User Interaction for Image Recolouring using $\mathcal{L}_2$

Mairéad Grogan
Trinity College Dublin
Dublin, Ireland
mgrogan@tcd.ie

Rozenn Dahyot
Trinity College Dublin
Dublin, Ireland
dahyotr@tcd.ie

Aljosa Smolic
Trinity College Dublin
Dublin, Ireland
smolica@scss.tcd.ie

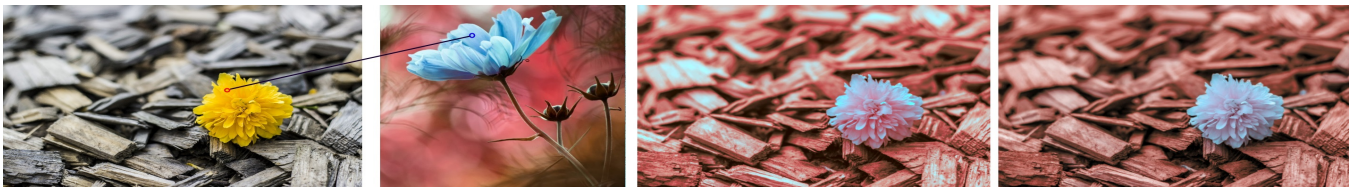

Figure 1: The target and palette images, with the user selected correspondences shown on the left. On the right we show the results when only the user selected correspondences are used to compute the colour transfer (column 3), and finally the results when additional correspondences are added (column 4).

ABSTRACT

Recently, an example based colour transfer approach proposed modelling the colour distributions of a palette and target image using Gaussian Mixture Models, and registers them by minimising the robust $\mathcal{L}_2$ distance between the mixtures. In this paper we propose to extend this approach to allow for user interaction. We present two interactive recolouring applications, the first allowing the user to select colour correspondences between a target and palette image, while the second palette based application allows the user to edit a palette of colours to determine the image recolouring. We modify the $\mathcal{L}_2$ based cost function to improve results when an interactive interface is used, and take measures to ensure that even when minimal input is given by the user, good colour transfer results are created. Both applications are available through a web interface and qualitatively assessed against recent recolouring techniques.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**;

KEYWORDS

Colour transfer, palette based image recoloring, $\mathcal{L}_2$ Registration

ACM Reference format:
Mairéad Grogan, Rozenn Dahyot, and Aljosa Smolic. 2017. User Interaction for Image Recolouring using $\mathcal{L}_2$. In *Proceedings of 14th European Conference on Visual Media Production (CVMP 2017), London, United Kingdom, December 11–13, 2017 (CVMP 2017),* 10 pages.
https://doi.org/10.1145/3150165.3150171

CVMP 2017, December 11–13, 2017, London, United Kingdom

ACM ISBN 978-1-4503-5329-8/17/12…$15.00
https://doi.org/10.1145/3150165.3150171

1 INTRODUCTION

Manipulating colours in images is an important step in many applications, including creating aesthetic effects in image and film post production, image and video restoration and image corrections which facilitate further image processing. Several methods have been proposed to generate the desired recolouring result given an input image and in many cases, a palette image is provided by the user indicating the colour distribution that the result image should have [18][10]. However, an image that includes the exact colour changes desired by the user may not always be available. Even when a palette image is provided, the results generated may look very different to what the user imagined. For this reason, many interactive recolouring approaches have been proposed, which give the user more control over the recolouring result. Some methods allow the user to select colour correspondences between the target and palette image, while others propagate sparse user edits throughout the image [15][6]. Recently, a number of palette based recolouring methods have been proposed, which are easy to use and understand [5][23]. However, they can produce unnatural results when palette entries are changed to darker or brighter colours [5].

In this paper, we extend a recent example based colour transfer approach [9–11] to include user interaction, and show that these extensions give users more freedom when editing their images, giving timely feedback about the results generated. Our first application allows the user to select correspondences between a target and palette image, and the second allows them to make changes to a target image by manipulating a palette of colours. We show qualitatively that both applications create good results, and that our palette based recolouring application outperforms a recent technique [5], avoiding the unnatural recolouring results that this algorithm can create.

2 STATE OF THE ART

One of the first colour transfer algorithms was an example based technique proposed by Reinhard et al. [18], requiring a target and palette image as input. Their method registers the statistics of the target and palette colour distributions in LAB space. Since

then, many other statistical approaches have also been proposed, along with histogram matching and optimal transport solutions [22][19][17][16][8]. When the target and palette image capture the same scene, or have many colour correspondences between them, techniques have been proposed to take advantage of these colour correspondences to improve the colour transfer result [13][12][14]. Other methods propose using the textural information from the target and palette images to guide the colour transfer [3]. However, many of these methods rely on the presence of a large number of correspondences between images, and cannot be applied when there are none or few available.

Recently, Grogan and Dahyot [9–11] presented a technique which can be applied to both palette and target images with and without correspondences. They estimate a parametric colour transfer function by minimising the $\mathcal{L}_2$ distance between Gaussian Mixture Models (GMMs) modelling the images' colour distributions. The robust $\mathcal{L}_2$ cost function can be easily augmented to include pixel correspondences, and outperforms similar state of the art techniques when correspondence outliers are present [13]. They can also interpolate between several colour transfer results, similar to the recently proposed Wasserstein approaches [4]. While their results perform better than other example based state of the art algorithms [8, 13, 17], their approach requires a palette image as input, which may not always be available. We propose to extend their approach to include user interaction in Section 3.

Many other techniques which allow user interaction in the recolouring process have also been proposed, including stroke based approaches and edit propagation [15][2][5][23]. In [15], Oskam et al. propose a colour transfer technique which allows user interaction when selecting colour correspondences between images, and was extended by Croci et al. for film colour restoration [7]. Recently, palette based recolouring techniques have become popular as they are easy to use, allowing the user to make changes to a small palette of colours when recolouring an image [20][5][23]. Chang et al. [5] introduce a palette based recolouring tool that generates results in real time. However, as demonstrated in our experiments, their algorithm can produce unnatural results especially when palette colours are mapped to colours that are a lot brighter or darker than the original palette entry. Other methods propose using a soft segmentation approach [20][1][23]. Aksoy et al. [1] propose to decompose the image into layers, with each layer associated with a palette colour. They then recolour each layer separately and combine them to create the final image. While this approach allows local colour changes to be made, it can be quite slow, and seems unnecessary when only small colour changes need to be made. Zhang et al. [23] define the colour of each pixel in the image as a linear combination of the palette colours, and recolour the image by recombining the new palette colours for every pixel. Their approach is significantly quicker than that of Aksoy et al. Our proposed technique generates similar results to Zhang et al. while performing much quicker than Aksoy et al.

3 INTERACTIVE $\mathcal{L}_2$ BASED COLOUR TRANSFER

In [9–11], Grogan and Dahyot propose to model the colour distributions of the target and palette images using GMMs, and register them by minimising the $\mathcal{L}_2$ distance between them. The GMMs $p_p(x)$ and $p_t(x|\theta)$, model the colour distributions of the pixels in the palette and transformed target images respectively, and are defined as

$$p_p(x) = \sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(x; \mu_p^{(k)}, h^2\mathrm{I}) \tag{1}$$

and

$$p_t(x|\theta) = \sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(x; \mu_\theta^{(k)}, h^2\mathrm{I}). \tag{2}$$

Here, the random vector $x \in \mathbb{R}^3$ takes values in a 3 dimensional colour space, K represents the number of Gaussians in each mixture, with $\{\mu_p^{(k)}\}_{k=1,..K}$ and $\{\mu_\theta^{(k)}\}_{k=1,..K}$ the Gaussian means, and each Gaussian is associated with the same isotropic covariance matrix, $h^2\mathrm{I}$. Here $\{\mu_\theta^{(k)} = \phi(\mu_t^{(k)}, \theta)\}$ has been computed by transforming $\{\mu_t^{(k)}\}$ by some transformation ϕ which depends on θ. The goal is to estimate the parameter θ, controlling the transformation ϕ, which registers $p_t(x|\theta)$ to $p_p(x)$.

When there are no pixel correspondences between the target and palette images available, Grogan and Dahyot let $\{\mu_t^{(k)}\} = \{z_t^{(k)}\}$, and $\{\mu_p^{(k)}\} = \{z_p^{(k)}\}$, the K cluster centres estimated by applying the K-means algorithm to the target or palette image respectively, and minimise:

$$C(\theta) = \|p_t\|^2 - 2\langle p_t | p_p \rangle \tag{3}$$

where

$$\langle p_t | p_p \rangle = \sum_{k=1}^{K} \sum_{l=1}^{K} \frac{1}{K^2} \mathcal{N}(0; \phi(\mu_t^{(k)}, \theta) - \mu_p^{(l)}, 2h^2\mathrm{I}) \tag{4}$$

and

$$\|p_t\|^2 = \sum_{k=1}^{K} \sum_{l=1}^{K} \frac{1}{K^2} \mathcal{N}(0; \phi(\mu_t^{(k)}, \theta) - \phi(\mu_t^{(l)}, \theta), 2h^2\mathrm{I}). \tag{5}$$

This cost function is a trade off between moving p_t as close as possible to p_p (Eq. 4), while ensuring that p_t does not collapse into a unique Gaussian distribution (Eq. 5).

When there are n colour correspondences $\{(c_t^{(k)}, c_p^{(k)})\}_{k=1,\cdots,n}$ between the target and palette images available, they instead let $\{\mu^{(k)}\} = \{c^{(k)}\}$ and minimise $-\langle p_t | p_p \rangle$, where

$$\langle p_t | p_p \rangle = \sum_{k=1}^{n} \frac{1}{n^2} \mathcal{N}(0; \phi(\mu_t^{(k)}, \theta) - \mu_p^{(k)}, 2h^2\mathrm{I}) \tag{6}$$

Here, the term $\|p_t\|^2$ is removed from the cost function as the correspondences ensure that p_t does not collapse into a unique Gaussian distribution.

The authors investigate a variety of functions for the transformation ϕ, and show that the Thin Plate Spline transformation gives better results in general.

In this paper we propose two extensions to this colour transfer application which enables user interaction, giving the user more control over the appearance of the result image. The methods used to create these applications are described in the following sections.

Following [9–11], we model the target and palette distributions, $p_t(x)$ and $p_p(x)$, as GMMs (cf. Eq. (1) and (2)) with mixture centres combining both the K-means cluster centres extracted from target

and palette images (noted z_t and z_p), and a set of correspondences $\{(c_t^{(k)}, c_p^{(k)})\}_{k=1,\cdots,n}$:

$$\{\mu_t^{(k)}\}_{k=1,\cdots,K+n} = \{c_t^{(k)}\}_{k=1..n} \cup \{z_t^{(k)}\}_{k=1,..K} \tag{7}$$

and

$$\{\mu_p^{(k)}\}_{k=1,\cdots,K+n} = \{c_p^{(k)}\}_{k=1..n} \cup \{z_p^{(k)}\}_{k=1,..K} \tag{8}$$

When a palette image is not available, an alternative choice for $\{z_p^{(k)}\}_{k=1,..K}$ is proposed in Section 3.4. The random vector $x \in \mathbb{R}^3$ takes values in a 3 dimensional colour space chosen as Lab colour space with each component scaled in between 0 and 255[1].

To allow for interactive colour transfer applications, we extend the colour transfer technique proposed by Grogan et al. [10] by combining the cost functions $C(\theta)$ (defined with Equations (3), (4) and (5) without correspondences), with an added interaction term $\langle p_{t_c} | p_{p_c} \rangle$ computed with user defined correspondences (Eq. (6)) and a penalty term E to constraint pixel values to stay inside the range 0 to 255 after transformation by the estimated warping function ϕ:

$$\tilde{C}(\theta) = C(\theta) - \lambda \langle p_{t_c} | p_{p_c} \rangle + \gamma \sum_{k=1}^{K+n} E(\mu_\theta^{(k)}) \tag{9}$$

where parameters λ and γ control the effects of the different terms (cf. Section 3.5).

Having n correspondences computed in a semi-supervised manner (Sec. 3.2 and 3.3), the interaction term $\langle p_{t_c} | p_{p_c} \rangle$ is defined as

$$\langle p_{t_c} | p_{p_c} \rangle = \sum_{k=1}^{n} \frac{1}{n^2} \mathcal{N}(0; \phi(c_t^{(k)}, \theta) - c_p^{(k)}, 2h^2 \mathrm{I}). \tag{10}$$

with only the selected correspondences $\{(c_t^{(k)}, c_p^{(k)})\}_{k=1,\cdots,n}$, while the term $C(\theta)$ is computed with all Gaussian components as defined in Eq. (7) and (8).

3.1 Penalty term E

The final term in this cost function, $E(\mu_\theta)$, penalises parameters θ which map the colours $\mu_\theta = \phi(\mu_t, \theta)$ outside of the colour space. It is defined on each component L, a and b of the colour space as follows:

$$E(\mu_\theta) = \sum_{\diamond = \{L,a,b\}} \mathcal{E}(\mu_\theta^\diamond) \tag{11}$$

with the function $\mathcal{E}$ defined from $\mathbb{R}$ to $(0;1)$:

$$\mathcal{E}(\mu_\theta^\diamond) = \frac{-1}{1+\exp(-\mu_\theta^\diamond)} + \frac{1}{1+\exp(-(\mu_\theta^\diamond - 255))} + 1 \tag{12}$$

The function $\mathcal{E}$ is plotted in Figure 2. This regularization term extends Grogan and Dahyot's original algorithm [10, 11] where recasting values in between the normal range is addressed as a post-processing operation instead. While their original algorithm often performs well, we found that when users select colour correspondences, colours are mapped out of gamut more frequently, hence making such a regularisation term necessary.

[1]We use OpenCV's cvtColor to convert RGB to LAB, giving LAB values from 0 to 255.

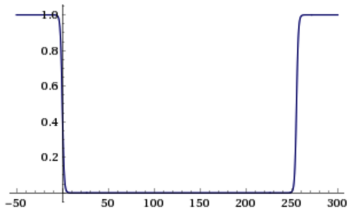

Figure 2: Function $\mathcal{E}$ used to penalise pixel values outside the range 0 and 255.

3.2 User defined correspondences

Our first web based application allows the user to choose a target and palette image (cf. Fig. 3, top: left image is target, middle is the palette image with correspondences appearing as line segments between the two, and the right image is the recoloured result), and select colour correspondences $\{(c_t^{(k)}, c_p^{(k)})\}_{k=1,\cdots,n}$ between the images to indicate which colours in the palette and target image should correspond after recolouring. Our second web based application (cf. Fig. 3, bottom) does not provide a palette image but instead allows the user to define how specific colours in the target image are changed. For instance in Fig. 3, the user proposes to transform green into yellow, leave brown as brown and pink as pink, change violet to green, and leave black as black.

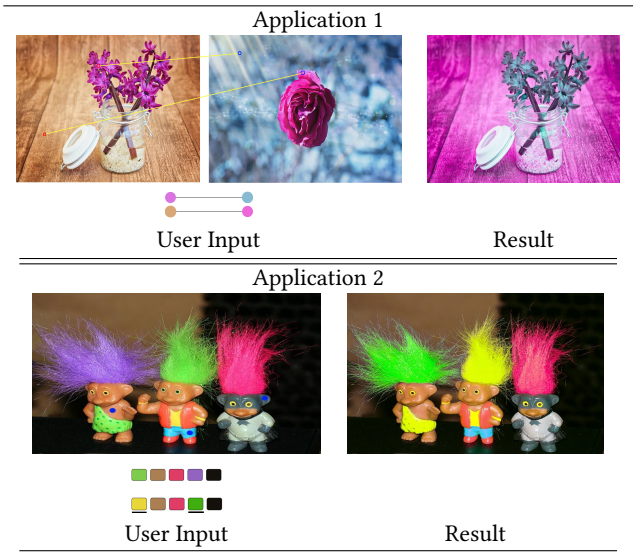

Figure 3: The inputs provided by the user in our proposed applications. Top row: Application 1. The user provides a target and palette image, and colour correspondences indicating which colours should match after recolouring. Bottom row: Application 2. The user provides a target image and a palette is automatically generated. The user can edit the palette, to change the colours of the image. They can also add constraints to the target image (blue dots) indicating which colours should remain unchanged after recolouring.

3.3 Unsupervised correspondence augmentation

In general, as manual interaction can be a tedious operation, user defined correspondences are only available in small numbers in practice (e.g. $n \leq 7$) and when testing our proposed colour transfer algorithm, we found that minimising Eq. 9 often did not create the desired colour transfer result. For instance, Figure 1 shows that when using only the user selected correspondences, the yellow colour of the flower changes to light blue, while the darker yellow colours become pink. To overcome this problem, we found that given a user defined correspondence $(\{u_t^L, u_t^a, u_t^a\}, \{u_p^L, u_p^a, u_p^b\})$[2], adding additional correspondences between colours that lie close to the original pair in LAB space, boosts the information provided by the user and improves the colour transfer result. For example, given a user selected correspondence, we also create correspondences between brighter and darker versions of these colours, computed based on their luminance values in LAB space. This ensures that darker and brighter versions of the user defined correspondences are mapped in the same way (cf. Fig 1). We add 4 additional correspondences between pairs in our first application, and 8 in our second application (cf. Appendix A) as fewer colours are available to model the palette distribution in our second application, and therefore more correspondences are needed.

3.4 Palette based Recolouring

For palette based Recolouring (cf. Application 2, Fig. 3) the user can recolour a target image by editing a palette of colours. They can also click on the target image to add constraints, indicating colours in the target image which should remain the same after recolouring. As no palette image is provided by the user in this case, cluster centres $\{z_p^{(k)}\}_{k=1,..K}$ are not available (Eq. (8)) and the distribution p_p would only be composed of Gaussian components generated from (augmented) user defined correspondences. Therefore, in this case, as well as defining the correspondence set $\{(c_t, c_p)\}$, we also need to define the set $\{z_p^{(k)}\}$, made up of colours from the target image and the user defined palette representing the desired colour distribution of the result image .

3.4.1 Creating the Palette. Given the target image, we use the method proposed by Chang et al. [5] to automatically generate a set of m colours representing the dominant colours in the image, where m is selected by the user. This set of colours can be edited by the user, creating m pairs of correspondences $\{(u_t^{(i)}, u_p^{(i)})\}_{i=1,\cdots,m}$. This set of correspondences is split in two sets:

- $\{(u_t^{new(i)}, u_p^{new(i)})\}$ - pairs that change colour (i.e. $u_t \neq u_p$),
- $\{(u_t^{c(i)}, u_p^{c(i)})\}$ - pairs acting as constraints (i.e $u_t = u_p$).

3.4.2 Selecting $\{\mu_t\}$ and $\{\mu_p\}$. The set of cluster centres $\{\mu_t\}$ is defined as before (Eq. (7)): the K cluster centres $\{z_t^{(i)}\}$ are provided using K-means on the target image and the selection of $\{c_t^{(i)}\}$ is explained in Section 3.4.3.

There is no palette image available however to define a set of clusters $\{z_p^{(i)}\}$. We propose to select palette centres $\{z_p^{(i)}\}$ amongst the target centres $\{z_t^{(i)}\}$ such that: z_t is selected as a cluster centre z_p only if z_t is different enough from the changing colours $\{u_t^{new(i)}\}$. Our strategy analyses the relative position of z_t w.r.t. to colours that remain the same $\{u_t^{c(i)}\}$ and the changing colours $\{u_t^{new(i)}\}$ using Dirichlet tesselations [21] where z_t is associated with the closest u_t: if that u_t is a changing colour then z_t is not considered as a candidate for z_p, alternatively if u_t is not changing colour (constraint) then $z_p = z_t$.

3.4.3 Computing $\{(c_t, c_p)\}$. In this palette based application 2, the set of correspondences is defined as:

(1) **User defined correspondences** using a palette (c.f. Section 3.4.1).
(2) **Unsupervised correspondence augmentation** (cf. Section 3.3).
(3) **Unsupervised constraint augmentation** We automatically add additional constraints to points in $\{z_t^{(i)}\}$ which lie close to the unchanging colours $\{u_t^{c(i)}\}$. For each colour u_t^c, we set constraints on it's 5 nearest neighbours in $\{z_t^{(i)}\}$, and use Dirichlet tessellation to ensure that these new constraints do not lie too close to the changing colours $\{u_p^{new(i)}\}$.

Note that in our web interface, the user can also provide more constraints about colours that need to remain the same after recolouring. The application is interactive and results are updated every time the user provides more correspondences. The colour transfer function is estimated by minimising Equation 9.

3.5 Optimisation and Parameter Settings

As in [10], we choose ϕ to be a Thin Plate Spline transformation:

$$\phi(x, \theta) = \mathrm{A}\, x + t + \sum_{j=1}^{m} w_j\, \psi(\|x - c_j\|) \tag{13}$$

where $\psi(x) = -x$ is the 3D TPS basis function, and $\{c_j\}$ are the control points. The parameter θ to be estimated is made up of the matrix A, the translation t and the TPS warping coefficients $\{w_j\}$. A regularisation term added to the cost function to control the non-linearity of the TPS transformation:

$$\tilde{C}(\theta) + \zeta \int \|D^2\phi(x, \theta)\|^2 dx \tag{14}$$

with $\|D^2\phi(x,\theta)\|^2 = \sum_{i,j\in\{1,\cdots,d\}} \left(\frac{\partial^2\phi}{\partial x_i \partial x_j}\right)^2$ with $d = 3$. We also use a simulated annealling strategy, which slowly decreases the scale parameter h over several iterations of the optimisation. This ensures that our algorithm can overcome local minima [10]. The parameter value for h at the final iteration is 10. The values of the other parameters are: $\lambda = 0.5$, $\zeta = 3e^{-5}$ and $\gamma = \frac{1}{h*(K+n)}$. As our parameter γ depends on the scale parameter h, at the early stages of the simulated annealing process less weight is given to the regularization term enforcing that the transformed colours lie in gamut. This allows the algorithm to transform the points more freely at the early stages of this process, ensuring that local minima are avoided and a better solution is estimated.

[2] $\{u_t^L, u_t^a, u_t^b\}$ denotes the colour u_t in Lab space

3.6 Web based User Interface

Both of our applications have a web based interface and are available to use online [3]. The first application allows the user to upload a target and palette image and select colour correspondences between the images to guide the colour transfer. Feedback is given to the user after each new pair of correspondences is selected. Our palette based application allows the user to change a palette of colours associated with the input image using a HSL colour picker. The user can also select pixels in the input image that they would like to constrain after recolouring. Both applications take 3-4 seconds to compute the new colour transfer results for an image with 1 million pixels.

4 EXPERIMENTAL RESULTS

4.1 Application 1

In Figure 5 we compare our first colour transfer application to the colour transfer technique proposed by Grogan and Dahyot [10], which does not take into account user defined colour correspondences. From these results we can see that while both techniques transfer the colours of the palette image to the target, our proposed approach allows the user to determine where particular colours should appear in the result image. In row 1 of Figure 5, the grass in the colour transfer result of Grogan and Dahyot [10] is brown, but can be changed by adding a correspondence between the grass in the target and palette using our approach. The colour of the flowers has also been adjusted to make them brighter. Similarly in rows 2-5, the user can easily change how objects are recoloured using a small number of correspondences.

4.2 Application 2

4.2.1 Comparison with state of the art. In Figure 6, we compare our second application to that of Chang et al. [5]. Given a target image and the palette of colours automatically generated from it (Figure 6, column 1), we edit some of the palette colours and present the recolouring results of both our approach and Chang et al. in Columns 2 and 3. In the first row, our colour transfer technique successfully changes the colours of the red and green apples without disrupting the colour of the yellow apple or the background, unlike Chang et al. Similarly in row 2, while Chang et al's technique changes the colour of the boat to blue, areas of the water and ground are also recoloured. Again, our technique successfully recolours the boat without making unexpected changed to other parts of the image.

In rows 3, 4 and 5 of Figure 6 we can also see the effect that changing a palette entry to a darker or brighter colour has on Chang et al's result images. In row 3, when the yellow door is recoloured to dark blue, the rest of the image becomes darker, as does the image of the flower in row 4. In row 5, when the red bow is recoloured to yellow, the girl's face becomes a lot brighter. These changes can also be seen in the new palettes, shown below the result images. For example, in row 5, column 2, while the only palette entry that was changed by the user was the third colour, from red to yellow, the second palette entry also automatically become a lot brighter. In all cases, our algorithm performs better than that of Chang et al., recolouring the door, flower and bow without making any unexpected changes to the rest of the image.

In Figures 7 and 8 we also compare our technique to that of Zhang et al. [23][4]. In Figure 7, we compare Chang et al., Zhang et al. and our approach, in the case where each method is given the same target image, original palette, and user modified palette. The original palette is shown below the target image (Column 1), and the user modified palette is shown below each result image, with the modified palette entries underlined in black. We see that both Zhang et al. and our method outperform Chang et al., which causes a loss of detail in the clothing in rows 2 and 3.

In Figure 8, both Zhang et al. and our technique are used to recolour the original image (Column 1) so that it matches the image given in Column 2. The images in column 2 have been created by an artist using Photoshop and are taken from [23]. While the results of both techniques are good, some areas of the results reported by Zhang et al. change colour unexpectedly. For example, in row 1, the blue colour of the sky becomes less vivid, in row 2 the grey wall become more yellow, and in row 3 the sky becomes more green in tone, while the grass becomes brown. Both user selected palette changes and user selected constraints were used to generate the results of our technique in Figure 8, and the user added constraints can help to avoid unexpected colour changes, such as those seen in the results reported by Zhang et al.

4.2.2 Adding User Constraints. In this section we explore the benefits of allowing the user to add constraints to the target image, indicating areas of the image that should remain unchanged after recolouring. These constraints can be added by the user after recolouring, when they have noticed areas of the image that may have changed colour unexpectedly. For example, in Figure 9, we present results of our colour transfer algorithm without (Column 2) and with (Column 3) constraints. In the first column we present the target image, with the original palette shown beneath it. This palette was edited by the user to create the result in column 2. The palette entries that were changed by the user are underlined in black. Some unexpected colour changes can be seen in the results in the second column and are highlighted in yellow. Areas of the water in the first row become more red, the girl's face in row 2 becomes cooler and the troll's shorts become green in row 3. Constraints can then be added by the user to the target image to reduce these changes. These constraints are represented by blue dots on the target image in Column 1. In column 3 we present the results of our recolouring taking into account both the palette changes and the user added constraints. In this case, the changes to the water (row 1), the girl's skin tone (row 2) and the troll (row 3) have been reduced. This constraint functionality gives the user more control over the final result, easily allowing them to reduce changes to particular areas of the image that may have changed when using the palette based recolouring approach.

4.2.3 Limitations. As ϕ is a TPS transformation, and is smooth by definition, we found that our technique was unable to successfully implement certain palette changes chosen by the user. This typically happens when several palette entries are changed to colours

[3]Demos and accompanying video available at: https://v-sense.scss.tcd.ie/?p=1289

[4]Results for [23] were taken from their paper

that are very different from their original (cf. Fig 4). As our transformation is global in nature, we also found that it could not recreate some results produced by local methods. In Figure 10 we compare our palette based recolouring result to Chang et al. and the soft-segmentation approach of Aksoy et al. [1]. While our algorithm produces better results than Chang et al. (Column 2), it does not perform as well as that of Aksoy et al. In row 1, when recolouring the jumper to pink, our result also recolours some of the girl's face to pink (see zoom in on right), while the face remains unchanged in Aksoy et al's result. In row 2, our method successfully recolours the flower to green without effecting the colour of the girl's face. However, errors occur where the hair and the flower overlap. Aksoy et al's technique successfully recolours the flower without any artifacts as it separates the pixels representing the hair from those of the flower before recolouring. Aksoy's technique is quite time consuming, while our framework gives quick feedback to the user, allowing them to easily experiment with new palettes and colours changes. Moreover our technique could also be extended to take advantage of segmentation maps, allowing the algorithm to fit a transfer function to each region of interest in the image.

Figure 4: If several palette entries are changed to very different values, the algorithm may not create a good result.

5 CONCLUSION

We have presented two interactive applications for image recolouring, both of which are based on an $\mathcal{L}_2$ optimisation function which registers a target and palette colour distribution, and is explicitly defined to take into account user selected correspondences. Both applications are easy to use, and available through a web interface, giving timely feedback to the user when a new colour correspondence is added or palette change made. We have shown qualitatively that the first creates good results when taking into account the user correspondences and a palette image, while the second outperforms a recent palette based recolouring technique [5].

ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776.

REFERENCES

[1] Y. Aksoy, T. O. Aydın, A. Smolić, and M. Pollefeys. 2017. Unmixing-Based Soft Color Segmentation for Image Manipulation. *ACM Trans. Graph.* 36, 2 (2017), 19:1–19:19.

[2] Xiaobo An and Fabio Pellacini. 2008. AppProp: All-pairs Appearance-space Edit Propagation. In *ACM SIGGRAPH 2008 Papers (SIGGRAPH '08)*. ACM, New York, NY, USA, Article 40, 9 pages. https://doi.org/10.1145/1399504.1360639

[3] Benoit Arbelot, Romain Vergne, Thomas Hurtut, and JoÃńlle Thollot. 2016. Automatic Texture Guided Color Transfer and Colorization. In *Non-Photorealistic Animation and Rendering*. The Eurographics Association. https://doi.org/10.2312/exp.20161060

[4] Nicolas Bonneel, Julien Rabin, Gabriel PeyrÃĬ, and Hanspeter Pfister. 2015. Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision* 51, 1 (2015), 22–45. https://doi.org/10.1007/s10851-014-0506-3

[5] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein. 2015. Palette-based Photo Recoloring. *ACM Transactions on Graphics (SIGGRAPH)* 34, 4 (July 2015).

[6] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang. 2014. Sparse Dictionary Learning for Edit Propagation of High-Resolution Images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2854–2861. https://doi.org/10.1109/CVPR.2014.365

[7] S. Croci, T. O. Aydin, N. Stefanoski, M. Gross, and A. Smolic. 2017. Advanced tools and framework for historical film restoration. *Journal of Electronic Imaging* 26, 1 (2017), 011021. https://doi.org/10.1117/1.JEI.26.1.011021

[8] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol. 2013. Regularized Discrete Optimal Transport. In *Scale Space and Variational Methods in Computer Vision*, A. Kuijper, K. Bredies, T. Pock, and H. Bischof (Eds.). Lecture Notes in Computer Science, Vol. 7893. Springer Berlin Heidelberg, 428–439. https://doi.org/10.1007/978-3-642-38267-3_36

[9] Mairead Grogan and Rozenn Dahyot. 2015. L2 Registration for Colour Transfer in Videos. In *Proceedings of the 12th European Conference on Visual Media Production (CVMP '15)*. ACM, New York, NY, USA, Article 16, 1 pages. https://doi.org/10.1145/2824840.2824862

[10] M. Grogan and R. Dahyot. 2017. Robust Registration of Gaussian Mixtures for Colour Transfer. *ArXiv e-prints* (May 2017). arXiv:cs.CV/1705.06091

[11] M. Grogan, M. Prasad, and R. Dahyot. 2015. L2 Registration for Colour Transfer. In *European Signal Processing Conference (Eusipco)*. Nice France. https://doi.org/10.1109/EUSIPCO.2015.7362799

[12] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. 2011. Non-rigid Dense Correspondence with Applications for Image Enhancement. *ACM Trans. Graph.* 30, 4, Article 70 (July 2011), 70:1–70:10 pages.

[13] Y. Hwang, J.-Y. Lee, I. S. Kweon, and S. J. Kim. 2014. Color Transfer Using Probabilistic Moving Least Squares. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3342–3349. https://doi.org/10.1109/CVPR.2014.427

[14] M. Oliveira, A. D. Sappa, and V. Santos. 2011. Unsupervised Local Color Correction for Coarsely Registered Images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*. IEEE Computer Society, Washington, DC, USA, 201–208. https://doi.org/10.1109/CVPR.2011.5995658

[15] T. Oskam, A. Hornung, R. W. Sumner, and M. Gross. 2012. Fast and Stable Color Balancing for Images and Augmented Reality. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. 49–56. https://doi.org/10.1109/3DIMPVT.2012.36

[16] N. Papadakis, E. Provenzi, and V. Caselles. 2011. A Variational Model for Histogram Transfer of Color Images. *Image Processing, IEEE Transactions on* 20, 6 (June 2011), 1682–1695. https://doi.org/10.1109/TIP.2010.2095869

[17] F. Pitié, A. C. Kokaram, and R. Dahyot. 2007. Automated Colour Grading Using Colour Distribution Transfer. *Comput. Vis. Image Underst.* 107, 1-2 (July 2007), 123–137. https://doi.org/10.1016/j.cviu.2006.11.011

[18] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. 2001. Color transfer between images. *Computer Graphics and Applications, IEEE* 21, 5 (Sep 2001), 34–41. https://doi.org/10.1109/38.946629

[19] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 747–754 vol. 1. https://doi.org/10.1109/CVPR.2005.215

[20] Baoyuan Wang, Yizhou Yu, Tien-Tsin Wong, Chun Chen, and Ying-Qing Xu. 2010. Data-driven Image Color Theme Enhancement. *ACM Trans. Graph.* 29, 6, Article 146 (Dec. 2010), 10 pages. https://doi.org/10.1145/1882261.1866172

[21] R. Webster and M.A. Oliver. 2001. *Geostatistics for Environmental Scientists*. John Wiley and Sons.

[22] Shuchang Xu, Yin Zhang, Sanyuan Zhang, and Xiuzi Ye. 2005. Uniform color transfer. In *IEEE International Conference on Image Processing 2005*, Vol. 3. III–940–3. https://doi.org/10.1109/ICIP.2005.1530548

[23] Q. Zhang, C. Xiao, H. Sun, and F. Tang. 2017. Palette-Based Image Recoloring Using Color Decomposition Optimization. *IEEE Transactions on Image Processing* 26, 4 (April 2017), 1952–1964. https://doi.org/10.1109/TIP.2017.2671779

A CORRESPONDENCE AUGMENTATION

Given $\{u_t, u_p\} = \{(u_t^L, u_t^a, u_t^b), (u_p^L, u_p^a, u_p^b)\}$, a user defined correspondence, we compute the scalar $t \in (-255; 255)$ as the difference between the luminances $t = u_p^L - u_t^L$. This correspondence is automatically augmented by four new correspondences when added in our first application (Rules (1) - (4)) and 8 when added in our second application(Rules (1) - (8)). The following rules are used for $t \in (-255; 0)$ (resp. $t \in (0; 255)$):

(1) $\{(0, u_t^a, u_t^b), (0, u_p^a, u_p^b)\}$
(resp. $\{(0, u_t^a, u_t^b), (0 + t, u_p^b, u_p^a)\}$)

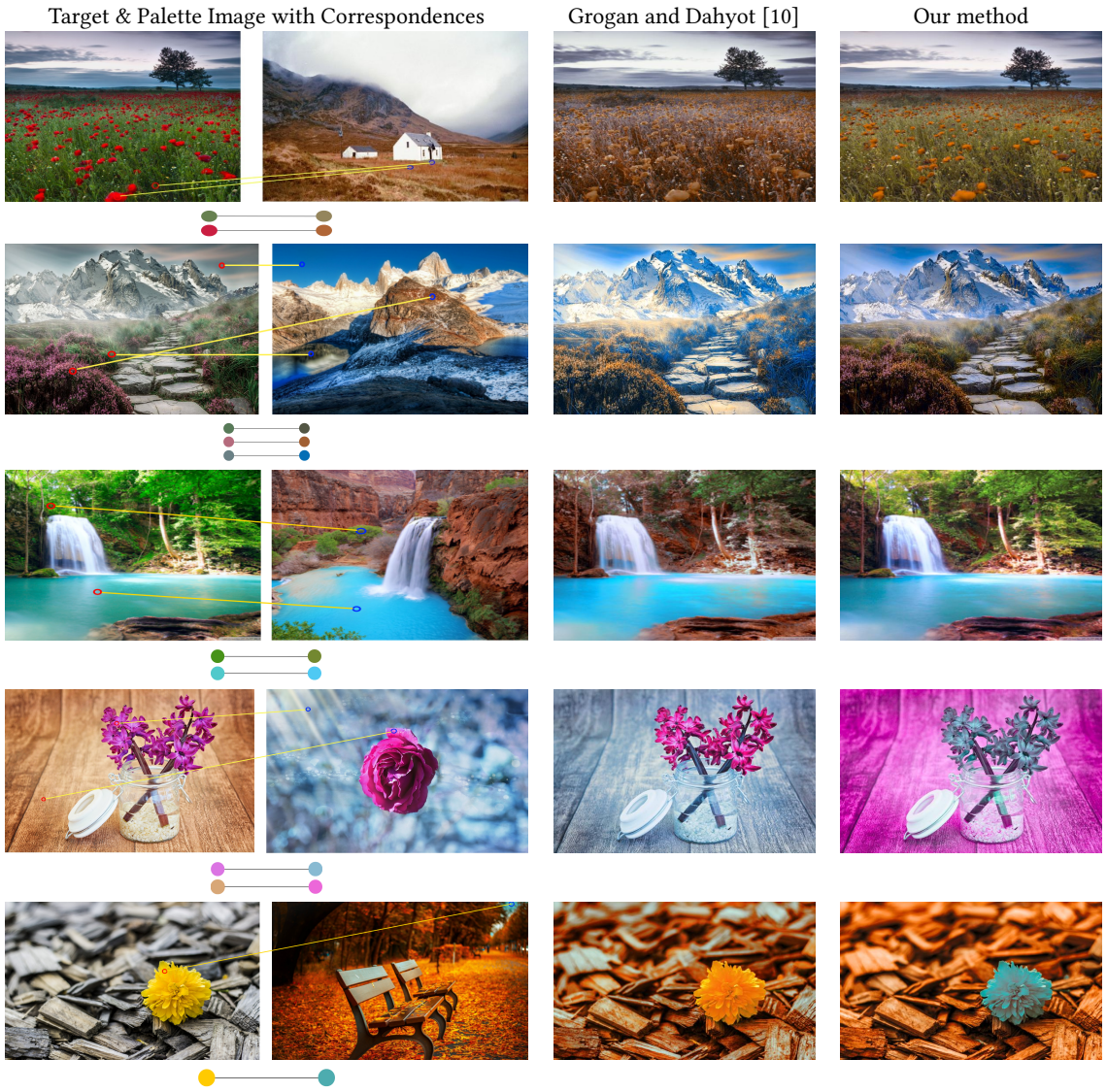

Figure 5: Column 1 and 2: The target and palette images with user selected correspondences joined by yellow lines. The correspondence colours are shown below. Column 3: Grogan and Dahyot's [10] colour transfer result (no correspondences used). Column 4: Our colour transfer result (with user defined correspondences). In row 1, the grass in Grogan and Dahyot's [10] result is brown, but by adding a correspondence, it can be easily changed to green. The colour of the flowers has also been adjusted to make them brighter. Similarly in rows 2-5, the user selected correspondences allow the user to easily change the recolouring result.

(2) $\{(255, u_t^a, u_t^b), (255 + t, u_p^a, u_p^b)\}$
(resp. $\{(255, u_t^a, u_t^b), (255, u_p^a, u_p^b)\}$)

(3) $\{(\frac{255+u_t^L}{2}, u_t^a, u_t^b), (\frac{255+u_t^L}{2} + t, u_p^a, u_p^b)\}$
(resp. $\{(\frac{u_t^L}{2}, u_t^a, u_t^b), (\frac{u_t^L}{2} + t, u_p^a, u_p^b)\}$)

(4) $\{(\frac{u_t^L}{2}, u_t^a, u_t^b), (\frac{u_p^L}{2}, u_p^a, u_p^b)\}$ If $t \in (-255; \frac{-u_t^L}{2})$.
$\{(\frac{u_t^L}{2}, u_t^a, u_t^b), (\frac{u_t^L}{2} + t, u_p^a, u_p^b)\}$ if $t \in (\frac{-u_t^L}{2}; 0)$
(resp. $\{(\frac{255+u_t^L}{2}, u_t^a, u_t^b), (\frac{255+u_p^L}{2}, u_p^a, u_p^b)\}$ if $t \in (\frac{255-u_t^L}{2}; 255)$
$\{(\frac{255+u_t^L}{2}, u_t^a, u_t^b), (\frac{255+u_t^L}{2} + t, u_p^a, u_p^b)\}$.) if $t \in (0; \frac{255-u_t^L}{2})$

(5) $\{(u_t^l, max(u_t^a + 10, 255), u_t^b), (u_p^l, max(u_p^a + 10, 255), u_p^b)\}$

(6) $\{(u_t^l, min(u_t^a - 10, 0), u_t^b), (u_p^l, min(u_p^a - 10, 0), u_p^b)\}$

(7) $\{(u_t^l, u_t^a, max(u_t^b + 10, 255)), (u_p^l, u_t^a, max(u_t^b + 10, 255))\}$

(8) $\{(u_t^l, u_t^a, min(u_t^b - 10, 0)), (u_p^l, u_t^a, min(u_t^b - 10, 0))\}$

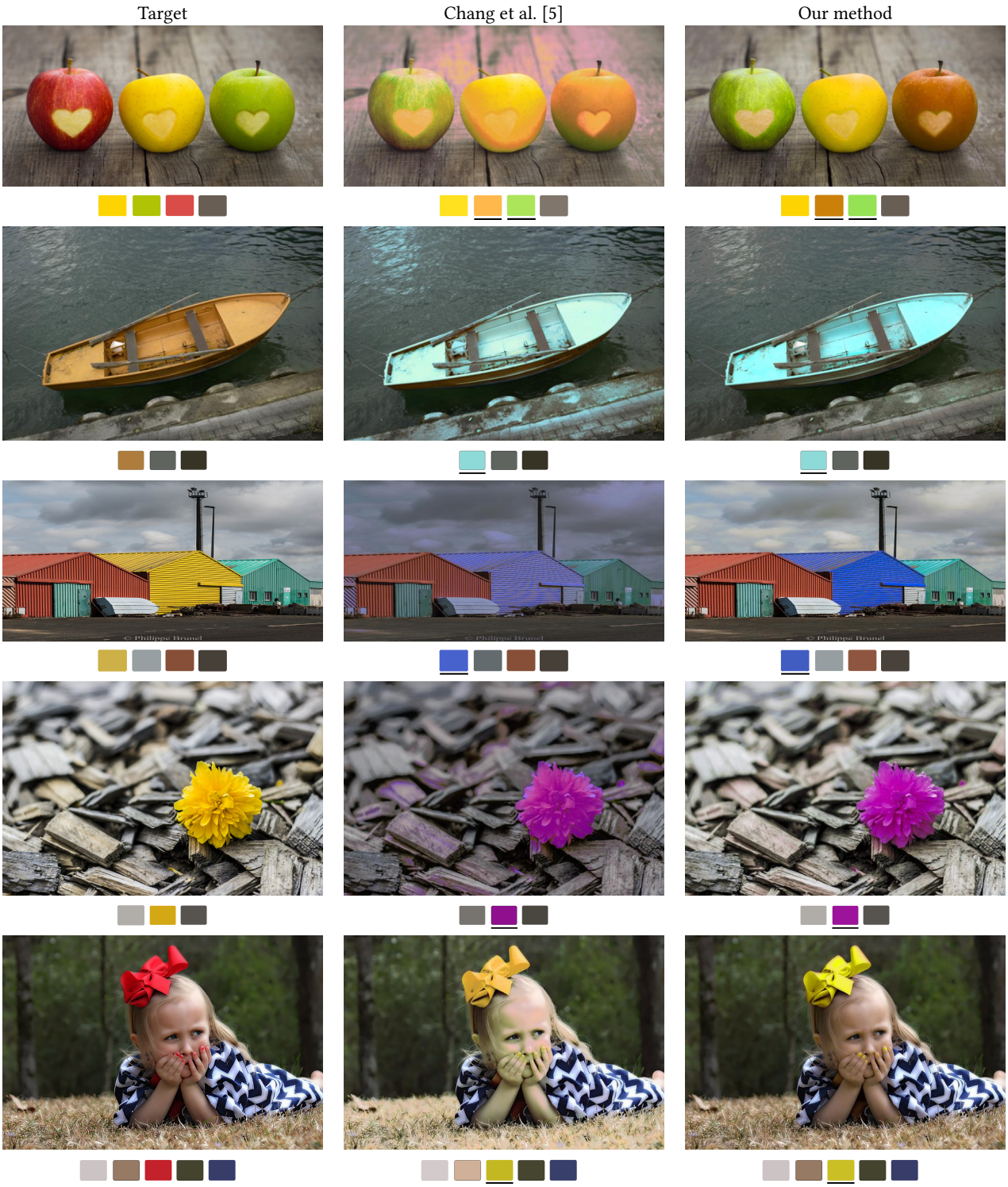

Figure 6: A comparison of our second application to that of Chang et al. [5]. Column 1: The original image (original palette used for recolouring shown beneath it); Column 2: Chang et al. [5] result, with the new user edited palette shown beneath it. The palette entries modified by the user are underlined in black. Column 3: Results of our method with user edited palette shown beneath. Chang et al's method makes unexpected changes to the yellow apple (row 1) and to the water (row 2). Their result images in row 3 and 4 appear a lot darker as a palette entry is changed to a darker colour, and the girl's face in row 5 is brightened considerably. In all cases our algorithm performs better than that of Chang et al.

Figure 7: Here, each recolouring technique is given the same target image (Column 1), original palette (shown below the target image), and user modified palette (shown below each result image. Modified palette entries are underlined in black). We compare the results of Chang et al. (Column 2), Zhang et al. (Column 3) and our approach (Column 4). Both Zhang et al. and our method outperform Chang et al., which causes a loss of detail in the clothing in rows 2 and 3.

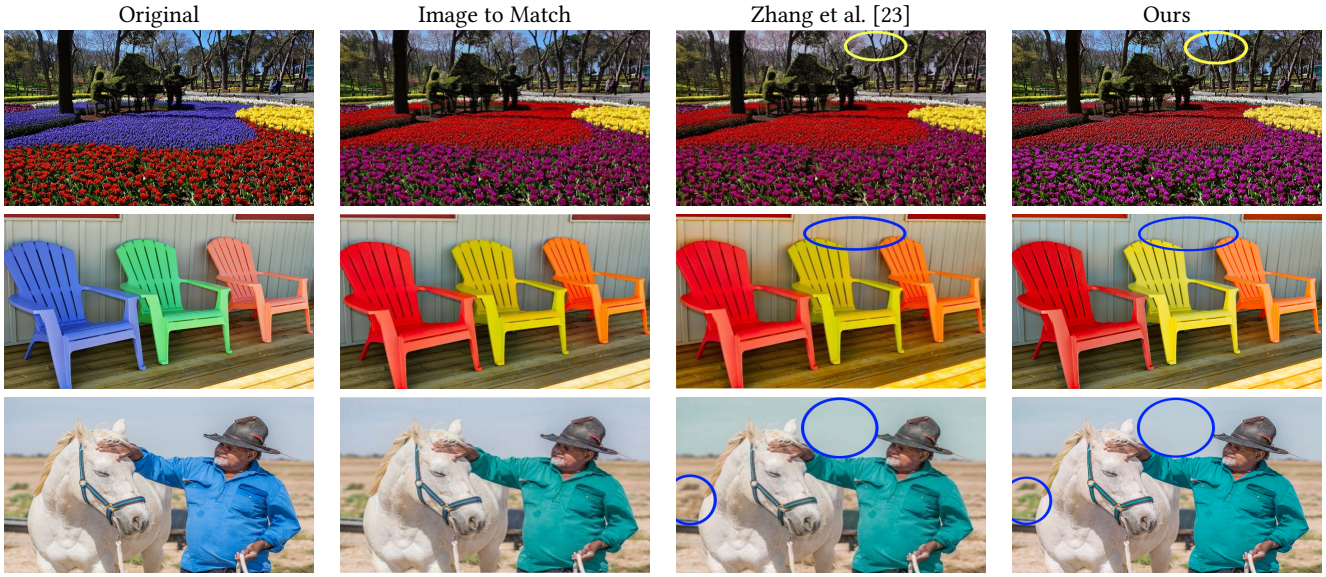

Figure 8: Here, Zhang et al. (Column 3) and our technique (Column 4) are used to recolour the original image (Column 1) so it matches the image given in Column 2. Some areas of Zhang et al.'s results change colour unexpectedly. In row 1, the blue sky becomes less vivid, in row 2 the grey wall become more yellow, and in row 3 the sky becomes more green in tone, while the grass becomes brown. Both user selected palette changes and constraints were used to generate the results of our technique, and user added constraints can help to avoid unexpected changes such as those in the results reported in Zhang et al.

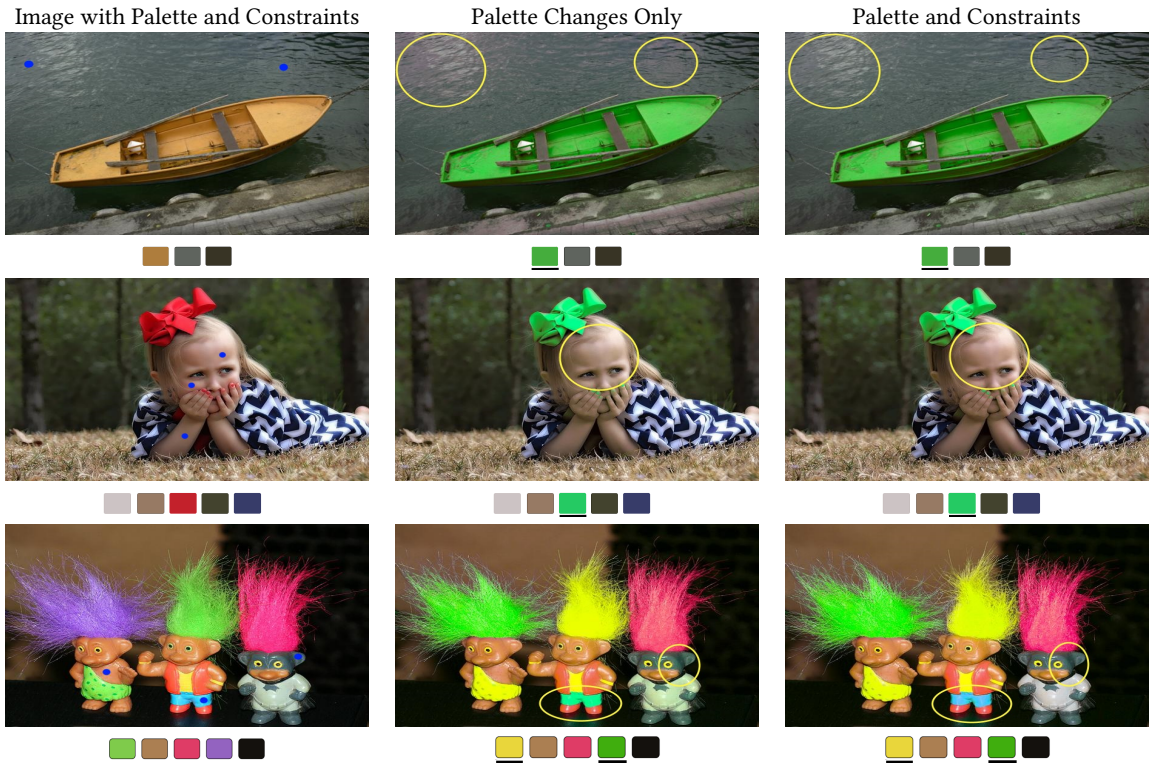

Figure 9: Here we present results of our colour transfer algorithm without (Column 2) and with (Column 3) constraints. Column 1: The target image, with the original palette shown beneath it. User constraints were also added to the target image (blue dots). Column 2: The result of our algorithm taking into account the palette changes only, and not the constraints. Some unexpected colour changes can be seen, and are highlighted in yellow. The water becomes more red (row 1), the girl's face becomes cooler (row 2), and the troll's shorts become green (row 3). Column 3: Results of our recolouring taking into account both the palette changes and the user added constraints. Here, the changes to the water (row 1), the girl's skin tone (row 2) and the troll (row 3) have been reduced.

Figure 10: Column 1: The original image on the left, with a close up section shown on the right; Column 2: Chang et al. [5] result, with a close up on the right ; Column 3: The results from our approach, with a close up on the right Column 4: Aksoy et al. [1] result, with close up on the right. In row 1, although many of the colour artifacts introduced by Chang et al's technique are removed using our approach, the colour of girl's face in the bottom left of the image (see zoom in on the right) still contains some pink around the eyes, which the soft segmentation approach can overcome.