

# Simple Primary Colour Editing for Consumer Product Images

Han Gong<sup>1</sup>, Luwen Yu<sup>2</sup>, and Stephen Westland<sup>2</sup>

<sup>1</sup>School of Computing Sciences, University of East Anglia, UK

<sup>2</sup>School of Design, University of Leeds, UK

## Abstract

*We present a simple primary colour editing method for consumer product images. We show that by using colour correction and colour blending, we can automate the pain-staking colour editing task and save time for consumer colour preference researchers. To improve the colour harmony between the primary colour and its complementary colours, our algorithm also tunes the other colours in the image. A preliminary experiment has shown some promising results compared with a state-of-the-art method and human editing.*

## Introduction

Research and data have been increasingly used to optimise the design process [33]. Previous research shows that product-colour appearance can affect consumers' purchase decisions while consumers' product-colour preferences vary from category to category [20, 34]. To understand consumer-product colour preference, the standard marketing images are manually recoloured using software such as Adobe Photoshop [3] or GIMP [1]. The recolouring process requires researchers (or designers) to manually adjust colours by picking colours and adopting non-binary per-layer masking. However, visible artefacts and incompatible background colours often remain even after very careful editing. We conclude four main requirements from colour preference researchers:

**Minimum machine processing time.** Users prefer a rapid processing speed as usually colour modifications are applied to multiple products for at least dozens or more target colours.

**Minimum user manipulation time.** Users prefer minimal interaction time. Methods that require multiple user strokes or manual selections of multiple colours would not be ideal. We need a method that only requires a single primary colour specification and takes care of the rest of the colour processing automatically.

**Artefact resiliency.** Artefacts, such as JPEG blocks and unnatural edges, are usually introduced after re-colouring. The algorithm should preserve all image details except for the primary colour modification.

**Colour harmony preservation.** The chosen colour for change may not fit the product's existing complementary colours. In some cases, tuning of complementary colours is desirable.

In this paper, our proposed method addresses these requirements by providing an alternative design tool which is fully automatic. Existing studies suggested that colour manipulations offer the potential for software to generate recoloured images (target colour images). More promising applications of automatic colour manipulations will encourage generative design systems in colour

and design [28, 33]. There have been a number of methods for colour manipulations such as colour transfer [23, 26, 27], colour hint propagation [4, 8–10, 19], or palette editing [7, 24, 35]. However, none of the previous methods is directly applicable to the primary colour editing problem. Rapid digital workflows in practice would also require automatic methods for evaluating and/or comparing colours and designs [5].

In this paper, we propose a simple method which automates primary colour editing at an optimised consumption of user and machine processing time and it preserves colour harmony to some extent. Our method is based on the assumption that simulating colour change as a 2-D colour homography [13] (*i.e.* as a change of light) usually avoids image processing artefacts [13, 16] such as JPEG blocks, sharp edges, and colour combination conflicts. Our colour editing pipeline is depicted in Figure 1 where the colour editing task is reformulated as a 2-D colour homography colour correction problem. Additionally, we may apply a gradient preservation step to remove some residual artefacts. Compared with the previous recolouring methods, ours requires minimum user input and its design is relatively simple.

## Related Work

Our work is relevant to the colour editing methods in three categories: A) colour transfer; B) colour hint propagation; C) palette-based colour editing.

### Colour transfer

Colour transfer is an image editing process which adjusts the colours of a picture to match a target picture's colour theme. This research was originated by Reinhard *et al.* [27] and recently followed up by the others [23, 25, 26] recently. Most of these methods align the colour distributions in different colour spaces, which usually involve statistics alignment [23, 25, 27] or iterative distribution shifting [26].

### Colour hint propagation

Some methods require user hints, *e.g.* strokes, to guide re-colouring of object surfaces. This direction of research was started by Levin *et al.* [19] where they colourise grey-scale images based on user colour stroke and solves for a large and sparse system of linear equations. Their key assumption is that the colours of neighbouring pixels with similar luminance should have similar chromaticities. More recent methods [4, 9, 10] make use of masks, either soft or hard, to assist re-colourisation. Their colour modification model is based on a diagonal colour correction matrix used for white balance, *e.g.* [15] with limits on the range of applicable colour changes. Some others, *e.g.* [8], have used sparse coding/learning that the sparse set of colour samples provide an

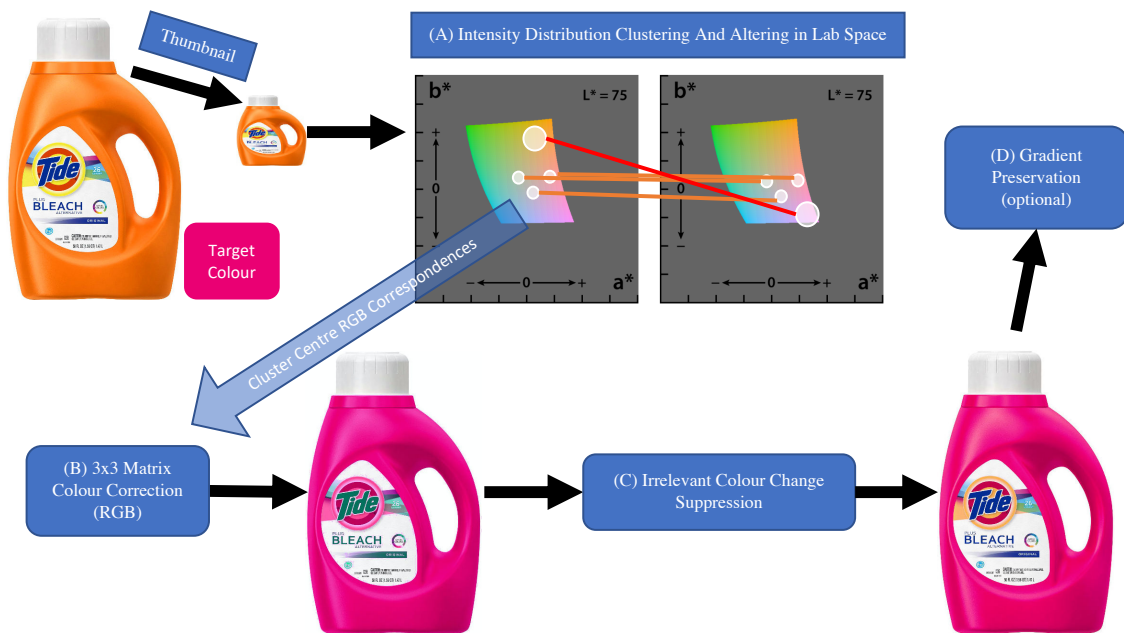


Figure 1: Primary colour editing pipeline. Given an input image, our method amends the product’s primary colour according to a target colour in 3-4 steps: A) Colour intensities clusters in CIE  $L^*a^*b^*$  colour space [29] are computed (left: original distributions; right: primary colour altered distributions); B) Cluster RGB correspondences are used for estimating a colour correction matrix. Note that the  $L^*$  channel intensities are also used for clustering but are not illustrated on the exemplar graphs; C) An alpha-blending process is applied to remove colour changes, which are less relevant to the primary colour change, from the colour corrected image; D) Some residual colour artefacts can be optionally removed using a gradient preservation method ‘regrain’ [26] (see the latter section for visualisation). The  $a^*b^*$  chromaticity gamut images are taken from Wikipedia [2].

intrinsic basis for an input image and the coding coefficients capture the linear relationship between all pixels and the samples. This branch of methods require heavy user inputs and therefore not immediately useful for our problem.

### Palette-based colour editing

Some methods adopt colour intensity clustering, *e.g.*  $k$ -means++ algorithm [6], to initially generate a colour palette of the input image. After palette adjustments, different approaches were applied for manipulating colour changes. Zhang *et al.* [35] decompose the colours of the image into a linear combination of basis colours before reconstructing a new image using the linear coding coefficients. Chang *et al.* [7] adopt a monotonic luminance mapping and radial basis functions (RBFs) for interpolating/mapping chromaticities. This branch of methods are most close to our solution however none of them is optimised for the particular task of rapid primary colour editing for consumer product images.

### Colour homography

Our solution is based on the colour homography colour change model. The colour homography theorem [11–13, 16] presents that chromaticities across a change in capture conditions (light color, shading and imaging device) are a homography apart. Suppose that we map an RGB  $\underline{\rho}$  to a corresponding RGI (red-

green-intensity)  $\underline{c}$  using a  $3 \times 3$  full-rank matrix  $C$ :

$$\underline{\rho}^T C = \underline{c}^T$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R \\ G \\ R+G+B \end{bmatrix}^T \quad (1)$$

The  $r$  and  $g$  chromaticity coordinates are written as  $r = R/(R+G+B)$ ,  $g = G/(R+G+B)$ . We treat the right-hand-side of Equation 1 as a homogeneous coordinate and we have  $\underline{c} \propto [r \ g \ 1]^T$ . When the shading is fixed, it is known that across a change in illumination or a change in device, the corresponding RGBs are approximately related by a  $3 \times 3$  linear transform  $M$  that  $\underline{\rho}^T M = \underline{\rho}'^T$  where  $\underline{\rho}'$  is the corresponding RGBs under a second light or captured by a different camera [21, 22]. We have  $H = C^{-1}MC$  which maps colours in RGI form between illuminants. Due to different shading, the RGI triple under a second light is written as  $\underline{c}'^T = \alpha \underline{c}^T H$ , where  $\alpha$  denotes the unknown scaling. Without loss of generality we regard  $\underline{c}$  as a homogeneous coordinate *i.e.* assume its third component is 1. Then,  $[r' \ g']^T = H([r \ g]^T)$  (rg chromaticity coordinates are a homography  $H()$  apart). In this paper, we will model the major colour change initially as a colour homography change but without considering the individual scale differences between

each RGB correspondences, *i.e.* a  $3 \times 3$  linear transform of colour change is applied.

### Simple Primary Colour Editing

Our algorithm starts with the simple observation that a simple 2-D colour homography model allows for a wider range of colour changes (as opposed to a diagonal colour correction matrix) and usually produces fewer colour combination conflicts [13, 16]. In Figure 1, we overview the colour processing pipeline which consists of three major steps and one optional step: A) Clustering: The CIE L\*a\*b\* [29] intensities of an input RGB image are clustered using MeanShift [14]. The primary colour cluster is altered to match the target colour (see the red line) that the cluster centres form the before-and-after sparse colour intensities correspondences; B) Colour correction: The L\*a\*b\* colour correspondences are converted to RGB space before being used to estimate a 2-D colour homography matrix (without considering scale differences); C) Irrelevant colour change suppression: a soft alpha-blending mask is computed to suppress aggressive colour changes irrelevant to the primary colour change; D) Gradient preservation (optional): a gradient preservation step can be applied to remove more residual artefacts. We also note that the computational cost can be reduced by using down-sampled thumbnail images for model parameter estimation. We provide the algorithm details in the following sub-sections.

### Intensity clustering and altering

To estimate a reliable colour change model, the first step is to extract the predominant colours which best capture the input image's colour theme. We adopt MeanShift [14] clustering to extract at most 5 predominant colours (*i.e.* cluster centres) from the input image. The intention of not collecting too many colours is to avoid noise and reduce computational cost. The cluster number of 5 is only an empirical value, *e.g.* 6 also works. Clearly, a fixed set of MeanShift parameters never guarantee a maximum number of 5 colour clusters. We thus propose a simple adaptive MeanShift clustering procedure which gradually increases the initial small kernel bandwidth value as shown in Algorithm 1 where

**Algorithm 1:** Adaptive MeanShift clustering

---

```

1  $w = 0.1, \beta = 1.5;$ 
2 repeat
3    $U = \text{MeanShift}(w, A_{lab});$ 
4    $n = \text{len}(U);$ 
5    $w = \beta w;$ 
6 until  $n > 5;$ 

```

---

**MeanShift** is the MeanShift function with a flat kernel and bandwidth  $w$ ,  $U$  is a  $n \times 3$  matrix of cluster centres (each row is a L\*a\*b\* intensity vector), **len** counts the number of cluster centres  $n$ ,  $\beta$  is a factor controlling the kernel width growth rate in each iteration.

Given the obtained predominant colours, we construct the sparse colour correspondences to be supplied for colour change model estimation. Since we aim to only change the one primary colour if possible, the remaining of target predominant colours are kept the same as the original predominant colours except that the only primary colour is modified as the target primary colour.

Through this, we construct a target predominant colours denoted as  $D$  (see also Figure 1 (A) for illustration).

### Colour Homography colour change

Given the source and target RGB colours denoted as  $N \times 3$  matrices  $C$  and  $D$  ( $N$  is the number of pixels), we make use of a simple 2-D colour homography matrix to achieve primary colour change while minimising colour artefacts. A full colour homography change is an optimised chromaticity mapping in RGB space. However, since the brightness of colour matters in this application, we omit the shading factor  $\alpha$  and only estimate a  $3 \times 3$  linear matrix transform (which is still a homography matrix) using weighted least-squares as the follows:

$$M = (C^T W C + k I_{3 \times 3})^{-1} C^T W D \quad (2)$$

where  $k = 10^{-3}$  is a regularisation term,  $W$  is a  $N \times N$  diagonal matrix whose diagonal elements are the associated normalised weights of all the predominant colours (*i.e.* cluster sizes),  $I_{3 \times 3}$  is a  $3 \times 3$  identity matrix. Denoting the RGB intensities of the input image as a  $N \times 3$  matrix  $A$ , we can compute its primary-colour-changed RGB intensities as  $B = AM$ . An intermediate processed example can be found in Figure 1 (B).

### Irrelevant colour change suppression

Some of the colour changes after the  $3 \times 3$  linear transform may look aggressive, *e.g.* the pink ring of the 'Tide' logo in Figure 1 (B). We adopt an alpha-blending procedure to address this as the follows:

$$B' = (1 - \text{diag}(\underline{d}))B + \text{diag}(\underline{d})A \quad (3)$$

where  $B'$  is the modified RGB colour output,  $\underline{d}$  is an  $N$ -vector denoting per-pixel scaling factors (in the range of  $[0, 1]$ ) and **diag**() places an  $N$ -vector along the diagonal of an  $N \times N$  diagonal matrix. Our intuition is to smoothly reduce the impact of the colour changes that are irrelevant to the primary colour and control this by  $\underline{d}$ . We measure the irrelevance by the a\*b\* chromaticity difference  $\Delta E$  between each colour (row) in  $B'$  and the target primary colour:

$$\Delta E = \sqrt{\Delta a^{*2} + \Delta b^{*2}} \quad (4)$$

where  $\Delta a^*$  and  $\Delta b^*$  are the errors in a\*b\* channels. We assumed an sRGB to CIE La\*b\* conversion and a D65 white point because we deal with rendered JPEG images by default. The colour space conversion formula can, of course, be changed to suit raw input image situation. A higher  $\Delta E$  indicates a higher degree of irrelevance but this value can be sometimes too big. Thus, we further cap and normalise  $\Delta E$  as  $\Delta E'$ :

$$\Delta E' = \begin{cases} 1 & \Delta E > \Delta E_{max} \\ \Delta E / \Delta E_{max} & \text{Otherwise} \end{cases} \quad (5)$$

where  $\Delta E_{max}$  is an upper threshold value. The individual  $\Delta E'$  is assigned as the corresponding element of  $\underline{d}$ . The processing result can be sensitive to  $\Delta E_{max}$  and thus  $\Delta E_{max}$  must be carefully chosen. An exemplar visualisation of  $\underline{d}$  in its image grid form is shown in Figure 2 (A). Aiming at obtaining a blending result

which preserves the edge details of the original image, we look for the optimum  $\Delta E_{max}$  which minimises Equation 6.

$$\min_{\Delta E_{max}} \sum_{c \in \{a^*, b^*\}} \text{entropy}(|\text{edge}(I_{B',c}) - \text{edge}(I_{A,c})|) \quad (6)$$

where  $||$  is the operator to output per element absolute value of a matrix,  $c$  indicates an intensity channel of  $a^*$  or  $b^*$ ,  $I_{B',*}$  and  $I_{A,*}$  indicate the grid-form images of the RGB intensity matrix  $B'$  and  $A$  respectively, **edge** is a binary edge detector using the Sobel approximation [32] to the derivative (without edge-thinning), **entropy** is a function which measures the amount of information – entropy [31] – as defined in Equation 7.

$$\text{entropy}(\underline{p}) = -\sum_i p_i \log p_i \quad (7)$$

where  $\underline{p}$  is a normalised input vector (summed up to 1) which, in our case, is the 1-D intensity array of the error-of-edge image (e.g. Figure 2 (C)),  $i$  is an element index. When the entropy of the error of two edge images is low, it indicates a higher similarity of edge features between two intensity images. However, we do not have a closed-form solution for its global minima. In practice, a suitable local minima in a reasonable range usually serves the purpose. We thus propose a brutal search for a local minima solution of  $\Delta E_{max}$  in the range of [10, 210] with an interval precision of 20. A visualised example of  $\underline{d}$  and its plot of  $\Delta E_{max}$  search are shown in Figure 2.

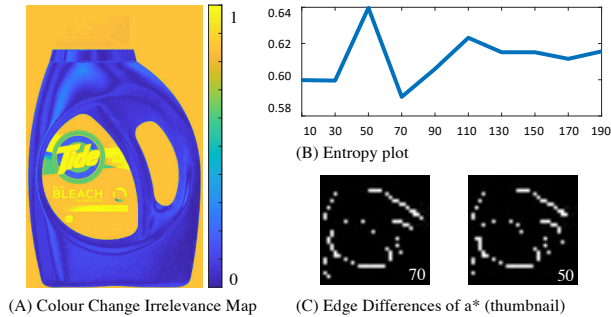


Figure 2: Visualisations of the alpha blending. A) Visualised alpha blending mask used in the irrelevant colour change suppression step. B) The associated entropy plot (horizontal ticks:  $\Delta E_{max}$ ). C) Visualisation of binary edge difference in channel  $a^*$  when  $\Delta E_{max} = 70$  or  $\Delta E_{max} = 50$  (more different). The bottom-right number indicates the value of  $\Delta E_{max}$ . See also Figure 1 (A) for the input image and the target primary colour.

### Artefact cleansing

As the previous alpha blending step has attempted the minimisation of edge artefacts, mostly users can get an artefact-free output image. However, for some rare cases, we also adopt an optional artefact cleansing step called 'regrain' which was first proposed in [26]. It provides strong gradient preservation but also has side effects which may cause minor undesired blurs along edges. Please refer to the cited paper for the algorithm details. Figure 3 shows an example where this optional step improves the result by removing some JPEG block artefacts.



Figure 3: Example of the 'regrain' [26] artefact cleansing.

### Acceleration

Our colour manipulation pipeline requires the solution of 10 key model parameters, namely  $M$  and  $\Delta E_{max}$ . Using full-resolution images is not necessary and we therefore adopt thumbnail images ( $32 \times 32$ ) for solving for  $M$  and  $\Delta E_{max}$  and apply the estimated parameters to a full-resolution input image to get a full-resolution output.

### Evaluation

In this section, we present the result comparison and some useful discussions about our method's practical use-cases.

### Results

We compare our method with a state-of-the-art palette-based re-colouring method [7] and the manually edited results produced by a professional colour preference researcher. Figure 4 shows some visual result comparisons. We found that our outputs are mostly comparable to the manually edited results which take 2-5 minutes' labour time per image. Most of the human labour time is spent on masking the image (for primary colour pixels). Once the mask is completed, the remaining recolouring time takes about 1 minute. All the results in Figure 4 have been produced without the 'regrain' step enabled. Our method also has some failure cases as shown in Figure 5. The first failure case might be caused by ill-conditioned Equation 2 as the number of correspondences is much lower than the number of model parameters. The second case was caused by the initial step of colour clustering. When the input image only has one colour, the MeanShift clustering algorithm can mishandle the primary colour extraction. Lowering the maximum cluster number (i.e. 5 in Algorithm 1) can resolve this issue. That said, we could provide this as an optional parameter for users.

Our method provides practical editing efficiency without user interventions. Using the thumbnail acceleration trick, our unoptimised MATLAB implementation (without the regrain step) takes about 1s to process an 1.2 mega-pixel image on a MacBook Pro 2015 laptop (2.5 GHz Quad-Core Intel Core i7 CPU).

### Discussions

'Work to forecast' has suggested the use of colour to forecast consumer demand or resource-saving levels [30]. Colour has also been suggested as one of the most powerful visual elements in packaging. Thus, choosing an appropriate colour for the design of packaging or product can significantly affect consumer decision-making [18]. This work could be applied as a product-colour predictor for studying product-packaging-colour in consumer purchase behaviour. Or, as an image generation tool, it



Figure 4: Result visual comparison. The target colours are shown at the top right of the input images. The label 'Human' refers to the column of results manually generated by a professional colour preference researcher.

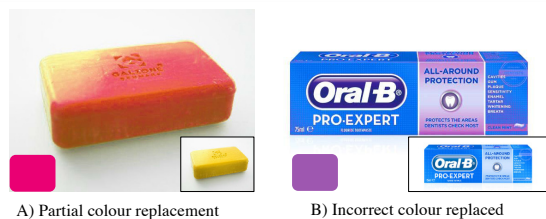


Figure 5: Failure cases. The input images are shown at the bottom right and the target primary colours are shown at the bottom left. A) Only a part of the primary colour was replaced. B) An incorrect primary colour was picked and replaced.

helps designers/researchers preview the multi-colour options of a product image.

We also acknowledge that more rigorous user experiments in controlled lighting/display conditions could still be possibly carried out after the UK Covid-19 lockdown [17]. We, therefore, have made our full code available <sup>1</sup> for the research community hoping that its evaluation and more potential use-cases can be further driven by the other cross-discipline communities.

## Conclusion

In this paper, we present a simple product re-colouring method for assisting consumer colour preference research. We show that by using a colour manipulation pipeline, we can automate this primary colour editing task for consumer colour preference researchers. The complementary colours in the product image will also be adjusted to potentially make the primary colour fits better. Future work is required to explore more of its use-cases and strengthen its artefact resiliency.

## Acknowledgment

We also thank Dr Qianqian Pan from the University of Leeds for her useful discussions.

## References

- [1] *GNU Image Manipulation Program (GIMP)*, 2020. <https://www.gimp.org> [Accessed: 06/04/2020].
- [2] *L\*a\*b\* gamut*, 2020. [https://en.wikipedia.org/wiki/CIELAB\\_color\\_space](https://en.wikipedia.org/wiki/CIELAB_color_space) [Accessed: 06/04/2020].
- [3] *PhotoShop*, 2020. <https://www.adobe.com/uk/products/photoshop.html> [Accessed: 06/04/2020].
- [4] Xiaobo An and Fabio Pellacini. Approp: all-pairs appearance-space edit propagation. In *ACM Transactions on Graphics (TOG)*, pages 1–9. 2008.
- [5] David Anderson, K Blake Perez, Zack Xuereb, Kevin Otto, and Kris Wood. Design processes of design automation practitioners. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2018.
- [6] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

<sup>1</sup>Github Code: [https://github.com/hangong/prod\\_recolor](https://github.com/hangong/prod_recolor)

- [7] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. Palette-based photo recoloring. *ACM Transactions on Graphics*, 34(4):139–1, 2015.
- [8] Xiaowu Chen, Dongqing Zou, Jianwei Li, Xiaochun Cao, Qingping Zhao, and Hao Zhang. Sparse dictionary learning for edit propagation of high-resolution images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2854–2861, 2014.
- [9] Xiaowu Chen, Dongqing Zou, Qingping Zhao, and Ping Tan. Manifold preserving edit propagation. *ACM Transactions on Graphics (TOG)*, 31(6):1–7, 2012.
- [10] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Diffusion maps for edge-aware image editing. *ACM Transactions on Graphics*, 29(6):1–10, 2010.
- [11] Graham D. Finlayson, Han Gong, and Robert B. Fisher. Color homography. In *Progress in Colour Studies*. John Benjamins Publishing Company, 2016.
- [12] Graham D. Finlayson, Han Gong, and Robert B. Fisher. Color homography color correction. In *Color Imaging Conference*. Society for Imaging Science and Technology, 2016.
- [13] Graham Finlayson, Han Gong, and Robert B Fisher. Color homography: theory and applications. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):20–33, 2019.
- [14] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.
- [15] Han Gong. Convolutional mean: A simple convolutional neural network for illuminant estimation. In *British Machine Vision Conference*. BMVA, 2019.
- [16] Han Gong, Graham D. Finlayson, and Robert B. Fisher. Recoding color transfer as a color homography. In *British Machine Vision Conference*. BMVA, 2016.
- [17] Richard Horton. Offline: Covid-19 and the nhs—“a national scandal”. *The Lancet*, 395(10229):1022, 2020.
- [18] Hannele Kauppinen-Räsänen. Strategic use of colour in brand packaging. *Packaging Technology and Science*, 27(8):663–676, 2014.
- [19] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM Transactions on Graphics*, pages 689–694. 2004.
- [20] Dan Luo, Luwen Yu, Stephen Westland, and Nik Mahon. The influence of colour and image on consumer purchase intentions of convenience food. *Journal of the International Colour Association*, 24:11–23, 2019.
- [21] L.T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *Journal of the Optical Society of America A*, 3:1673–1683, 1986.
- [22] D.H. Marimont and B.A. Wandell. Linear models of surface and illuminant spectra. *Journal of the Optical Society of America A*, 9(11):1905–1913, 92.

- [23] R. M. H. Nguyen, S. J. Kim, and M. S. Brown. Illuminant aware gamut-based color transfer. *Computer Graphics Forum*, 33(7):319–328, October 2014.
- [24] Rang MH Nguyen, Brian Price, Scott Cohen, and Michael S Brown. Group-theme recoloring for multi-image color consistency. In *Computer Graphics Forum*, volume 36, pages 83–92. Wiley Online Library, 2017.
- [25] F Pitié and A Kokaram. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. In *European Conference on Visual Media Production*. IET, 2007.
- [26] François Pitié, Anil C. Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision And Image Understanding*, 107(1-2):123–137, July 2007.
- [27] Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics And Applications*, 21(5):34–41, September 2001.
- [28] Othman Sbai, Mohamed Elhoseiny, Antoine Bordes, Yann LeCun, and Camille Couprie. Design: Design inspiration from generative networks. In *European Conference on Computer Vision*, pages 0–0, 2018.
- [29] Janos Schanda. Cie colorimetry. *Colorimetry: Understanding the CIE system*, pages 25–78, 2007.
- [30] Richard J Schonberger. Reconstituting lean in healthcare: From waste elimination toward ‘queue-less’ patient-focused care. *Business Horizons*, 61(1):13–22, 2018.
- [31] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.
- [32] I Sobel and G Feldman. A 3x3 isotropic gradient operator for image processing, presented at a talk at the stanford artificial project. *Pattern Classification and Scene Analysis*, pages 271–272, 3.
- [33] Jie Yang, Yun Chen, Stephen Westland, and Kaida Xiao. Predicting visual similarity between colour palettes. *Color Research & Application*, 2020.
- [34] Luwen Yu, Stephen Westland, Zhenhong Li, Qianqian Pan, Meong Jin Shin, and Seahwa Won. The role of individual colour preferences in consumer purchase decisions. *Color Research & Application*, 43(2):258–267, 2018.
- [35] Qing Zhang, Chunxia Xiao, Hanqiu Sun, and Feng Tang. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing*, 26(4):1952–1964, 2017.