustration of the data manipulation, based on the 1-dimensional pdf transfer on one axis.

$$\tau(\mathbf{u}) = \mathbf{u} + \mathbf{R} \begin{pmatrix} t_1(\mathbf{e}_1^T \mathbf{u}) - \mathbf{e}_1^T \mathbf{u} \\ \vdots \\ t_N(\mathbf{e}_N^T \mathbf{u}) - \mathbf{e}_N^T \mathbf{u} \end{pmatrix} \quad (6)$$

where t_i is the 1-dimensional pdf transfer mapping for the axis \mathbf{e}_i .

The idea is that iterating this manipulation over different axes will result in a sequence of distributions $f^{(k)}$ that hopefully converges to the target distribution g . The overall algorithm will be referred to as the **Iterative Distribution Transfer (IDT)** algorithm.

A theoretical proof that the algorithm converges when the target distribution is the standard normal distribution is given in the appendix A. The experimental study in the following section suggests that convergence occurs for any distribution. The study also considers the problem of finding sequence of axes that maximises the convergence speed of the algorithm.

3.4. Results and Choice of Axis Sequence

To explore the convergence, the Kullback-Leibler (KL) *divergence* is used as a measure to quantify how well the transformed distribution $f^{(k)}$ matches the target pdf g . The Kullback-Leibler is defined as follows:

$$D_{KL}(f\|g) = \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f(\mathbf{u})}{g(\mathbf{u})} \right) d\mathbf{u} \quad (7)$$

One simple experiment that could be used to assess the impact of axis sequences on convergence is simply to choose two particular datasets, use one as a target pdf and the

other as a source pdf. Then for various axis sequences, the KL measure could be used to assess convergence as each iteration of the algorithm proceeds. Since the KL divergence is estimated here on sample datasets, the kernel density modelling [23] is used to estimate the underlying pdfs as follows:

$$D(f \parallel g) = \frac{1}{M} \sum_{i=1}^M \ln \left[\frac{\sum_j K\left(\frac{\|\mathbf{u}_i - \mathbf{v}_j\|}{h_i}\right)}{\sum_j K\left(\frac{\|\mathbf{u}_i - \mathbf{u}_j\|}{h_i}\right)} \right] \quad (8)$$

where K is the Epanechnikov kernel. The Epanechnikov kernel is the function $K(u) = (3/4)(1 - u^2)$ for $-1 < u < 1$ and zero for u outside that range. To account for the sparseness of samples, it is crucial to use a variable bandwidth approach. Typically, for a sample \mathbf{u}_i , the bandwidth h_i increases with the sparseness of the samples around \mathbf{u}_i . A clear outline of the bandwidth selection is available in [25], and that is used here. A major aspect of the experiment is that the KL divergence has to be non-negative. This is indeed observed in numerical results (see figure A.1), but not for any choice of bandwidth values. To counterbalance this sensitivity of the measure to the quality of the pdf estimation, the pdf is over-smoothed by taking large values of kernel bandwidths. The resulting KL divergence measure is under-evaluated since both estimated pdfs are more uniform. Another consequence is that the convergence speed reported on the figure A.2 is actually slower than the true one.

It is now possible to study the convergence of the distribution transfer method by measuring the mean KL divergence over the iterations. In the following experiments, the plots are obtained by averaging the KL measure for 100 different pdfs. Figure A.2 shows the evolution of the KL divergence for a random selection of axis set sequences, with both random initial and target distributions. It transpires from the results that the algorithm converges for any continuous distribution.

Taking a random selection of orthogonal bases seems to be sufficient to obtain convergence, however it is shown hereafter, that a better choice of axis sequence can still substantially improve the convergence speed. This can be understood by considering that random axes are probably correlated. Figure A.3 confirms this intuition. The figure displays the average KL divergence after two iterations of the algorithm for 2D pdfs. At iteration 1, a fixed set of axes is chosen, thus the plot shows the evolution of the KL divergence depending on the choice of axes at iteration 2. The graph clearly shows that the KL improvement depends on the correlation between axes.

Intuitively then, an interesting heuristic would be to consider a sequence of rotations that maximises the distances between the current axis set at iteration k and the previous axis sets. Define the distance between two axes by

$$d(\mathbf{e}_1, \mathbf{e}_2) = \min(\|\mathbf{e}_1 - \mathbf{e}_2\|_2, \|\mathbf{e}_1 + \mathbf{e}_2\|_2) \quad (9)$$

with \mathbf{e}_1 and \mathbf{e}_2 the supporting axis vectors. To find axes that are far apart, one solution is to maximise the distances $d(\mathbf{e}_1, \mathbf{e}_2)$. This turns out to be a numerically unstable solution. A better formulation is to consider instead the minimisation of the potential $1/(1+d(\mathbf{e}_1, \mathbf{e}_2))$ and then to insure that distances are far apart:

$$\begin{aligned} [\mathbf{e}_{k+1}^1, \dots, \mathbf{e}_{k+1}^N] = \\ \arg \min_{[\mathbf{e}^1, \dots, \mathbf{e}^N]} \left\{ \sum_{l=1}^k \sum_{i=1}^N \sum_{j=1}^N \frac{1}{1 + d(\mathbf{e}_l^j, \mathbf{e}_i^i)} \right\} \end{aligned} \quad (10)$$

with the constraint that the bases are orthogonal. This minimisation problem can be numerically simulated under MATLAB using the downhill Simplex algorithm [26]. The constraint of normalisation $\|e\| = 1$ can be avoided by taking hyperspherical coordinates [27]. The orthogonality of the base can be obtained using Gram-Schmidt orthogonalisation. The resulting first bases for dimension 2 and 3 are given in appendix tables A.1 and A.2. Note that since the algorithm is iterative, it is not crucial to require high accuracy in the estimation of the bases. The code used to find the bases is available on [28].

The improvements on the convergence speed are shown on figures A.4 and A.5. One can expect on average a speed improvement of 1.5 in 3D to 2.15 in 2D. This is consistent with the fact that axes are more likely to be decorrelated in higher dimensions. The sequence which is sought, is optimal on average over the couple of pdfs. Note that other strategies could be explored, like finding the optimal sequence of axis for a particular couple of distributions.

4. Reducing Grain Noise Artefacts

Applying the colour mapping as it is on the picture might however produce some grain artefacts as shown in figure 3 and 4. When the content differs, or when the dynamic ranges of both pictures are too different, the resulting mapping function can be stretched on some parts (see figure 3-e), and thus enhances the noise level (see figure 3-c). This can be understood by taking the simple example of a linear transformation t of the original picture I : $t(I) = a I + b$. The overall variance of the resulting picture is changed to $\text{var}\{t(I)\} = a^2 \text{var}\{I\}$. This means that a greater stretching ($a > 1$) produces a greater noise.

The solution proposed here to reduce the grain is to run a post-processing algorithm that forces the noise level to remain the same. The idea is to adjust the gradient field of the resulting picture so that it matches the gradient field of the original picture. If the gradient fields of both pictures are similar, the noise level will be the same. Matching the gradient of a picture has been addressed in different computer vision applications like high dynamic range compression [29]; the value of this idea has been thoroughly demonstrated by Pérez et al. in [11]. Manipulating the image gradient can be efficiently done by using a variational approach. The problem here is slightly different, since re-colouring also implies changing the contrast levels. Thus the new gradient field should only loosely match the original gradient field.

Denote $I(x, y)$ the 3-dimensional original colour picture. To simplify the discussion, coordinates are omitted in the expressions and I, J, ψ, ϕ , etc. actually refer to $I(x, y)$, $J(x, y)$, $\psi(x, y)$ and $\phi(x, y)$. Let $t : I \rightarrow t(I)$ be the colour transformation. The problem is to find a modified image J of the mapped picture $t(I)$ that minimises on the whole picture range Ω

$$\min_J \iint_{\Omega} \phi \cdot \|\nabla J - \nabla I\|^2 + \psi \cdot \|J - t(I)\|^2 dx dy \quad (11)$$

with Neumann boundaries condition $\nabla J|_{\partial\Omega} = \nabla I|_{\partial\Omega}$ so that the gradient of J matches with the gradient of I at the picture border $\partial\Omega$. The term $\|\nabla J - \nabla I\|^2$ forces the image gradient to be preserved. The term $\|J - t(I)\|^2$ ensures that the colours remain close to the target picture and thus protects the contrast changes. Without $\|J - t(I)\|^2$, a solution of equation (11) will be actually the original picture I .

The weight fields $\phi(x, y)$ and $\psi(x, y)$ affect the importance of both terms. Many choices are possible for ϕ and ψ , and the following study could be easily be changed, depending on the specifications of the problem.

The weight field $\phi(x, y)$ has been here chosen to emphasise that only flat areas have to remain flat but that gradient can change at object borders:

$$\phi = \frac{30}{1 + 10 \|\nabla I\|} \quad (12)$$

The weight field $\psi(x, y)$ accounts for the possible stretching of the transformation t . Where ∇t is big, the grain becomes more visible:

$$\psi = \begin{cases} 2 / (1 + \|(\nabla t)(I)\|) & \text{if } \|\nabla I\| > 5 \\ \|\nabla I\|/5 & \text{if } \|\nabla I\| \leq 5 \end{cases} \quad (13)$$

where $(\nabla t)(I)$ is the gradient of t for the colour I and thus refers to the colour stretching. The case $\|\nabla I\| \leq 5$ is necessary to re-enforce that flat areas remains flat. While the gradient of t is easy to estimate for grayscale pictures, it might be more difficult to obtain for colour mappings. The field can then be changed into:

$$\psi(x, y) = \begin{cases} 1 & \text{if } \|\nabla I\| > 5 \\ \|\nabla I\|/5 & \text{if } \|\nabla I\| \leq 5 \end{cases} \quad (14)$$

Numerical Solution. The minimisation problem in equation (11) can be solved using the variational principle which states that the integral must satisfy the Euler-Lagrange equation:

$$\frac{\partial F}{\partial J} - \frac{d}{dx} \frac{\partial F}{\partial J_x} - \frac{d}{dy} \frac{\partial F}{\partial J_y} = 0 \quad (15)$$

where

$$F(J, \nabla J) = \phi \cdot \|\nabla J - \nabla I\|^2 + \psi \cdot \|J - t(I)\|^2 \quad (16)$$

from which the following can be derived:

$$\phi \cdot J - \operatorname{div}(\psi \cdot \nabla J) = \phi \cdot t(I) - \operatorname{div}(\psi \cdot \nabla I) \quad (17)$$

This is an elliptic partial differential equation. The expression $\operatorname{div}(\psi \cdot \nabla I)$ at pixel $\mathbf{x} = (x, y)$ can be approximated using standard finite differences [30] by:

$$\operatorname{div}(\psi \cdot \nabla I)(\mathbf{x}) \approx \sum_{\mathbf{x}_n \in \mathcal{N}_{\mathbf{x}}} \frac{\psi_{\mathbf{x}_n} + \psi_{\mathbf{x}}}{2} (I_{\mathbf{x}_n} - I_{\mathbf{x}}) \quad (18)$$

where $\mathcal{N}_{\mathbf{x}}$ corresponds to the four neighbouring pixels of \mathbf{x} . Using this in equation (17) yields a linear system as follows:

$$\begin{aligned} a_1(x, y)J(x, y - 1) + a_2(x, y)J(x, y + 1) + \\ a_3(x, y)J(x - 1, y) + a_4(x, y)J(x + 1, y) + \\ a_5(x, y)J(x, y) = a_6(x, y) \end{aligned} \quad (19)$$

with

$$\begin{aligned}
a_1(x, y) &= -\frac{\psi(x, y - 1) + \psi(x, y)}{2} \\
a_2(x, y) &= -\frac{\psi(x, y + 1) + \psi(x, y)}{2} \\
a_3(x, y) &= -\frac{\psi(x - 1, y) + \psi(x, y)}{2} \\
a_4(x, y) &= -\frac{\psi(x + 1, y) + \psi(x, y)}{2} \\
a_5(x, y) &= \\
&\quad \frac{1}{2} \left(4\psi(x, y) + \psi(x, y - 1) + \psi(x, y + 1) \right. \\
&\quad \left. + \psi(x - 1, y) + \psi(x + 1, y) \right) + \phi(x, y) \\
a_6(x, y) &= \\
&\quad \frac{1}{2} \left((\psi(x, y) + \psi(x, y - 1))(I(x, y - 1) - I(x, y)) \right. \\
&\quad + (\psi(x, y) + \psi(x, y + 1))(I(x, y + 1) - I(x, y)) \\
&\quad + (\psi(x, y) + \psi(x - 1, y))(I(x - 1, y) - I(x, y)) \\
&\quad \left. + (\psi(x, y) + \psi(x + 1, y))(I(x + 1, y) - I(x, y)) \right) \\
&\quad + \phi(x, y)I(x, y)
\end{aligned}$$

The system can be solved by standard iterative methods like SOR, Gauss-Seidel with multigrid approach. Implementations of these numerical solvers are widely available and one can refer for instance to the Numerical Recipes [26]. The main step of these methods is to solve iteratively for $J(x, y)$. Note that $J(x, y)$ and $a_i(x, y)$ are of dimension 3, but that each colour component can be treated independently. For instance, the iteration for the red component field is of the form

$$\begin{aligned}
J_R^{(k+1)}(x, y) &= \frac{1}{a_5^R(x, y)} \left(a_5^R(x, y) \right. \\
&\quad - a_1^R(x, y)J_R^{(k)}(x, y - 1) - a_2^R(x, y)J_R^{(k)}(x, y + 1) \\
&\quad \left. - a_3^R(x, y)J_R^{(k)}(x - 1, y) - a_4^R(x, y)J_R^{(k)}(x + 1, y) \right) \tag{20}
\end{aligned}$$

where $J_R^{(k)}(x, y)$ is the result in the red component at the k^{th} iteration.

5. Results

The proposed colour grading technique has been tested on a number of different scenarios. Examples of colour grading for matching lighting conditions are presented in the figure 5. On the first row, the colour properties of the sunset are used to synthesise the ‘evening’ scene depicted at sunset. On the second row, the colour grading allows correction of the change of lighting conditions induced by clouds. Even when using the grain artefact reducer, an unavoidable limitation of colour grading is the clipping of the colour data: saturated areas cannot be retrieved (for instance the sky on the golf image cannot be recovered). A general rule is to match pictures from higher to lower range dynamics.

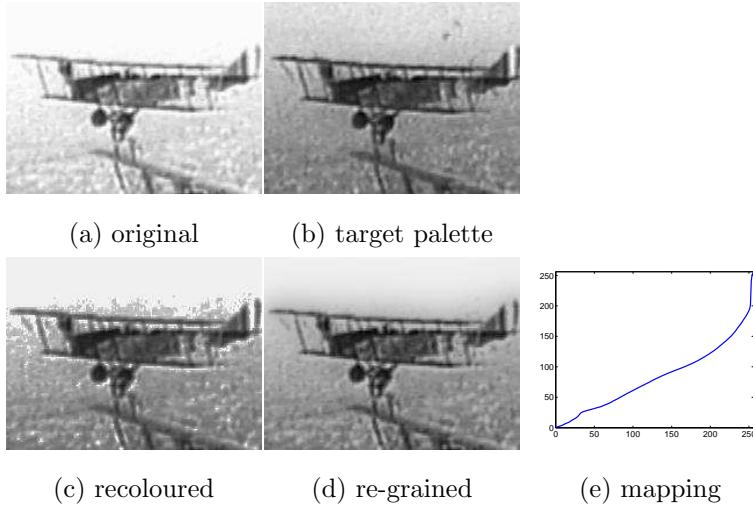


Fig. 3. Result of grain reducing. The two consecutive archive frames (a) and (b) suffer from extreme brightness variation (this artefact is known as flicker [31]). The corresponding mapping transformation (e) is overstretched, which results in an increased level of noise on the mapped original frame (c). The proposed grain artefact reducer is able to reproduce the noise level of the original picture. The top of the original picture is saturated and cannot be retrieved but the algorithm succeeds in preserving the soft gradient.

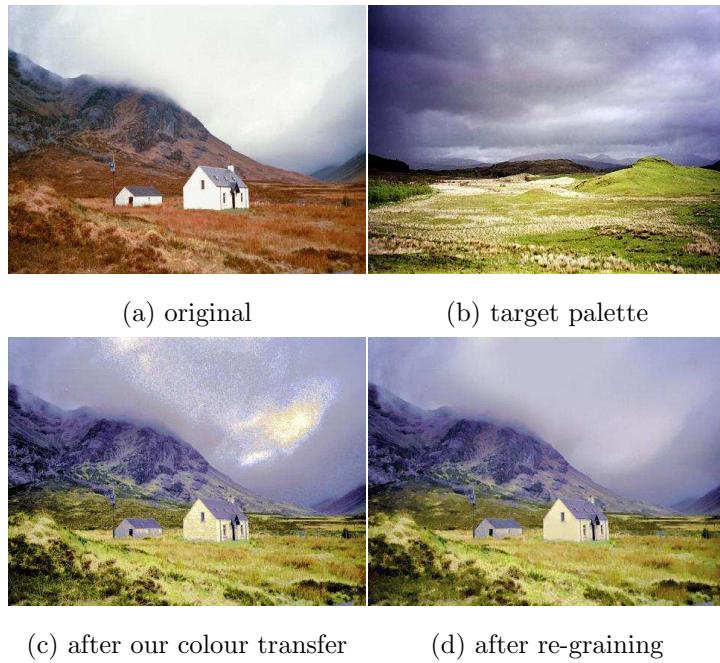


Fig. 4. Artefact grain reducing for colour picture. See how the details of the picture are preserved, while the spurious graininess in the sky is washed out.

The figure 6 displays examples of colour restoration of faded movies. The idea is similar to colour grading as the idea is to recreate different atmospheres. The target pictures used for recreating the atmosphere are on the second row.

Comparisons with other methods. The comparison has been done using Figure 7 as the original and Figure 8 as the target palette. Neumann's results are displayed on Figure 9 and ours on Figures 10 and 11. Our method seems to give a slightly better colour mapping (the tree is brown and the houses remain red). However these methods are both based on exact colour pdf transfers, thus these methods will produce results of similar nature. The major difference in the results occurs after the re-graining process. As seen on Figure 12, the re-graining process helps a lot to reduce the graininess resulting from the colour mapping.

Results on Figure 13 using a PCA-based method similar to [2] and [6] are also visually pleasing. This is usually the case with PCA methods because the mapping is linear and no obvious artefacts are generated. PCA methods however cannot reduce the colour palette and non authorised colours are still present. Results using Reinhard's technique are displayed on Figure 14. To be successful, Reinhard's technique requires that the user places swatches to decide of colour correspondences.

The overall method takes around a second per image at PAL resolution on a 2GHz machine using a commercial implementation of the method.

6. Conclusion

This paper has proposed an original technique for colour grading. The technique is based on an exact transfer of colour pdf of the target picture. The possible grain artefacts are removed in a second step. The overall technique is simple, easy to implement and works for a large variety of scenarios, even when the example picture is very different from the processed images.

The idea of re-using the statistics of real images to grade images is a simple but powerful method for rendering images. The method is part of the larger domain of non-parametric image processing, which has been proved to be a successful approach in the recent years. Future works will explore new applications using statistics of other features (shape, textures, etc.).

Appendix A. Convergence Study

This appendix investigates the theoretical convergence properties of this algorithm. The following theorem establishes the convergence of the algorithm when the target pdf is the standard normal distribution (denoted by $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$). Thus by combining the mappings at each iteration, it is possible to find for any distribution f a differentiable bijective mapping $t_f = \tau_k \circ \dots \circ \tau_1$ that transforms f into the standard distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Consider then the two differentiable mappings t_f and t_g that transform f and g into $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Using the standard distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$ as a pivot results in the transformation $t_g^{-1} \circ t_f$, which is a differentiable bijective mapping that transforms f into g , and is therefore a solution of the distribution transfer problem:

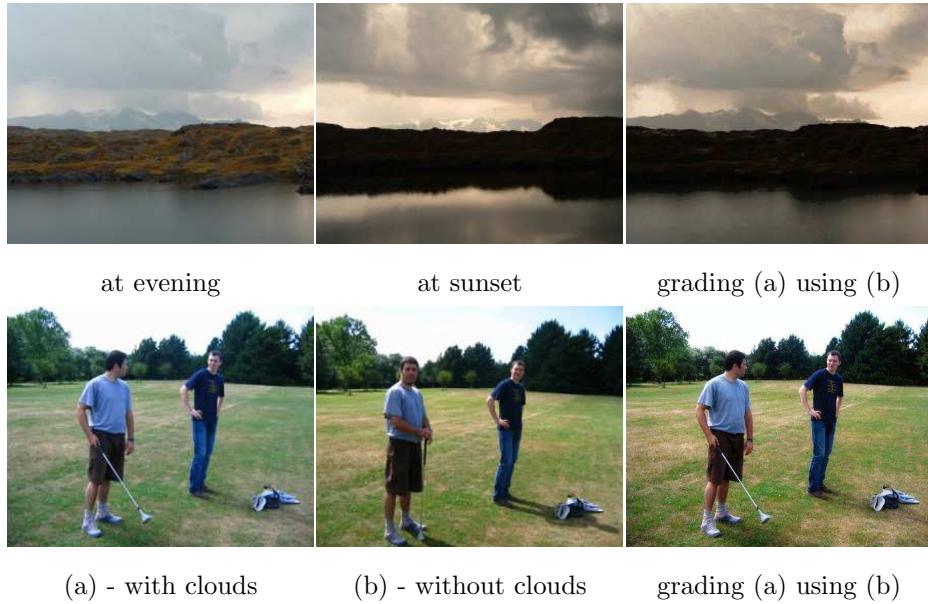


Fig. 5. Examples of colour grading for matching lighting conditions. On the first row, the colour properties of the sunset are used to synthesise the ‘evening’ scene depicted at sunset. On the second row, the colour grading allows to correct the change of lighting conditions induced by clouds.

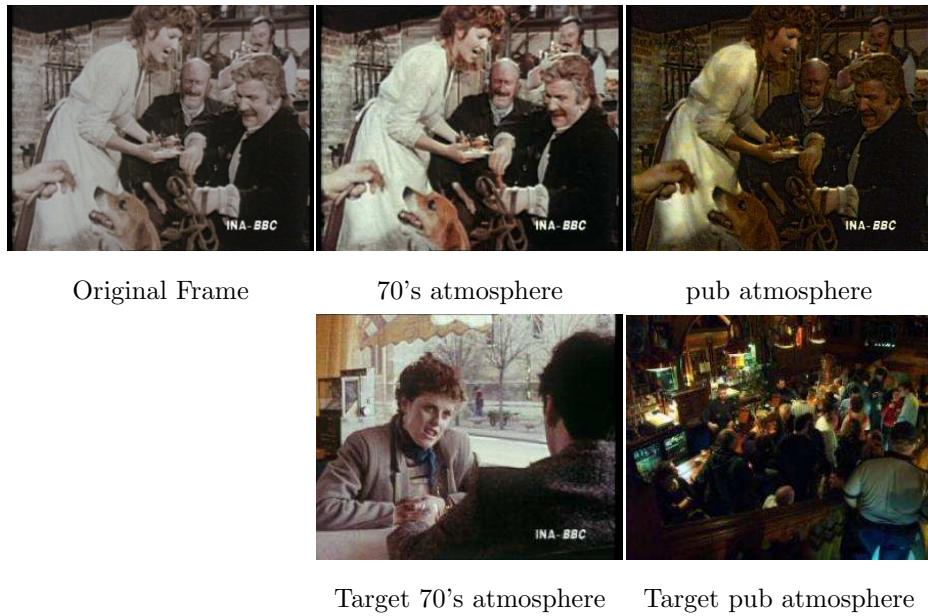


Fig. 6. Example of colour grading for image and video restoration. It is possible to recreate different atmospheres. Here an old faded film is transformed to match the colour scheme of a movie from the 70’s and a pub ambiance.



Fig. 7. original (Kitaj, R.B, The Oak Tree, 1991)

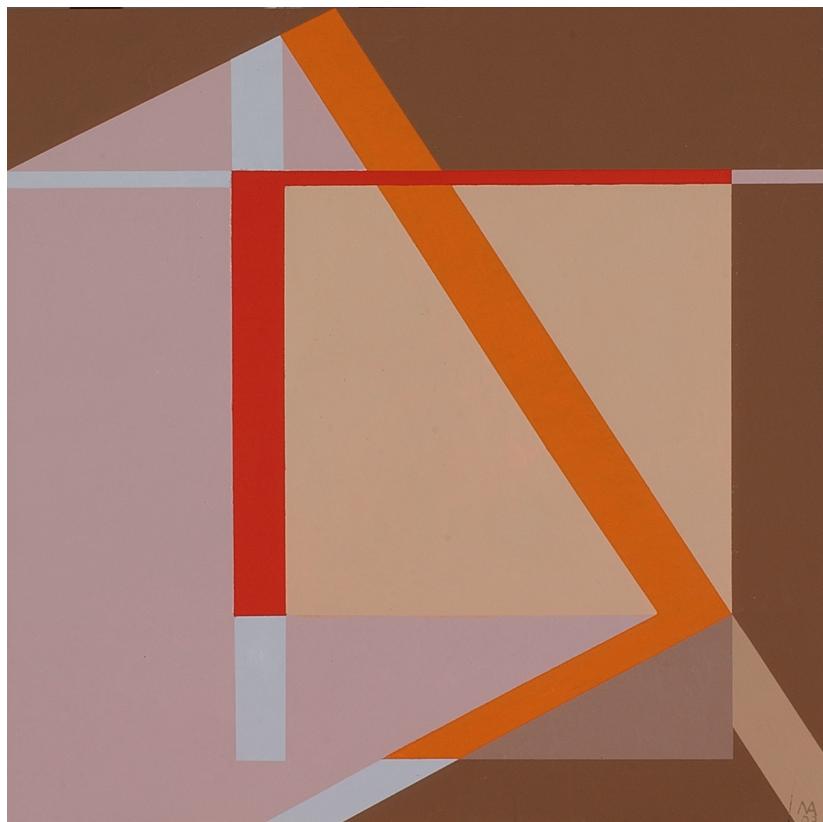


Fig. 8. Target palette (Nemcsics, A. ,Europe, 2003)

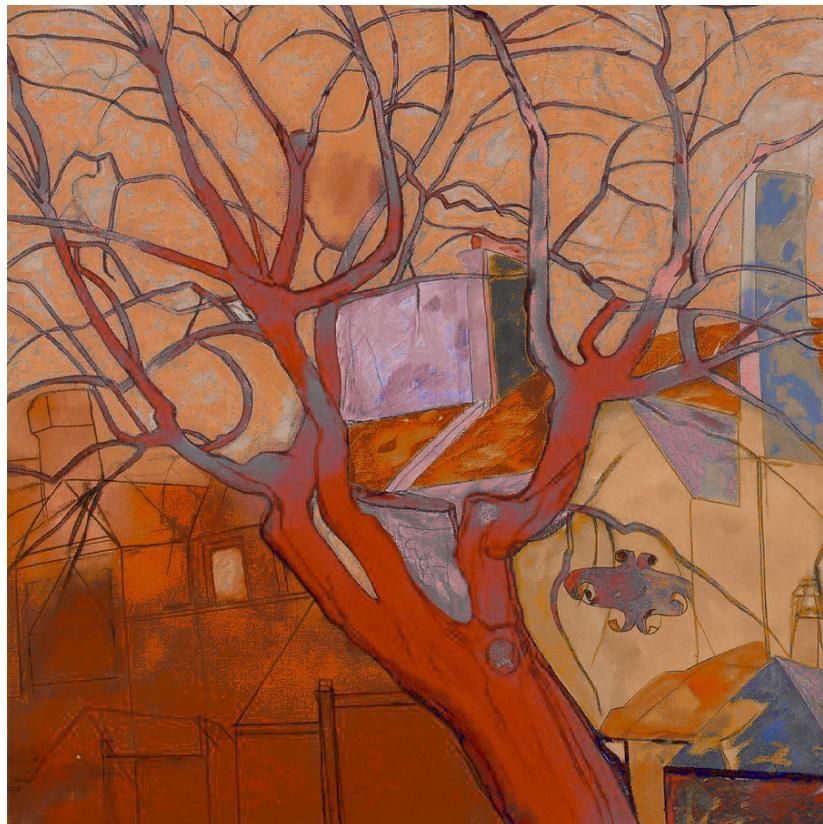


Fig. 9. Results from Neumann [3]



Fig. 10. Our iterative colour transfer results.

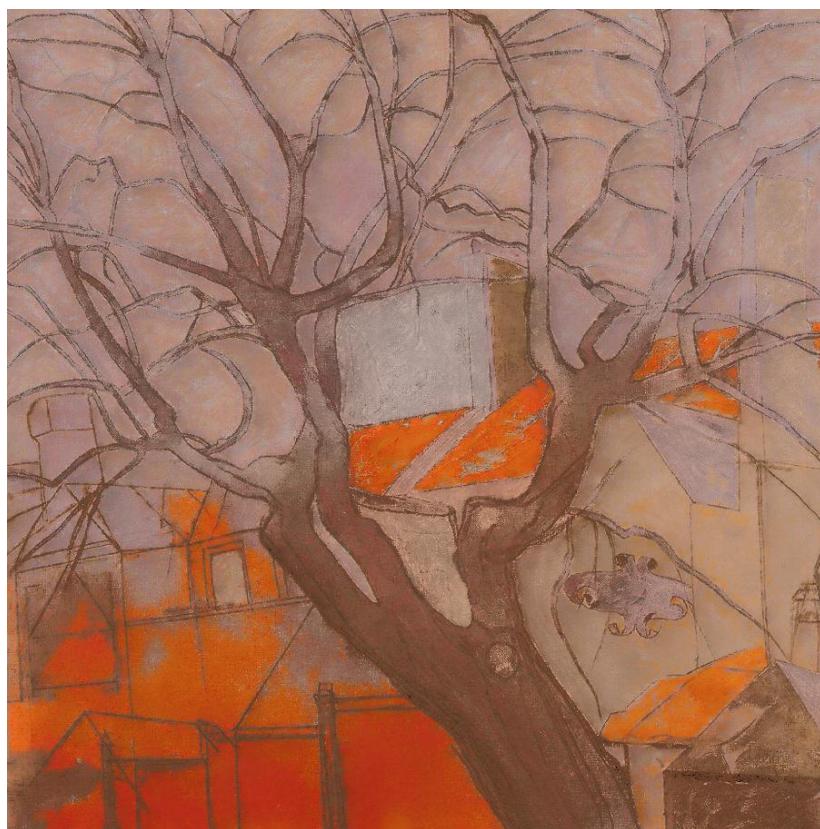


Fig. 11. Our results after re-graining.

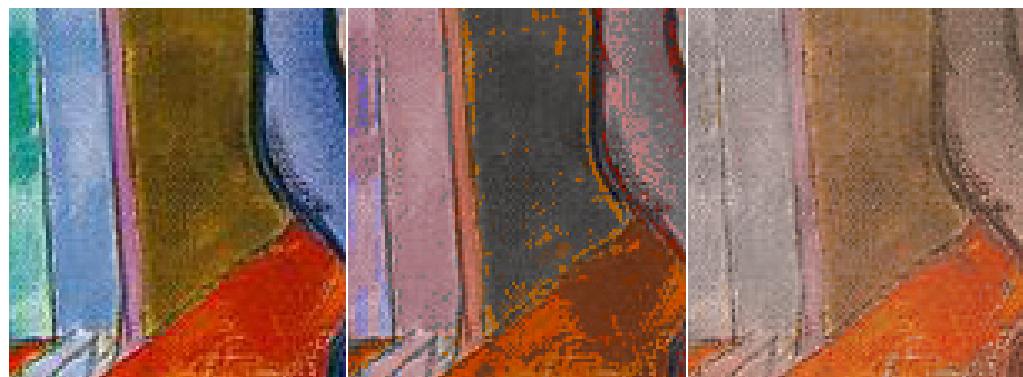


Fig. 12. Detail of the colour transfer between Figures 7 and 8. On the left the original, in the middle results from Neumann and on the right our results.



Fig. 13. Results using a PCA based technique, similar to [6,2]

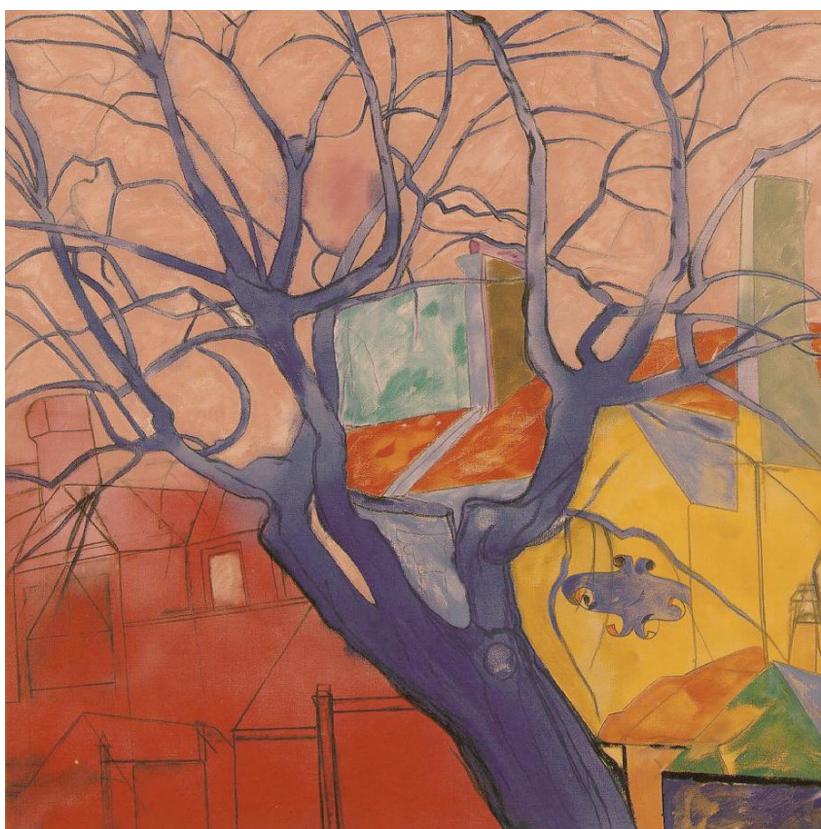


Fig. 14. Results using Reinhard's method without any interaction. To be successful the method would require a user interaction.

$$\begin{aligned}
t_f : \quad f &\rightarrow \mathcal{N}(\mathbf{0}, \mathbf{id}_N) \\
t_g : \quad g &\rightarrow \mathcal{N}(\mathbf{0}, \mathbf{id}_N) \\
\forall \mathbf{u} \in \mathbb{R}^N, \quad t(\mathbf{u}) &= t_g^{-1}(t_f(\mathbf{u}))
\end{aligned} \tag{A.1}$$

The results of the convergence study presented in the paper indicate however that the algorithm converges to the solution even if the target distribution is not the normal distribution. Figure A.1 shows an example illustrating the convergence of the process for 2-dimensional pdfs. Thus the method can also be used directly to find a mapping between f and g , without having to resort to the standard distribution as a pivot.

Theorem 1 (Isotropic Case) *Let f be an N -dimensional pdf and denote by g the pdf of the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Consider the differentiable transformation τ_k that matches as in equation (6) the marginals of f to the projections of g for a random set of axes. Then the sequence defined by $f^{(0)} = f$ and $f^{(k+1)} = \tau_k(f^{(k)})$ converges to the target distribution: $f^{(\infty)} = g$.*

Proof. Denote the original pdf f and the target standard normal pdf g . For a particular set of axes, denote f_1, \dots, f_N the marginals of f and g_1, \dots, g_N the marginals of g . The standard distribution is isotropic and for all axes, it can be written as the product of its marginals: $g = g_1 \cdots g_N$. The key of the proof is to show that the Kullback-Leibler divergence decreases for any set of axes.

The Kullback-Leibler divergence, or relative entropy, is a quantity which measures the difference between two probability distributions. It is computed as follows:

$$D_{KL}(f\|g) = \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f(\mathbf{u})}{g(\mathbf{u})} \right) d\mathbf{u} \tag{A.2}$$

As with many measures over distributions, the KL divergence is not a proper distance. The KL divergence does not satisfy the triangular inequality and is not symmetric. However the KL divergence is always non-negative and $D_{KL}(p, q)$ is zero iff $p = q$.

$$\begin{aligned}
D_{KL}(f\|g) &= \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f(\mathbf{u})}{g(\mathbf{u})} \right) d\mathbf{u} \\
&= \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f(\mathbf{u})}{f_1(u_1) \cdots f_N(u_N)} \right) d\mathbf{u} \\
&\quad + \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f_1(u_1) \cdots f_N(u_N)}{g(\mathbf{u})} \right) d\mathbf{u} \\
&= D_{KL}(f\|f_1 \cdots f_N) \\
&\quad + \int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f_1(u_1) \cdots f_N(u_N)}{g_1(u_1) \cdots g_N(u_N)} \right) d\mathbf{u}
\end{aligned}$$

Then by marginalising,

$$\begin{aligned}
&\int_{\mathbf{u}} f(\mathbf{u}) \ln \left(\frac{f_i(u_i)}{g_i(u_i)} \right) d\mathbf{u} \\
&= \int_{u_1} \cdots \int_{u_N} f(u_1, \dots, u_N) \ln \left(\frac{f_i(u_i)}{g_i(u_i)} \right) du_1 \cdots du_N \\
&= \int_{u_i} f_i(u_i) \ln \left(\frac{f_i(u_i)}{g_i(u_i)} \right) du_i \\
&= D_{KL}(f_i\|g_i)
\end{aligned} \tag{A.3}$$

Table A.1
Optimised rotations for $N = 2$

No.	1	2	3	4	5	6
x	1.000000 0.000000	0.707107 0.707107	0.923880 -0.382683	0.923880 0.382683	0.555553 0.831481	0.980781 0.195110
y	0.000000 1.000000	0.707107 -0.707107	-0.382683 -0.923880	-0.382683 0.923880	0.831481 -0.555553	-0.195110 0.980781
No.	7	8	9	10	11	12
x	0.555587 -0.831458	-0.195071 -0.980789	0.989456 0.144836	0.597236 0.802066	0.969564 -0.244837	0.858711 0.512460
y	-0.831458 -0.555587	0.980789 -0.195071	-0.144836 0.989456	0.802066 -0.597236	-0.244837 -0.969564	-0.512460 0.858711

it follows eventually that

$$D_{KL}(f\|g) = D_{KL}(f\|f_1 \cdots f_N) + \sum_{i=1}^N D_{KL}(f_i\|g_i) \quad (\text{A.4})$$

Consider now that the mapping transforms f into f' and $f_1 \cdots f_N$ into $f'_1 \cdots f'_N$. It holds for f' that:

$$D_{KL}(f'\|g) = D_{KL}(f'\|f'_1 \cdots f'_N) + \sum_{i=1}^N D_{KL}(f'_i\|g_i) \quad (\text{A.5})$$

The transformation is such as for each axis i , $f'_i = g_i$, thus $\sum_{i=1}^N D_{KL}(f'_i\|g_i) = 0$. Also the KL divergence is left invariant by bijective transformation, which implies that $D_{KL}(f'\|f'_1 \cdots f'_N) = D_{KL}(f\|f_1 \cdots f_N)$. Thus the KL difference decreases at each iteration by:

$$D_{KL}(f\|g) - D_{KL}(f'\|g) = \sum_{i=1}^N D_{KL}(f_i\|g_i) \geq 0 \quad (\text{A.6})$$

Since $D_{KL}(f\|g)$ is decreasing and non-negative, it must have a limit. Consider that the orthogonal basis at each iteration is picked up randomly from a finite set of orthogonal bases \mathcal{O} . Then for each orthogonal matrix of \mathcal{O} , the corresponding sub-sequence $D_{KL}(f^{(\sigma(k))}\|g)$ is also decreasing and convergent and $\sum_{i=1}^N D_{KL}(f_i^{(\sigma(k))}\|g_i) \rightarrow 0$. This means that for this basis the marginals converge to the corresponding marginals of g . Since this is true for any choice of \mathcal{O} , all marginals of f converge to the marginals of g . The important point here is that by picking up bases randomly, each possible basis becomes an adherent point of the basis sequence. The Radon transform of $f^{(k)}$ converges to the Radon transform of g and since the Radon transform admits a unique inverse, this shows also that the pdf $f^{(k)}$ converges to g . \square

References

- [1] E. Reinhard, M. Ashikhmin, B. Gooch, P. Shirley, Color transfer between images, IEEE Computer Graphics Applications 21 (5) (2001) 34–41.
- [2] A. Adabpour, S. Kasaei, A fast and efficient fuzzy color transfer method, in: Proceedings of the IEEE Symposium on Signal Processing and Information Technology, 2004, pp. 491–494.

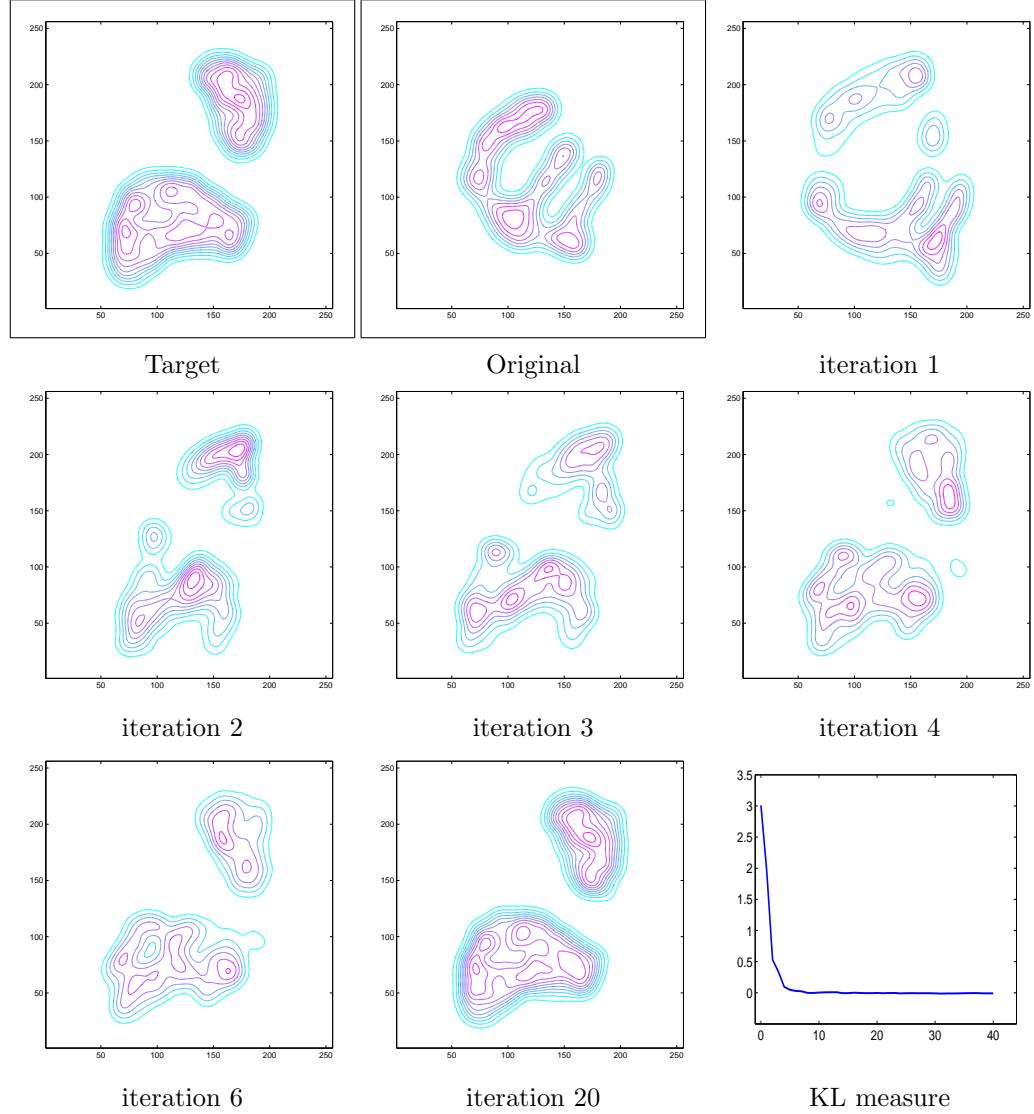


Fig. A.1. Example of a two dimensional pdf transfer using **IDT**. The decrease of the Kullback-Leibler divergence illustrates the convergence of method.

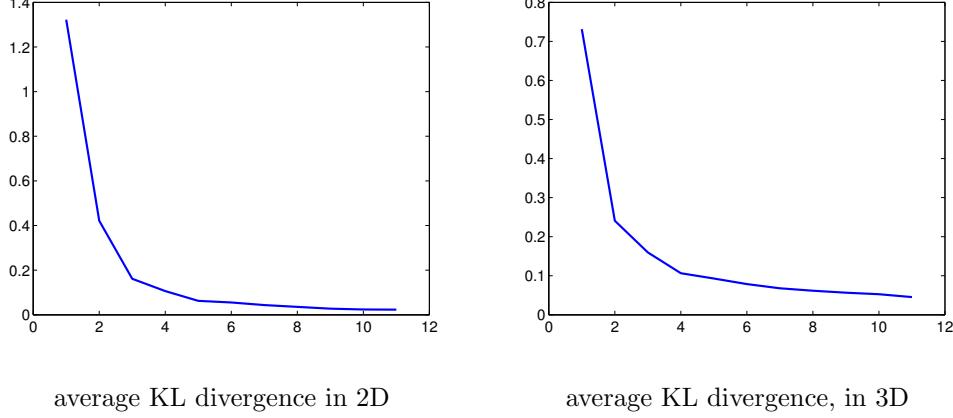


Fig. A.2. Averaged evolution of the Kullback Leibler divergence for 100 simulations with both random initial distribution and target distribution. Rotations are taken randomly. It transpires from the results that convergence occurs for any distribution.

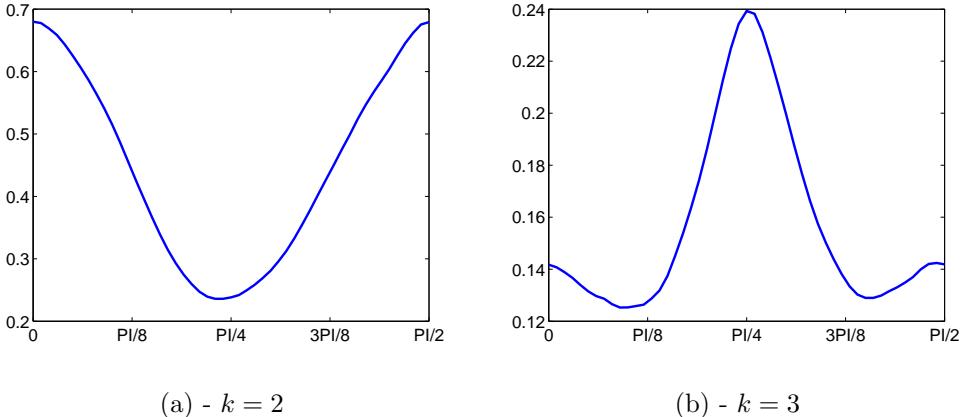


Fig. A.3. Average KL divergence to target distribution in function of the rotation sequence. The distributions are 2D and the rotation matrix can be represented by a single angle α between 0 and $\pi/2$. On the left, after one iteration at $\alpha = 0$. The best rotation is for $\alpha = \pi/4$, which corresponds to the less correlated axes. On the right, after $\alpha = 0, \pi/4$. The best new rotation occurs around $\alpha = 3\pi/8$ and $\pi/8$.

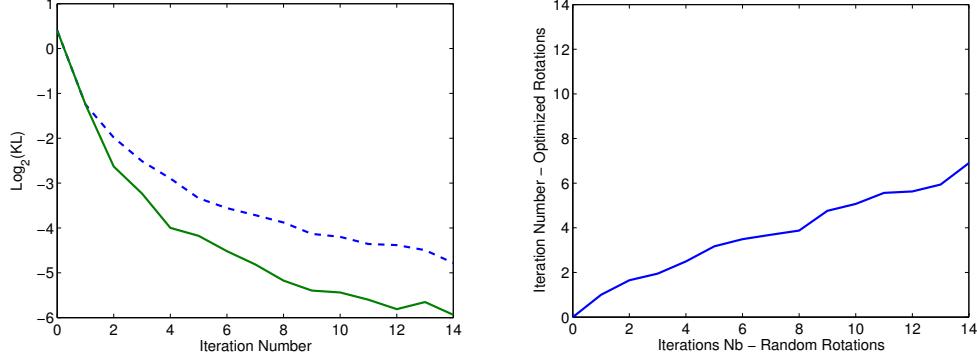


Fig. A.4. Optimised rotation sequence speed up for $N = 2$. On the left, the averaged evolution of the log Kullback Leibler divergence for 100 simulation for both a random sequence of rotations and an optimised sequence of rotations. On the right, the correspondence between iteration numbers for both strategies. It requires in average 53% less iterations for $N = 2$ when using an optimized rotation sequence.

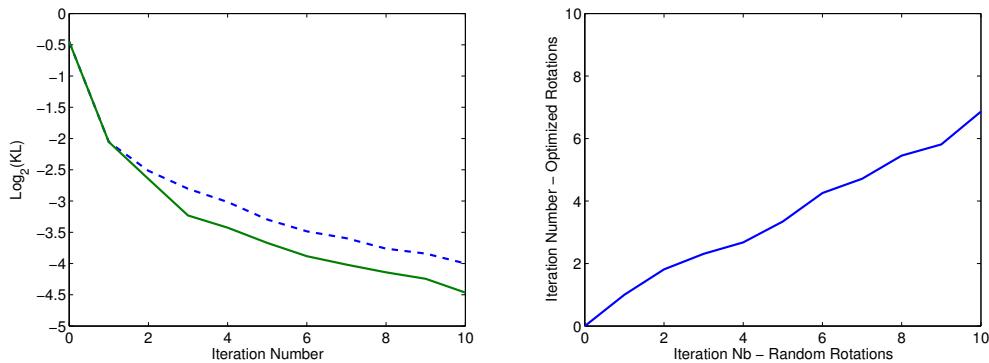


Fig. A.5. Optimised rotation sequence speed up for $N = 3$. On the left, the averaged evolution of the log Kullback Leibler divergence for 100 simulation for both a random sequence of rotations and an optimised sequence of rotations. On the right, the correspondence between iteration numbers for both strategies. It requires in average 33% less iterations for $N = 3$ when using an optimized rotation sequence.

Table A.2
Optimised rotations for $N = 3$

No.	1	2	3	4
x	1.000000 0.000000 0.000000	0.333333 0.666667 0.666667	0.577350 0.211297 0.788682	0.577350 0.408273 0.707092
y	0.000000 1.000000 0.000000	0.666667 0.333333 -0.666667	-0.577350 0.788668 0.211352	-0.577350 -0.408224 0.707121
z	0.000000 0.000000 1.000000	-0.666667 0.666667 -0.333333	0.577350 0.577370 -0.577330	0.577350 -0.816497 0.000029
No.	5	6	7	8
x	0.332572 0.910758 0.244778	0.243799 0.910726 0.333376	-0.109199 0.810241 0.575834	0.759262 0.649435 -0.041906
y	-0.910887 0.242977 0.333536	0.910699 -0.333174 0.244177	0.645399 0.498377 -0.578862	0.143443 -0.104197 0.984158
z	-0.244295 0.333890 -0.910405	-0.333450 -0.244075 0.910625	0.756000 -0.308432 0.577351	0.634780 -0.753245 -0.172269
No.	9	10	11	12
x	0.862298 0.503331 -0.055679	0.982488 0.149181 0.111631	0.687077 -0.577557 -0.440855	0.463791 0.822404 0.329470
y	-0.490221 0.802113 -0.341026	0.186103 -0.756525 -0.626926	0.592440 0.796586 -0.120272	0.030607 -0.386537 0.921766
z	-0.126988 0.321361 0.938404	-0.009074 0.636722 -0.771040	-0.420643 0.178544 -0.889484	-0.885416 0.417422 0.204444

- [3] L. Neumann, A. Neumann, Color style transfer techniques using hue, lightness and saturation histogram matching, in: Proceedings of Computational Aesthetics in Graphics, Visualization and Imaging, 2005, pp. 111–122.
- [4] C. M. Wang, Y. H. Huang, A novel color transfer algorithm for image sequences, *Journal of Information Science and Engineering* 20 (6) (2004) 1039–1056.
- [5] F. Pitié, A. Kokaram, R. Dahyot, N-Dimensional Probability Density Function Transfer and its Application to Colour Transfer, in: International Conference on Computer Vision (ICCV'05), Beijing, 2005.
- [6] H. Kotera, A scene-referred color transfer for pleasant imaging on display, in: Proceedings of the IEEE International Conference on Image Processing, 2005, pp. 5–8.
- [7] J. Morovic, P.-L. Sun, Accurate 3d image colour histogram transformation, *Pattern Recognition Letters* 24 (11) (2003) 1725–1735.
- [8] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Addison Wesley, 1992.
- [9] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, A. N. Venetsanopoulos, Vector Filtering for Color Imaging, *IEEE Signal Processing Magazine* 22 (1) (2005) 74–86.
- [10] R. Lukac, K. N. Plataniotis, A Taxonomy of Color Image Filtering and Enhancement Solutions, *Advances in Imaging and Electron Physics* 140 (2006) 187–264.
- [11] P. Pérez, A. Blake, M. Gangnet, Poisson image editing, *ACM Transactions on Graphics* 22 (3) (2003) 313–318.
- [12] M. Pappas, I. Pitas, Digital Color Restoration of Old Paintings, *Transactions on Image Processing* (2) (2000) 291–294.
- [13] E. Pichon, M. Niethammer, G. Sapiro, Color histogram equalization through mesh deformation, in: IEEE International Conference on Image Processing (ICIP'03), 2003, pp. II 117–120.
- [14] D. Ruderman, T. Cronin, C. Chiao, Statistics of Cone Responses to Natural Images: Implications for Visual Coding, *Journal of the Optical Society of America* (8) (1998) 2036–2045.
- [15] M. Grundland, N. A. Dodgson, Color Histogram Specification by Histogram Warping, in: Proceedings of the SPIE, Vol. 5667, 2004, pp. 610–624.
- [16] L. Lucchese, S. K. Mitra, a new Method for Color Image Equalization, in: Proceedings of the IEEE International Conference on Image Processing (ICIP'01), 2001, pp. 133–136.
- [17] H. L. Shen, J. H. Xin, Transferring color between three-dimensional objects, *Applied Optics* 44 (10) (2005) 1969–1976.
- [18] T. Welsh, M. Ashikhmin, K. Mueller, Transferring Color to Greyscale Images, in: Proceedings of ACM SIGGRAPH, San Antonio, 2002, pp. 227–280.
- [19] Y. Ji, H.-B. Liu, X.-K. Wang, Y.-Y. Tang, Color Transfer to Greyscale Images using Texture Spectrum, in: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 2004, pp. 4057–4061.
- [20] J. Jia, J. Sun, C.-K. Tang, H.-Y. Shum, Bayesian correction of image intensity with spatial consideration, in: Proceedings of the 8th European Conference on Computer Vision (ECCV'04), 2004, pp. 342–354.
- [21] Y. H. Chang, S. Saito, M. Nakajima, Color transfer between images based on basic color category E68D (12) (2003) 2780–2785.
- [22] Y. H. Chang, K. Uchikawa, S. Saito, Example-based color stylization based on categorical perception, in: Proceedings of the 1st Symposium on Applied perception in graphics and visualization (APGV), ACM Press, 2004, pp. 91–98.
- [23] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.
- [24] E. W. Weisstein, Radon transform, from MathWorld—A Wolfram Web Resource.
URL <http://mathworld.wolfram.com/RadonTransform.html>

- [25] D. Comaniciu, V. Ramesh, P. Meer, The Variable Bandwidth Mean Shift and Data-Driven Scale Selection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV'01), Vancouver, Canada, 2001, pp. 438–445.
- [26] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 1992.
- [27] Wikipedia, Hypersphere.
URL <http://en.wikipedia.org/wiki/Hypersphere>
- [28] F. Pitié, PDF Transfer - Thesis Material (2006).
URL www.sigmedia.tv/research/PDFTransfer/
- [29] R. Fattal, D. Lischinski, M. Werman, Gradient domain high dynamic range compression, in: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02), ACM Press, New York, NY, USA, 2002, pp. 249–256.
- [30] J. Weickert, B. ter Haar Romeny, M. Viergever, Efficient and Reliable Schemes for Nonlinear Diffusion Filtering, *IEEE Transactions on Image Processing* 7 (3) (1998) 398–410.
- [31] F. Pitié, R. Dahyot, F. Kelly, A. C. Kokaram, A new robust technique for stabilizing brightness fluctuations in image sequences, in: 2nd Workshop on Statistical Methods in Video Processing in conjunction with ECCV 2004, Prague, Czech Republic, 2004.