A mobile compatible skin colour transfer algorithm for the human hand

by

T. Han

Supervisor: Dr. Parham Aarabi
August 2017

Abstract

In the make up, beauty, and apparel industries, there has been a growing trend to personalize images of models to take on the user's own bodily features in order to allow for users to better visualize products worn on themselves when making a purchasing decision. Specifically for this project, we have a nail polish virtual try-on mobile application that demonstrates nail polish colours on a video of a model hand of mid-toned skin colour, and our goal is to develop a software that would be able to tailor this video to each user by automatically adjusting the skin colour of the hand within each frame of the video to the user's skin colour. We developed an algorithm in C++ using OpenCV libraries and tested the effect on several hand images of various skin colours. We iterated through several different versions of the algorithm and determined that the best results are yielded by using an adjustment of the RGB colours to match the average skin colour exactly to the target image, while proportionally fading off the adjustment based on distance to the average. Our results appear realistic on moderate skin colour changes but not on extremely large skin colour changes. Future work should better address this and also test on a wider range of images and video frames, as well as optimize the algorithm for a mobile application.

Acknowledgements

The author is extremely grateful to Parham Aarabi, Brendan Heath and Shahad Abdulnour for all their help and encouragement.

Special thanks to Curtis Yu, Ann Sai
and Kevin.

Contents

List of Figures

List of Tables

List of Symbols

α The constant used to determine the strength of the dark spot correction

$\bar{C_S}$ The average RGB color of the output hand

$\bar{C_S}$ The average RGB color of the source hand

$\bar{C_T}$ The average RGB color of the target hand

$\boldsymbol{C_O}$ The vector of RGB values for the pixels in the output image

$\boldsymbol{C_S}$ The vector of RGB values for the pixels in the source image

$\boldsymbol{C_T}$ The vector of RGB values for the pixels in the target image

O The set of pixels in output image

S The set of pixels in source image

T The set of pixels in target image

U The subset of pixels used to calculate the average color of the hand in the target image

V The subset of pixels used to calculate the average color of the hand in the source image

W The subset of pixels used to calculate the average color of the hand in the output image

1 Introduction: the need for an effective mobile compatible skin recolouration algorithm

The modern beauty, cosmetics and apparel shopping experiences increasingly take place online and digitally, enhanced by *virtual try-on* applications which demonstrate products on digital models to provide previews and vicarious experiences of the products for the users [7]. Increasingly these applications can either use images of the user themselves to demonstrate products or digitally modify the models to take on the user's physical characteristics such as skin colour, facial features and body measurements [8, 9]. A study by Merle et al. shows that this increasing "perceived resemblance" between the user and the model increases the user's sense of connection to the model and is an important factor in increasing the user's perceived usefulness of the application and the positive user responses to the demonstrated products [10].

Our project is concerned with improving the perceived user-model resemblence for a particular virtual try-on mobile application, which currently demonstrates different nail polish colours on a video of a generic model hand. We would like instead to have the skin colour of the model hand be matched to each user's skin colour, so that the contrast between the nail polish colours and the user's skin can be clearly visualized. However, because preparing even a single video for virtual try-on is an extremely time intensive task, it is not feasible to manually prepare a large number of different demo videos with models of a range skin colours that would satisfy all users.

To address this challenge, we will develop an algorithm to incorporate into the application so that for each user using an instance of the application, the algorithm would quickly and automatically edit each frame of the video of the generic model hand so that the skin colour of the user is transfered to the image of the model hand. The user should only need to provide an image of their own hand as input, and a wide range of user skin colours should be supported by a single base video of a model of mid-toned skin colour. The process should be also able to run quickly on a mobile device, such that the user notices no significant time lag to see the resulting video upon inputting their own skin colour. We will discuss the requirements for our project in detail in Section 3.

Currently, we aren't aware of an existing algorithm for skin colour transfer that satisfies all our specific requirements. While there has been a large body of work done addressing transfer of colour between images in general [2, 11–14], only a smaller subset of the work addresses skin colour specifically [4–6]. All such studies address images of facial portraits rather than hands, which often means that the bulk of the study is dedicated to solutions for the colour transfer of the various complex features of the face [5]. Simple skin colour transfer is also used as a part of certain other imaging processing applications, but similarly, since the skin colour transfer is often only a small part of the whole project, the algorithms used are often relatively simple and not heavily designed for achieving accuracy to the user skin colour [8, 9]. In the related field of skin colour enhancement applications, the methods used are generally meant for small skin colour adjustments and not for making large changes to the user skin colour [15,16]. Finally, algorithms developed by most of the prior studies do not appear to be meant for use with the limited resources on a mobile device. We will discuss these previous studies of methods of skin colour transfer in detail in the following section, Section 2.2.

2 Background: existing methods of skin colour transfer

2.1 Changing and matching skin colour in Photoshop

Skin colour correction is a frequently encountered problem in photo retouching and there are a wide range of online video tutorials available documenting the methods artists use to manually adjust human skin colour in individual images using Adobe Photoshop, a widely used commercial image manipulation software. The purposes of these videos include giving the subject of an image the appearance of a tan, matching the skin tone of the subject to a desired skin tone on another individual, or matching the skin tone of a subject's face to the rest of the subject's body, which is often a slightly different colour [17–19]. Bearing in mind that techniques described by such tutorials expect artistic input from a human editor to achieve the results and are therefore not entirely applicable to the purpose of this project, it is useful to study these methods because the results achieved are usually extremely realistic and aesthetically pleasing and should be a standard that the algorithm developed for this project is designed towards. We therefore surveyed a number of these videos and summarize below the techniques of some of the most relevant tutorials.

Summary of Photoshop techniques

Shaver demonstrated how to change a person's skin colour from dark to light [1]. Shaver used levels and curves, which are tools that manipulate the RGB colour histogram of the image, to increase brightness to an extent, then performed further brightening by using a grey scale conversion to brighten the skin area of a black and white image and then using the luminosity blend mode to place the colour back into the image. We show the results achieved in Table 1.

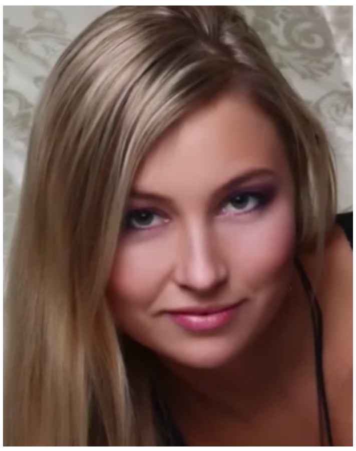
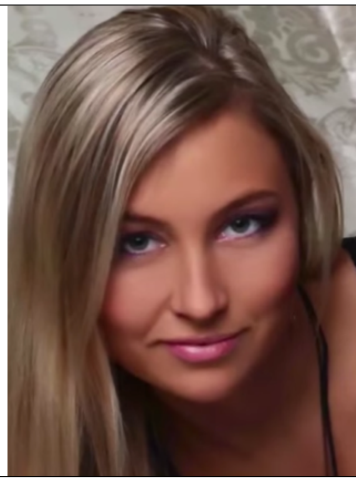
Table 1: Screen captures from Shaver's Photoshop tutorial for changing skin colour from dark to light [1].

Source	Output
	

Phlearn demonstrated an effect in the reverse direction by demonstrating a technique for giving the model the appearance of a dark tan [17]. The highlights and shadows of the image are adjusted separately by using the "blend if" function of Photoshop, which blends in an effect only if the original pixel is above or below a certain threshold of brightness.

Phlearn also demonstrated a method for matching the skin colour of body and face in an image where the two appear mismatched [19]. The author sampled a range of colours from the body and adjusted the face with the levels tool for each colour channel. We show the results achieved in Table 2.

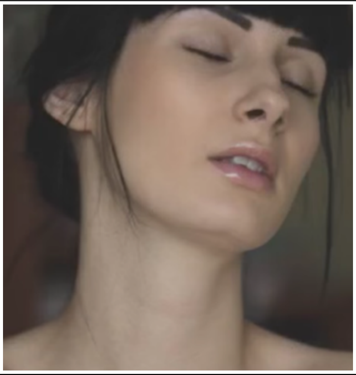
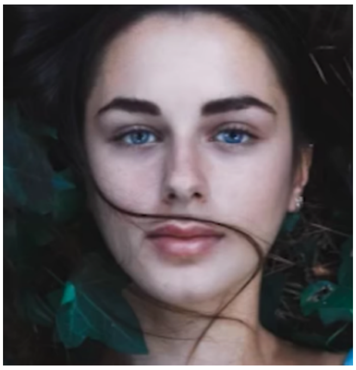
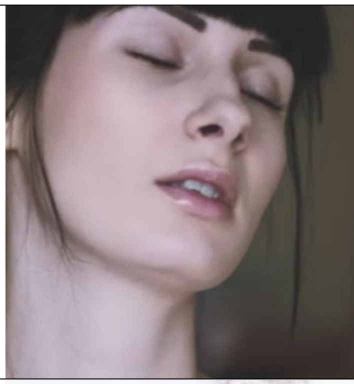
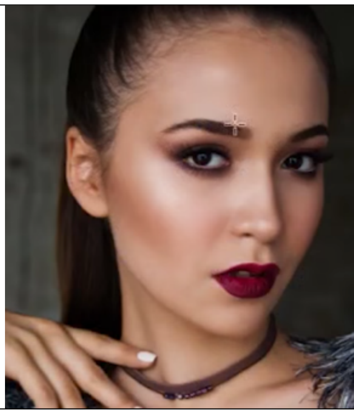
Table 2: Screen captures from Photoshop tutorial for matching the skin tones of face and body.

Source	Target	Output
		

PiXimperfect demonstrated a method for matching skin colour in one portrait to another

[18]. PiXimperfect first calculates the two average colours of the faces and uses the Photoshop curves tool to match the average colours of the original image to the target image. There must then be further adjustments by eye to change colour, brightness and contrast. Examples of the results from PiXimperfect is show in Table 3

Table 3: Screen captures from Photoshop tutorial for matching the skin tones of portraits of different people.

Source	Target	Output
		
		

Generally, for most of the techniques surveyed, levels and curves are used for small brightness adjustments [1,18,19], and often to reduce the vividness of the colour adjustments the saturation must be slightly decreased [1,19]. After all other effects are applied, the opacity of the overall effect is often reduced from 100% for a more natural appearance [1, 19].

Limitations of Photoshop techniques

The Photoshop techniques surveyed are not meant for automation; instead, they are meant to be tailored to each specific image that a human is adjusting, and there are many junctures where the specific numerical amount of an adjustment have to be judged by eye. While Photoshop has a method for automating processes to an extent using *actions* [20], the processes are meant for increasing ease of use by artists who can make additional adjustments and are familiar with the tool, rather than for use in commercial applications where the process is entirely automated.

Another limitation is that Photoshop operates at a higher level of abstraction than image processing code, which has much more control over processes that can be applied to images, and the regions on the image that processes are applied to.

Finally, some Photoshop effects may be proprietary and are, of course, limited to the platforms that support Photoshop, while a program coded with image processing libraries can be made open source and can be integrated into applications running on a variety of different platforms.

2.2 Academic work related to colour transfer and skin colour enhancement

We have also surveyed the current state of the academic work relevant to our project, which fall into four rough categories:

Colour transfer for general images. There is a large body of work on the subject of automatically transferring the "style" or colours in an example image to another image. The work in this area is focused on being effective for a wide range of colours in general images, not necessarily skin colour; however, skin colour transfer algorithms and processes often refer to this body of work and so this area is of interest to us.

Colour transfer for human skin colour. There have also been several prior studies transferring colour specifically for images wherein skin colour is prominent. These are most similar in purpose to our project and we will discuss each study in detail.

Skin colour transfer as part of other applications. We have also found several examples of practical application of skin transfer algorithms, where the practical use of usually a relatively simple skin transfer algorithm is demonstrated as part of a larger project.

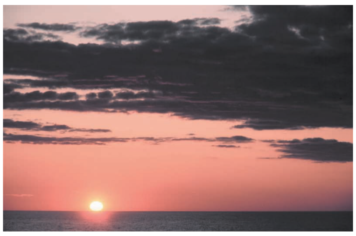
Skin colour enhancement mobile applications. Finally, there is the related field of skin tone enhancement software, where algorithms are usually intended to adjust the user skin colour towards a more pleasing tone and not to a specific target colour. We include the this field because unlike the other categories of prior work there are several studies of adjusting skin tone on a mobile device, which is part of the requirements for this project.

2.2.1 Colour transfer by example image for general images

Colour transfer refers to modifying the colours of an image to give it the desired appearance and style demonstrated by an example image, which we will refer to as the *target image*. Table 4 illustrates an example of this effect.

There have been a wide range of studies done in this area beginning with the seminal work of Reinhard et al. in 2001 [2]. The authors convert the image into $l\alpha\beta$ colour space, a colour space designed for natural scenes and based on research into human perception to reduce the correlation between each channel and remove the need to consider cross-channel effects when performing transformations on each channel. The authors then perform a simple operation to match the average and standard deviation of each channel of the original image to that of the target image. The resultant image is then converted back into rgb space.

Table 4: Example of image colour transfer using the algorithm from Reinhard et al. All images from [2]

Source	Target	Output
		

In a later study, Pitié et al. developed a method for entirely transferring the exact statistical distribution of the colours of the target image to the original image [11], and later improved on the technique with the motivation of automating film grading, or the process of enhancing frames in films to ensure consistency of colour and “feel” [3]. We show an example the effects they achieve in Table 5

Table 5: Example of film grading based on an example image using the algorithm from Pitié et al. All images from [3]

Source	Target	Output
		
		

More recently, Bonnel et al. conducted a further study on colour transfer for film grading considering both spacial and temporal information [21] and Chang et al. created a tool for user editing of an image based on a automatically generated colour palette [13].

While these techniques are interesting possibilities to try when transferring human skin colour, because these prior studies are all concerned with different problems that can arise with general images but not specifically for human skin colour, studies that specifically relate to human skin colour demonstrate that the general colour transfer techniques can be improved upon.

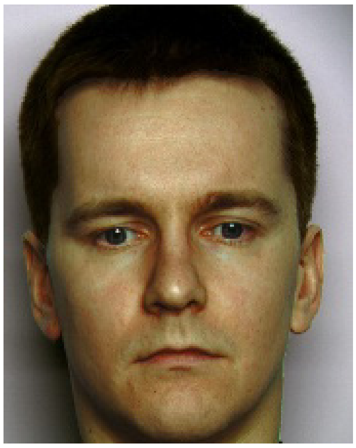
2.2.2 Colour transfer by example for images with human skin

There are fewer studies done specifically on the transfer of human skin colour, but there are several of great interest to us.

Seo et al. has a purpose most similar to the purpose of this project, which is to find a method of transferring human skin colours as realistically and accurately as possible

[4]. In their study, authors demonstrate results that improve in realistic appearance compared to the Reinhard's algorithm. To achieve this, authors model the skin colour as an elongated distribution around a line referred to as the *principle line* in *rgb* space. To perform the colour transfer, the authors transform the distribution of the original image such that the principle line aligns with that of the target image. The authors then break the colour values into bins along the principle lines and also match the distribution of each bin. Table 6 demonstrates the output of this method.

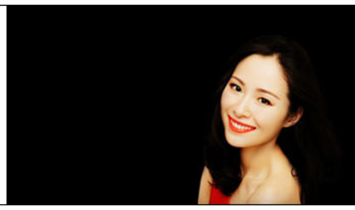
Table 6: Example of image colour transfer using the algorithm from Seo et al. All images from [4]

Source	Target	Output
		

However, as far as the other requirements of this project is concerned, it is not clear how fast this algorithm can run, particularly on a mobile device, nor the range of colours that the algorithm can realistically transform between, and it is in these areas that our project will attempt to improve upon.

Yang et al. performed a more recent study of colour transfer for human portraits [5] - Table 7 demonstrates their results. However, as Table 7 shows, this study focuses on the effect on the whole portrait, and places emphasis on transferring colour for different features of the human face. The actual algorithm the authors use to transfer skin colour is actually Reinhard's algorithm. This method also ranks the preferred target image for similarity to the original image before performing the colour transfer, which differs from our project where a key issue is that we have no control over the target image that the user will provide us and must be able to transfer to a wide range of colours.

Table 7: Example of image colour transfer using the algorithm from Yang et al. All images from [5]

Source	Target	Output
		

Yin et al. performed a study on the transfer of skin colour between races in order to aid a psychological study [6]. The authors use a process of first global adjustment of the two images to match the average colours, and then using the pixels from the target image with a colour most resembling the colour in the original image to entirely replace that pixel in the original image. Table 8 shows an example of the results using this technique.

Table 8: Example of image colour transfer using the algorithm from Yin et al. All images from [6]

Source	Target	Output
		

We include this study because it is one of the few that explicitly tries to transfer colour between large difference of skin colour, however, we feel it has many flaws with respect to our other requirements. Particularly in the case of large skin colour changes shown above, the results are not realistic and show bright spots of saturation. Furthermore, the authors describe their algorithm as having $O(n^2)$ time, which likely means that this algorithm will perform too slowly for our purposes.

2.2.3 Skin colour transfer as part of other applications

Several applications performing different functions make use of skin colour transfer.

Shilkrot et al. published a study on transferring identity of the user on to a model image wearing garments the user may desire to purchase to create the virtual experience of the user trying on the garment [8]. The purpose of this article is fundamentally related to the purpose of the application that is this project's goal to support, and so this article is of great interest to us.

In fact, as part of the identity transfer, this article performs skin colour transfer on the model image to take on the skin colour of the user. Shilkrot uses a Gaussian Mixed Model for transfer skin colours and seems to use it for a relatively wide range of colour differences.

The difference between Shilkrot's study and ours is that skin is only a small part of their final image, and the skin transfer process is only a small part of their study, which also places emphasis on the transfer of the user's head and the reshaping of the model's body proportions. In our case, the hand will be the only object of interest in our inputs and outputs. While we need not devote our efforts to any aspect but the skin colour change, any flaws in the colour transfer causing unrealistic results will be much more noticeable.

Another interesting case of skin colour matching is the work of Bitouk et al to create a face swapping software that seamlessly changes faces in photos to stock photo faces [22]. Since the skin colour of stock photo face does not exactly match the rest of the skin colour in the original photo, the authors adjust the lighting and skin colour until they do match. However, the author specifically state large skin colour changes cannot be made, and to support a wide range of skin colours, the authors rely on having a large library of stock photos of every lighting position and skin colour. On the other hand, in our case, we are motivated by that fact that it is difficult to prepare videos for the full range of user skin colours.

Another application we've found is the work of Baba et al. to develop a software that edits portraits in yearbook photos to all have a uniform skin colour given an example skin colour image [23]. The algorithm that the authors use to achieve this uses Piti's colour grading algorithm and guided image filtering. However, the goal of the project in terms of skin colour appears to be to have skin colours close to the target image but not necessarily exactly the same; the authors are focusing on the overall appearance of the set of portraits rather than the accuracy of the skin colour transfer for each individual image. On the other hand, the goal of our project is to ensure that the transformed skin colour of the model is as accurate as possible to the user's skin colour.

2.2.4 Skin colour enhancement mobile applications

For the most part the studies we have found are not meant to be run on a mobile platform and there are few studies specifically pertaining to human skin colour. There are, however, many skin colour enhancement applications that modify human skin colour and several studies that perform this on mobile devices.

For example, Lee et al. enhances skin colour of users to a *preferred skin colour*, or a skin colour close to the user's original skin colour that a member of the user's ethnic group would find the most pleasing, in addition to changing the background of the user's video on a mobile phone [16]. They perform the skin colour transformation in HSV or hue, saturation and value colour space, simply adding the difference between the average hue and saturation values between the image and the target colour, damping the offset they add by a multiplicative factor for more realistic appearing results.

The difference in purpose between such studies and ours is that we have a specific target colour that could be different from the aesthetically pleasing colour that skin enhancement is aiming for, and our requirements for accuracy to the target colour is more stringent.

2.3 Summary of differences between prior studies and this project

In summary, we find that the prior work can satisfy some but all of our requirements for this project. Most of the sophisticated colour transfer work done is for general images and not meant to accurately and realistically transfer human skin colour. On the other hand, much of the work done related to correcting human skin colour is meant to improve the subjective appearance of human skin, and not meant to accurately change the human skin colour to a specific shade. For studies specifically addressing the recolouring of human skin, many of the techniques are not meant to achieve skin colour transfer between a large difference in skin colour. The studies we found that did perform transforms across a large range often do not have sufficiently convincing and artifact-free results. Finally, most of the results we surveyed do not perform the transformation at a speed fast enough for a mobile application, or else have not given the speed of the algorithm significant consideration. We will attempt to address all these areas in our project.

3 The constraints and requirements for our skin colour transfer algorithm

Our project is intended to manipulate image frames in a video of a model hand demonstrating nail polish product so that the model hand takes on the user's skin colour. The images we must process will mostly consist of the back of a single hand shown prominently in the image. We expect image sizes the algorithm should be able to handle to be relatively large, at least approximately 800 by 800 pixels in size. We show an example of the desired output of our algorithm in Table 9.

Table 9: Example of our desired output given a source (the model) and a target image (from the user)

Source	Target	Output
		

To narrow the scope of our project, we will not include skin detection as part of this project and assume that our algorithm is already given a mask of the skin areas of all the images. We will focus solely on the transfer of the hand skin colour.

Based on our goals and the nature of our project, we list below several constraints and design paradigms for our algorithm:

Compatible with mobile device: Our algorithm is ultimately intended to support an application on a mobile device, so we must ensure that our code is easily portable to mobile platforms and that the algorithm we develop can operate quickly with the limited resources of a mobile device so that the user will be able to see near-instant results.

Fully automatic: Since the goal of our project is for a commercial user to be able to easily adjust the model image to have his or her own skin colour, our algorithm cannot rely on any user input to perform the image editing and should be only require an image containing the user's own hand as the target image for performing the colour transfer on the model.

Realistic skin colour transfer: Since the results of the algorithm are meant to invoke for the user the impression that the user's own hand is wearing a cosmetic product, and since we can expect that the user would be sensitive to both an image of their own hand and the subtle effect of a particular cosmetic product, our final images must look as realistic as possible to avoid displeasing, uncanny valley effects. Furthermore, the images we process will be large and feature the skin on the back of a hand very prominently, so we can expect that unrealistic flaws in our result will be very noticeable the user.

Accurate skin colour transfer: Since the purpose of the nail polish try-on application is to demonstrate to the user how a particular shade of nail polish will appear on the user's own hand, we must ensure that the results of the algorithm, more than looking pleasing to the user, actually matches the skin colour sample provided by the user so as to give the user a realistic preview of the product.

Wide range of colour transfer: Since the project is motivated by the goal to support users with a wide range of skin colours while avoiding the necessity of manually creating a large number of nail polish try-on videos of different skin colours, our algorithm needs to be able to transfer the skin colour of a mid-toned hand to as wide a range of skin colour as possible.

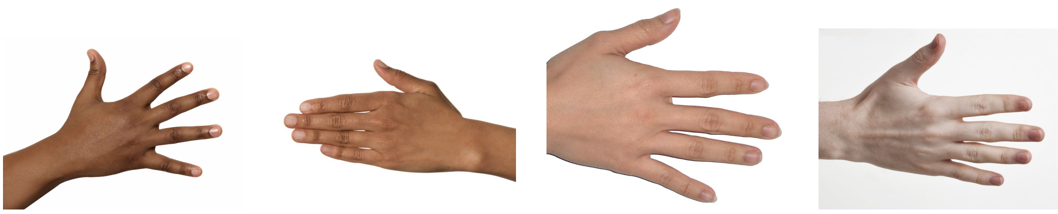
4 Hand Recolouration Algorithms

We chose to write our algorithms in C++ using OpenCV libraries. Since C++ and OpenCV are available on both Android and iOS platforms as well, the code should be relatively easy to port to those platforms. To allow for faster testing of our code, we develop the code on OS X.

Eclipse is used to compile each iteration of the algorithm into a debug-mode executable program named Recolor. For ease of testing, as the algorithm is modified, we add more functionality to the Recolor program and retain the ability to use previous versions of the algorithm. We use a custom Python script to run new versions of Recolor from the terminal to test it. All of the relevant code and its versions are hosted on a git repository at github.com/tiantianhan/recolor

Recolor takes as input a hand image, a mask instructing it where to find the average skin colour of the hand, and a desired target skin colour. (Other flags and inputs are also used for testing purposes, see the Github repository README file for a full description of the usage.) Recolor then outputs the processed image where the skin tone is adjusted to the target colour.

We iterated from simple to more complex algorithms, at each step testing the algorithm and evaluating the results. We tested progressive iterations on a set of hand images with varying skin tones. The images are are shown in Figure 1.



(a) (b) (c) (d)

Figure 1: Different hand images used for testing

For each test, we called the Recolor program to transform the image of one hand to have the skin tone of the hand in another image, then visually compared the processed image to the image of the target hand. We performed the process on all possible combinations of our test images, paying particular attention to the extreme cases, such as transforming from Figure 1a to Figure 1d and vice versa, as well as cases that start with a hand with mid-tone skin such as in Figure 1b (as this is the most likely use case for applications that change the image of a model's hand to match a range of skin tones). We evaluated the resulting images subjectively, based on whether the processed hand looks believably like a hand naturally of that skin tone, and noted any flaws that we then attempted to correct with the next iteration of the algorithm.

In the following subsections we summarize the results of each algorithm and our evaluation of the results.

4.1 Naive approach: simple RGB colour addition

To begin, we performed a simple addition of a value to each of the RGB channels of the hand, such that the average colour of the hand in the processed image is equal to the average colour of the hand in the target image.

Algorithm 1 Simple addition to RGB channels

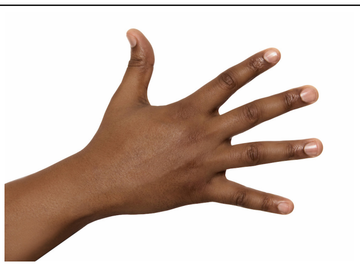
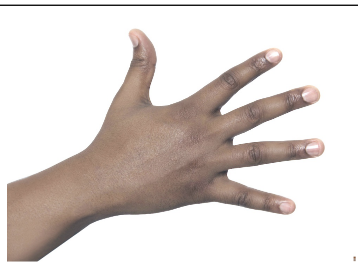
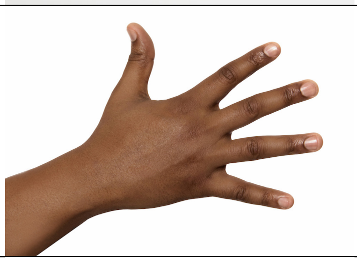
Let T be the set of pixels in the target image, and $U \subseteq T$ be the set of pixels used to calculate the average color of the hand in the target image.
Let S be the set of pixels in the source image, and $V \subseteq S$ be the set of pixels used to calculate the average color of the hand in the source image.
Let $\boldsymbol{C_T}$ be the vector of RGB values for the pixels in the target image.
Let $\boldsymbol{C_S}$ be the vector of RGB values for the pixels in the source image.
Let $\boldsymbol{C_O}$ be the vector of RGB values for the pixels in the output image.

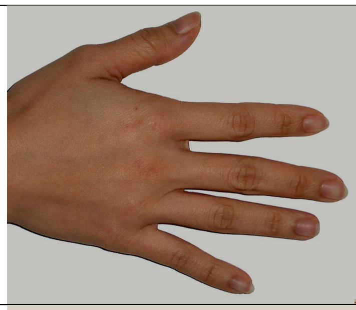
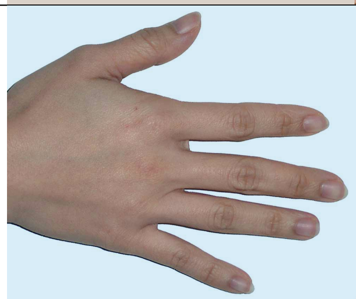
We first calculate the average colour of the source and target images
$\bar{C_S} \leftarrow \textsc{Mean}(\boldsymbol{C_S}(V))$
$\bar{C_T} \leftarrow \textsc{Mean}(\boldsymbol{C_T}(U))$

We then perform a simple addition to each pixel in S
for each pixel $i \in S$ **do**
  $\boldsymbol{C_O}(i) \leftarrow \boldsymbol{C_S}(i) + \left(\bar{C_T} - \bar{C_S}\right)$
end for

In Table 10 we show the results for colour transfers between all possible combinations of our test images.

Table 10: Test results of Algorithm 1, simple addition function

No.	Source	Target	Output
10.1			
10.2			
10.3			
10.4			
10.5			
10.6			

10.7			
10.8			
10.9			
10.10			
10.11			
10.12			

Our results show that images of darker skin tones and smaller changes from the original skin tone to the target colour to begin with (Row 10.4) tend to have better results than images with large changes (Rows 10.2, 10.6). In the case of changes towards brighter colours, this is because large changes force bright points in the original image to be truncated at white, and also causes dark regions on the image, such as shadows and grooves, to become significantly brighter and less close to true black, giving the image a "high-key" appearance (Row 10.2 and 10.5).

In addition, we noted that at this stage the transformation from a dark coloured hand to a very pale hand, or even from a mid-toned hand to a pale hand and vice versa is especially unconvincing. (Row 10.6, also see 10.3 and 10.10)

4.2 Proportional adjustment relative to average colour

To correct for the effect of the bright spots in the image being over bright and the high-key appearance resulting from all the shadows being brightened, we used an algorithm that maps the *black and white points*, the *RGB* colours of $(0, 0, 0)$ and $(255, 255, 255)$ respectively, of the image to the same value, and adjusts the colours inbetween proportionally so that colours closer to the average are adjusted more, and the source average colour is matched to the target average colour.

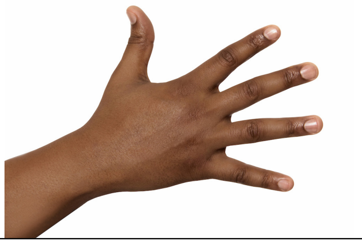
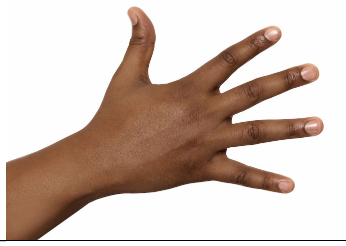
Algorithm 2 Proportional adjustment relative to average color

We calculate the average colours of the source and target images
$\bar{C_S} \leftarrow \textsc{Mean}(\boldsymbol{C_S}(V))$
$\bar{C_T} \leftarrow \textsc{Mean}(\boldsymbol{C_T}(U))$

We adjust pixels with colours below the average of the source according to their distance to black and the pixels above average according to their distance to white
for each pixel $i \in S$ **do**
  if $\boldsymbol{C_S}(i) \leq \bar{C_S}$ **then**
$$\boldsymbol{C_O}(i) \leftarrow \left(\frac{\bar{C_T}}{\bar{C_S}}\right) \boldsymbol{C_S}(i)$$
  else
$$\boldsymbol{C_O}(i) \leftarrow 255 - \left(\frac{255 - \bar{C_T}}{255 - \bar{C_S}}\right)(255 - \boldsymbol{C_S}(i))$$
  end if
end for

In Table 11 we show the results for colour transfers between all possible combinations of our test images.

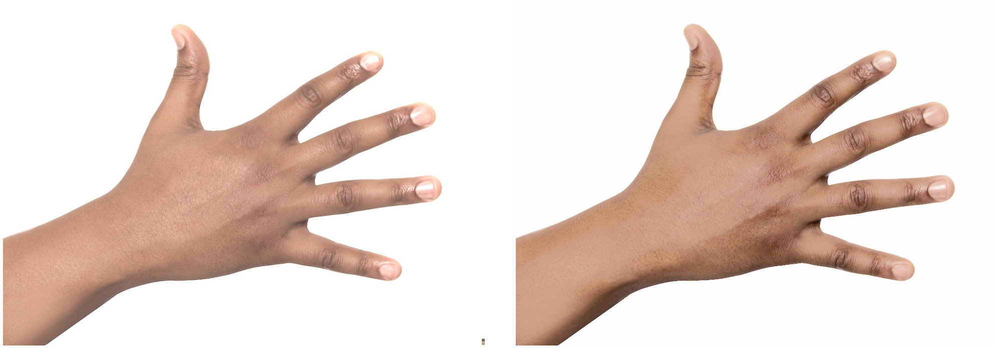
Table 11: Test results of brightening proportionally based on distance of color to the average.

No.	Source	Target	Output
11.1			
11.2			
11.3			
11.4			
11.5			

11.6			
11.7			
11.8			
11.9			
11.10			
11.11			

11.12			

This method improved the appearance of cases with over-bright spots or "high-key" appearance issues, as Figure 2 shows:



(a) Simple addition algorithm (Algorithm 1) result

(b) Proportional adjustment algorithm (Algorithm 2) result

Figure 2: Comparison of Algorithms 1 and 2 results for transforming a dark hand (Figure 1a) to a light hand (Figure 1c).

We noted however, that this method noticeably does not correct for, and even exacerbates slightly relative to the simple addition algorithm the dark spots at the joints and creases of a hand of darker skin tone when it is transformed to a lighter skin tone (Row 11.5). We note that results for colour transfer between very different colours, such as in Row 11.10, have similar issues to the results of Algorithm 1 and are lacking in realism.

4.3 Correcting for dark spots

We attempted to correct the dark spot issue by significantly reducing the absolute difference between dark pixels and the average colour, ensuring that the dark spots would instead have colours close to the average. We perform this correction on the output of the proportional adjustment algorithm.

Algorithm 3 Dark spot correction

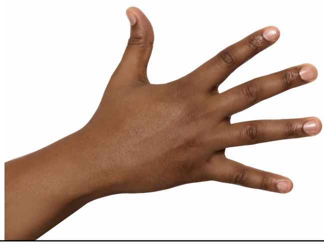
Let α be a constant, $\alpha > 1$. We expect that the larger the value of alpha, the stronger the effect of the reduced dark spot.
Let O be the set of pixels in the output image, and $W \subseteq O$ be the set of pixels used to calculate the average colour of the output image. W corresponds to the region $V \subseteq S$, and $|W| = |V|$

We calculate the average skin colour of the output of Algorithm 2
$\bar{C_O} \leftarrow \textsc{Mean}(\boldsymbol{C_O}(W))$

For all pixels below the average colour we limit their distance to the average
for each pixel $i \in O$ **do**
  if $\boldsymbol{C_O}(i) \leq \bar{C_O}$ **then**
    $\boldsymbol{C_O}(i) \leftarrow \bar{C_O} - \dfrac{(\bar{C_O} - \boldsymbol{C_O}(i))}{\alpha}$
  end if
end for

In Table 12 we show the results for colour transfers between all possible combinations of our test images for $\alpha = 1.1$. The results for some of the other α values we tried are shown in Appendices A and B.

Table 12: Test results of proportional brightening with correction for dark spots, $\alpha = 1.1$

No.	Source	Target	Output
12.1			
12.2			

12.3			
12.4			
12.5			
12.6			
12.7			
12.8			

12.9			
12.10			
12.11			
12.12			

As shown in Figure 3, the dark spots and creases noted in Section 4.2 are reduced.

(a) Proportional adjustment algorithm (Algorithm 2) result

(b) Proportional adjustment algorithm with correction (Algorithm 3) result

Figure 3: Comparison of Algorithms 2 and 3 results for transforming a mid-toned hand (Figure 1b) to a light hand (Figure 1c).

We tried this effect for a range of α and found that $\alpha = 1.1$ gives an acceptably realistic result. A larger α (such as the case for $\alpha = 5$ shown in Appendix B) would further reduce the dark spots on the skin but may begin to strongly brighten the shadows of the image, resulting in an unrealistic effect.

Up to the current iteration the more extreme changes of colour, such as from Figure 1a to Figure 1d and vice versa are especially unrealistic. Part of the reason is that the shadows, most prominent in Figure 1d is causing the calculated average colour of the entire hand to be of lower luminosity than it should be.

4.4 Ignoring shadows in calculating average colour

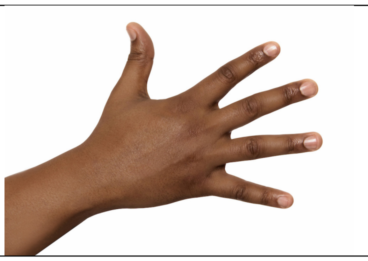
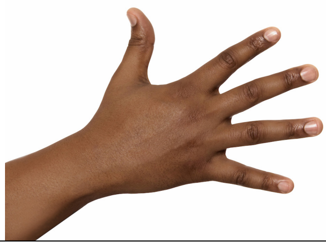
We noticed that in the case where the target image has a pale hand and relatively dark shadows such as in Figure 1c, the average colour calculated for the target hand is too dark, causing the skin colour of the results to appear darker than the skin colour of the target. We correct this by calculating the average skin colour of the target hand with only a percentage of the brightest pixels in the original region of interest used to calculate the average.

Algorithm 4 Calculation of average skin colour with brightest pixels

Let p be a constant, $0 < p < 100$
Let $U_p \subseteq U \subseteq T$ be the brightest pth percentile of pixels in U
$\bar{C_T} \leftarrow \textsc{Mean}(\boldsymbol{C_T}(U_p))$

We generated the resultant images using several different values for p, the percentile of brightest pixels, and determined that the $p = 10$ gave the best results, as shown in Table 13. In Appendices C and D we show the results for some other percentiles.

Table 13: Test results of proportional brightening with correction for dark spots, $\alpha = 1.1$, using brightest 10 percent of pixels to calculate average colour

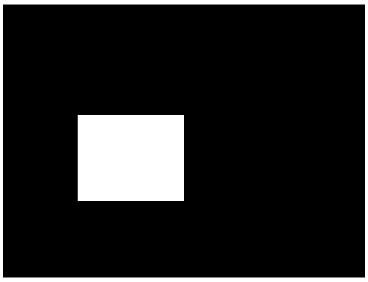
No.	Source	Target	Output
13.1			
13.2			
13.3			
13.4			

13.5			
13.6			
13.7			
13.8			
13.9			
13.10			

13.11			
13.12			

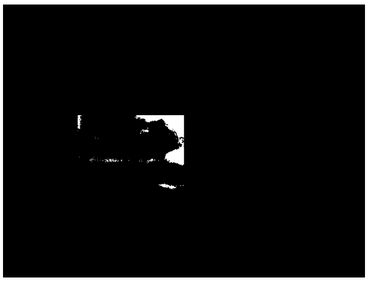
Figure 4 demonstrates the new regions used to calculate the average skin colour. We can see that the areas with shadows are effectively discarded, and the average colour calculated is significantly lighter and visually more accurate to the skin colour of the target hand.

(a) Input hand image with significant shadows



(b) Mask used to calculate color in Algorithm 3

(c) Average color calculated with Algorithm 3

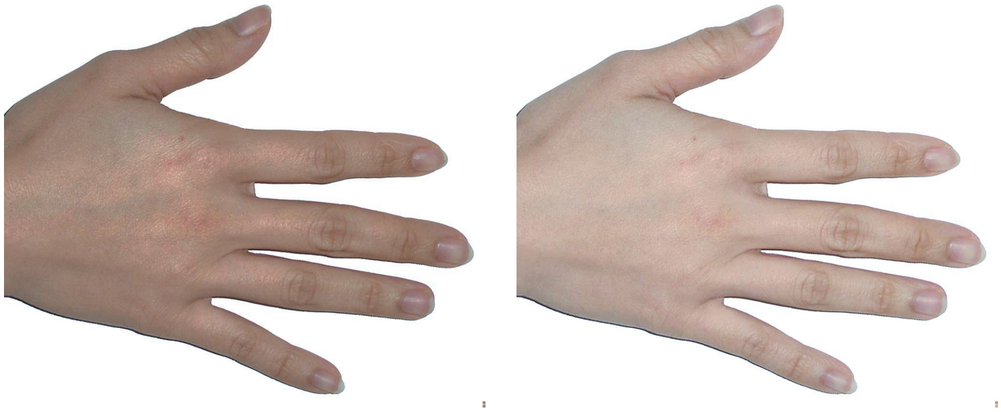


(d) Mask used to calculate color in Algorithm 4

(e) Average color calculated with Algorithm 4

Figure 4: The average color calculation process in Algorithm 4 compared to that in the previous version, Algorithm 3

As a result of the more accurate target colour, the resulting output image is more accurate to the colour of the hand in the target image, as show in Figure 5



(a) Output obtained using all pixels for target average (Algorithm 3)

(b) Output obtained using brightest pixels for target average (Algorithm 4)

Figure 5: Comparison of Algorithms 3 and 4 results for transforming a light hand (Figure 1c) to a pale hand (Figure 1d).

We note that, overall, the more moderate skin colour changes which use mid-toned and light hands as the source image in Rows 13.4 to 13.9 now show relatively realistic and accurate results. However, for other tests with more extreme colour changes, such as in Row 13.3 and 13.10, the colour change is still not convincing.

4.5 Summary and evaluation of the complete algorithm

The complete algorithm (comprised of Algorithms 2, 3 and 4) is summarized in the flow chart in Figure 6.

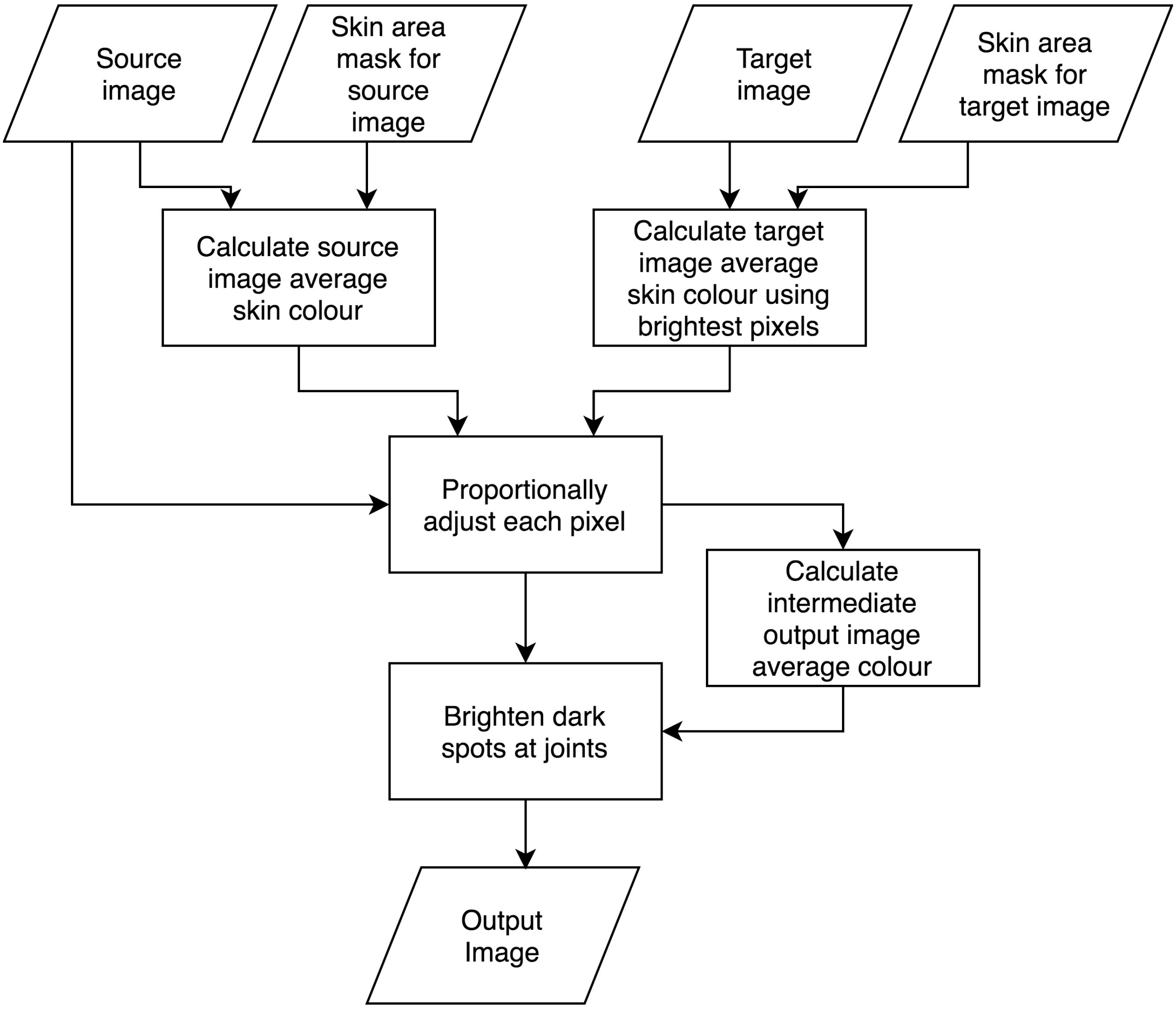


Figure 6: Summary of complete algorithm for hand colour transfer

In reference to our original goals and constraints listed in Section 3, we have made the most progress in the area of accurate and realistic transfer of skin colour from mid-toned and light coloured hands to other colours. However, the range of colours that we can realistic transfer between remains limited. For future work, we should further test our algorithm on a larger set of hand images to determine whether there are any other issues. Finally, we have yet to profile and optimize the code for performance on a mobile devices; however, since we have written the code using a language and libraries that should be easy to port to a mobile platform, we believe that optimizing and porting the code should be a very feasible task for future work.

5 Conclusion and Future Work

We have determined that it is feasible to produce a software that satisfies our requirements for fully automatic skin transfer for a range of skin colours starting from a hand of mid-toned or light skin colour. However, for a realistic appearance and for the final result to be accurate to the skin colour of the target hand, the original hand cannot have very significant skin colour differences with the target.

As a next step, we would like to improve our algorithm so that the results for transforming between very large colour differences are more convincing. We would also like to optimize our results to have sufficient speed to operate on a mobile platform.

References

[1] J. Shaver. (2014) How to change a person's skin color from dark to light in photoshop. [Online]. Available: http://www.designpanoply.com/blog/how-to-change-a-persons-skin-color-from-dark-to-light-in-photoshop

[2] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics Applications*, vol. 21, no. 5, pp. 34–41, 2001.

[3] F. Pitie, A. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123 – 37, 2007.

[4] K. C. Seo, G. Shin, and S. W. Lee, "Interactive transfer of human facial color," in *Advances in Multimedia Information Processing - PCM 2005: 6th Pacific Rim Conference on Multimedia, Jeju Island, Korea, November 13-16, 2005, Proceedings, Part II*, Y.-S. Ho and H.-J. Kim, Eds. Berlin, Germany: Springer, 2005, pp. 993–1003.

[5] Y. Yang, H. Zhao, L. You, R. Tu, X. Wu, and X. Jin, "Semantic portrait color transfer with internet images," *Multimed. Tools Appl. (Germany)*, vol. 76, no. 1, pp. 523 – 41, 2017.

[6] L. Yin, J. Jia, and J. Morrissey, "Towards race-related face identification: Research on skin color transfer," in *Proceedings - Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Republic of Korea, 2004, pp. 362 – 368.

[7] T. Zhang, L. Cao, and W. Y. Wang, "The impact of virtual try-on image interaction technology on online shoppers' purchase decision," in *Proceedings of the 8th International Conference on E-Education, E-Business, E-Management and E-Learning*, ser. IC4E '17. New York, NY, USA: ACM, 2017, pp. 6–10.

[8] R. Shilkrot, D. Cohen-Or, A. Shamir, and L. Liu, "Garment personalization via identity transfer," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 62–72, July 2013.

[9] L. Jun, Z. Mingmin, and P. Zhigeng, "Virtual try-on by replacing the person in image," *J. Comput. Aided Des. Comput. Graph. (China)*, vol. 27, no. 9, pp. 1694 – 700, Sept. 2015.

[10] A. Merle, S. Senecal, and A. St-Onge, "Whether and how virtual try-on influences consumer responses to an apparel web site," *International Journal of Electronic Commerce*, vol. 16, no. 3, pp. 41–64, 2012.

[11] F. Pitie, A. C. Kokaram, and R. Dahyot, "N-dimensional probability density function transfer and its application to colour transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1434 – 1439.

[12] X. Chen, D. Zou, J. Li, X. Cao, Q. Zhao, and H. Zhang, "Sparse dictionary learning for edit propagation of high-resolution images," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, 2014, pp. 2854 – 61.

[13] H. Chang, O. Fried, Y. Liu, S. DiVerdi, and A. Finkelstein, "Palette-based photo recoloring," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 139:1–139:11, Jul. 2015.

[14] Q. Zhang, C. Xiao, H. Sun, and F. Tang, "Palette-based image recoloring using color decomposition optimization," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1952 – 1964, 2017.

[15] H. Aradhye, G. D. Toderici, and J. Yagnik, "Adaptive, selective, automatic tonal enhancement of faces," in *Proceedings of the 17th ACM International Conference on Multimedia*, ser. 2009. New York, NY, USA: ACM, 2009, pp. 677–680.

[16] J.-H. Lee, J. Heo, and Y.-S. Ho, "Skin tone enhancement and background change for mobile phones," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2719 – 2726, 2010.

[17] Phlearn. (2016) How to get a tan in photoshop. [Online]. Available: https://www.youtube.com/watch?v=Har8CSjei2k

[18] PiXimperfect. (2017) How to match skin tone in photoshop cc 2017. [Online]. Available: https://www.youtube.com/watch?v=zEeknlHydfE

[19] Phlearn. (2014) How to match skin tones in photoshop. [Online]. Available: https://www.youtube.com/watch?v=5-DE49Muc0g

[20] M. Woloszynowicz. (2014) How to easily correct colors and match tones in photoshop. [Online]. Available: http://www.vibrantshot.com/how-to-easily-correct-colors-and-match-tones-in-photoshop/

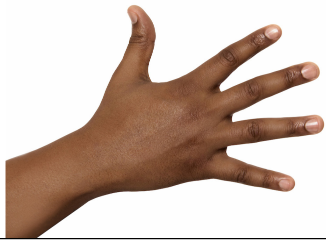
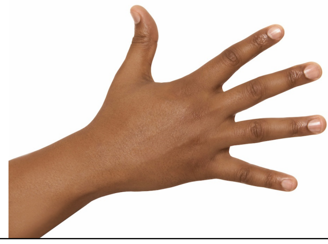
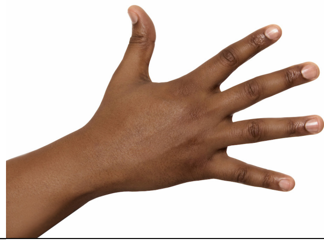
[21] N. Bonneel, K. Sunkavalli, S. Paris, and H. Pfister, "Example-based video color grading," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 39:1–39:12, Jul. 2013.

[22] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. Nayar, "Face swapping: Automatically replacing faces in photographs," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 39 (8 pp.) –, Aug. 2008.

[23] T. Baba, M. Okuda, P. Perrotin, T. Yusuke, and K. Shirai, "An automatic yearbook style photo generation method using color grading and guide image filtering based facial skin color correction," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). Proceedings*, Piscataway, NJ, USA, 2015, pp. 504 – 8.

A Results for proportional adjustment with dark-spot correction, $\alpha = 1.5$

Table 14: Test results of proportional brightening with correction for dark spots, $\alpha = 1.5$

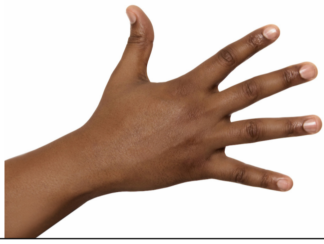
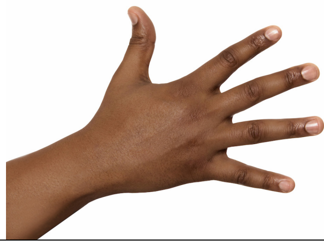
No.	Source	Target	Output
14.1			
14.2			
14.3			
14.4			
14.5			

14.6			
14.7			
14.8			
14.9			
14.10			
14.11			

14.12			

B Results for proportional adjustment with dark-spot correction, $\alpha = 5$

Table 15: Test results of proportional adjusting with correction for dark spots, $\alpha = 5$

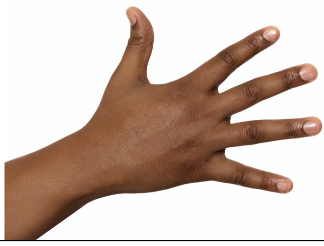
No.	Source	Target	Output
15.1			
15.2			
15.3			
15.4			

15.5			
15.6			
15.7			
15.8			
15.9			
15.10			

15.11			
15.12			

C Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 5th percentile bright pixels

Table 16: Test results of proportional adjusting with correction for dark spots, $\alpha = 1.1$, using brightest 5 percent of pixels for calculating target colour

No.	Source	Target	Output
16.1			
16.2			

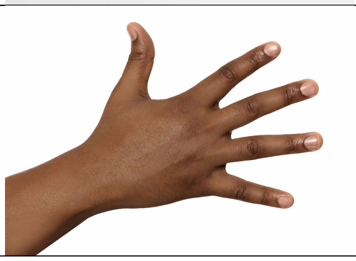
16.3			
16.4			
16.5			
16.6			
16.7			
16.8			

16.9			
16.10			
16.11			
16.12			

D Complete results for proportional adjustment with darkspot correction, $\alpha = 1.1$, calculating target average color with 25th percentile bright pixels

Table 17: Test results of proportional adjusting with correction for dark spots, $\alpha = 1.1$, using brightest 25 percent of pixels for calculating target colour

No.	Source	Target	Output

17.1			
17.2			
17.3			
17.4			
17.5			
17.6			

17.7			
17.8			
17.9			
17.10			
17.11			
17.12