Automated Restyling of Human Portrait Based on Facial Expression Recognition and 3D Reconstruction

Cheng-Han(Dennis) Wu[1] and Gordon Wetzstein[*]
Department of Electrical Engineering, Stanford University
350 Serra Mall, Stanford, CA 94305, USA
[1]chw0208@stanford.edu
[*]Project adviser

Abstract

This project demostrated an innovative automatically restyling system that turns a plane human portrait to one with effects that correspond to his/her facial expression. By training recognition model using convolutional neural network and machine learning algorithm, the system is able to detect emotion of the person in a picture. Based on the emotion, the system modifies the photo in a way that highlight the user's feeling. This restyling was done using a Kinect camera module collecting color and depth information of the character. Utilizing techniques such as graphical relighting and background modulation, the system turned an ordinary human portrait into a dramatic photo with cinematic effects.

1. Introduction

Inspired by the film Inside Out, I was curious about how image processing in combination with artificial intelligence can be incorporated with movie, video streaming, or photography industry. As of today, large amounts of video editing was done through post-editing with intense and costly human labor work. Based on the fact that scene rendering of video or photos heavily depends on the actors' or models' expression, the project aim to perform post-rendering based on auto-perception of character emotions. Through a Kinect camera, color and depth data were captured, calibrated, and rendered for the final special effect. The report will first give an overview on how the system is laid out, then dive into individual components of hardware calibration and interface, neural network feature extraction, graphical rendering based on 3D data, and the final results and discussion.

2. Related work

2.1. Depth Based Visual Effect

Visual effect based on depth information will give a more realistic rendering of characters or scene objects with respect to the background lighting and texture. The recent research of Matthias [1] demonstrated a way of capturing synchronized depth frames to perform character relighting of a movie clip. Depth information can also be used as masks to clearly distinguish targets in the scene specified by the range of depth.

2.2. Facial Relighting

Research by Yang et. al [3] demonstrated realistic relighting based on image morphing of facial images with a 3D facial model. The MRF-based method mentioned in the research was effective in facial relighting, delighting, and correction in extreme lighting condition.

2.3. RGB-D Registration

In the researches by Lembit [5]and Smisek [6] described clearly the intrinsic and extrinsic characteristics of Kinect cameras. From the parameters given by the Kinect module, image rectification can be performed based on intrinsic camera matrices and image registration can be performed by extrinsic parameters of translation and rotation.

2.3.1 Expression Recognition

Pattern recognition has long been a popular and growing field in computer vision. Current expression researches have shown dramatic improvements in precision and efficiency. Using convolution neural network, Yu [8] and Fasel [7] demonstrated success in higher precision expression recognition. Due to the limit of dataset quantity as well as inter-class variation of expression, high precision recognition is till an active research goal in this field.

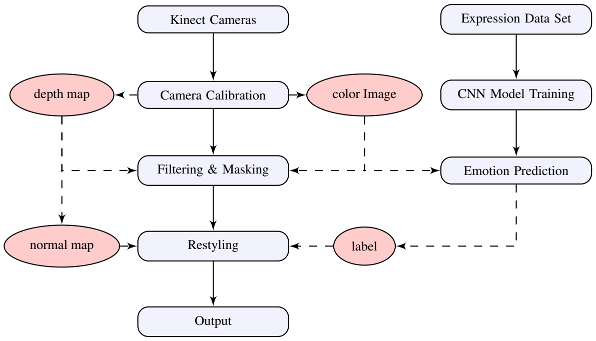
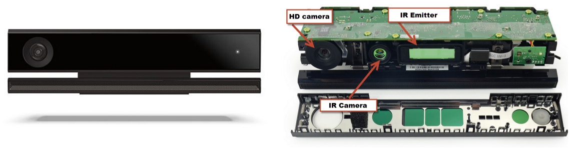


Figure 1: System architecture



(a)

(b)

Figure 2: Neural Net Layers. (a) A Kinect for Windows V2 (b) Camera geometry of the kinect [6]

3. Project overview

The project can be divided into two parts of image processing and machine learning using convolutional neural network, on which image processing is more heavily resided. The machine learning and convolutional neural network part of this project is jointly developed by Hsin Chen(hsinc@stanford.edu) and the author, and the rest of the project is done solely by the author. An overview of the project layout is depicted in Fig. 1.

4. Implementation

4.1. Kinect

This project uses a Kinect for Windows® V2 module. Kinect consists of an infrared(IR) camera and a color (RGB) camera. One emitter project patters to be picked up by the IR camera are used as a stereo pair to triangulate points in 3D space. Fig. 2 shows the outside and camera geometry for a Kinect model. The IR camera has a resolution of 512 × 424 pixels the RGB camera has a resolution of 1920 × 1080 pixels. The field of view is 70 × 60 degrees while framerate rates at 30 frames per second with operative measuring range from 0.5 to $4.5m$.

4.1.1 Hardware Interface

The computational platform of choice for the project is an Apple Macbook Pro with OSX Sierra, which does not officially support Kinect. Furthermore, this project uses a V2 module, which is newer but is not supported by the popular libfreenect[2] module's MATLAB interface. In order for the system to work out the interface, I first extracted images using libfreenect2 and libusb and at the same time, using OpenCV to perform pre-processing calculations.

4.1.2 Camera Calibration and Registration

In order to perform pixel matching between RGB and depth camera, camera calibration has to be performed first before stacking the two images together. For this project, the RGB image is taken as the reference and project the depth camera images onto the RGB camera. First is the undistortion for the depth image using intrinsic camera matrix and distortion coefficients of the depth camera, then do reprojection with respect to the color camera.
Camera matrices:

$$K_{IR} = \begin{bmatrix} f_x = 1051.79 & 0 & c_x = 981.68 \\ 0 & f_y = 1048.55 & c_y = 544.68 \\ 0 & 0 & 1 \end{bmatrix}$$

$$K_{RGB} = \begin{bmatrix} f_x = 365.45 & 0 & c_x = 254.87 \\ 0 & f_y = 365.45 & c_y = 205.39 \\ 0 & 0 & 1 \end{bmatrix}$$

Distortion coefficients:

$$k_{IR} = \begin{bmatrix} k_1 = 0.0905474 & k_2 = -0.26819 & k_3 = 0.0950862 \\ p_1 = 0 & p_2 = 0 & \end{bmatrix}$$

$$k_{RGB} = \begin{bmatrix} k_1 = 0.04229 & k_2 = -0.05348 & k_3 = -0.00024 \\ k_4 = 0.00335 & p_1 = 0 & p_2 = 0 \end{bmatrix}$$

Camera translation and rotation:

$$T = \begin{bmatrix} 41.05705 & 1.23825 & 1.41714 \end{bmatrix}$$

$$R \sim I$$

To undistort the depth image, first we look at the steps in how a point in the real world is projected into the depth camera and gets its readings. X is the coordinates of the world points, and we first change the coordinate system with respet to the camera.

$$R(X - C) = \begin{bmatrix} pz \\ qz \\ z \end{bmatrix}$$

where R is the rotation(set to identity) and C is the camera center. Real world points is projected to the camera via the camera matrix as follows.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}$$

The distortion function containing the radial and tangential distortions using the distortion coefficients as below.

$$\begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = (1+k_1r^2+k_2r^4+k_5r^6) \begin{bmatrix} p \\ q \\ 0 \end{bmatrix} + \begin{bmatrix} 2k_3pq + k_4(r^2+2p^2) \\ 2k_3pq + k_3(r^2+2q^2) \\ 1 \end{bmatrix}$$

$$r^2 = p^2 + q^2$$

The information we get from the camera is $[x, y, d]^T$ which is calculated with respect to the real world depth z as

$$\begin{bmatrix} x \\ y \\ d \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \\ \dfrac{1}{zc_x} - \dfrac{c_y}{c_x} \end{bmatrix}$$

So now we just perform inverse operation of the process described above.

$$X = \frac{1}{c_1d + c_0} dist^{-1}(K_{IR}^{-1} \begin{bmatrix} x + u_0 \\ y + v_0 \\ 1 \end{bmatrix}, k_{IR})$$

Where $dist$ is the distortion method used above. Then we reproject the world point onto the RGB camera using

$$u_{RGB} = K_{RGB} dist(RT(X_{IR} - C_{RGB}), k_{RGB})$$

4.1.3 Face tracking

The project perform face detection using a pre-trained Haar feature-based cascaded classifier to mark the regoin of interest(ROI) for the later image processing. The basic working principle of this classifier is to learn a collection of Haar-like features which yield a high detection rate on faces. Each Haar-like feature considers adjacent rectangular regions in a window, sums up the pixel intensities in each region, and calculates the difference between these sums. A weak classifier is trained for each such feature by determining a threshold value for the difference which best separates faces from non-faces on the training set. By cascading several hundred weak classifiers in series, the accuracy of the system increases. The project uses Haar cascade classifiers for this task because they are simple, and most importantly, fast enough to achieve real-time classification on a modern laptop.

4.2. Expression Recognition

4.2.1 Data Collection

The dataset used for this project comes from the second version of the Cohn-Kanade database called Cohn-Kanade Plus (CK+/CKP)[10]. CK+ includes 593 sequences of images across 123 subjects, and each of the sequences contains photos of a human face changing from neutral (no expression) gradually to peak expression. 327 sequences among the 593 have been given an expression label among 7 prototypic emotions: Anger, Contempt, Fear, Disgust, Happiness, Sadness and Surprise. We used the 327 sequences with label as our dataset, the total number of the images is 5876.

The images and the labels are provided separately and are scattering in several files. We further processed the given data and arranged them into one single .lmdb file as the preparation for Caffe usage.

4.2.2 Model Training

To achieve our goal of expression recognition, we need to train a model on facial expressions images to classify the input photos into eight main emotions: Anger, Contempt, Fear, Disgust, Happiness, Sadness and Surprise and an additional Neutral tag for those images with no emotions. The deep learning framework Caffe [11] is used to achieve feature extraction from the image dataset. After training Caffe and collecting features, we replace the built-in classification model with our own learning algorithm. The training result of raw Caffe data also give us an insight of the expected performance of our classification algorithm.

Environmental Settings The pre-trained GoogLeNet Caffe model [12] was used as a starting point of our model training (feature extraction). We inherited and fine-tuned the structure and the parameters form GoogLeNet and plugin our dataset to retrieve the preliminary training result. There are 22 layers with parameters (convolution layer & fully-connected layer) and five pooling layers scattering in the middle connecting them, so the network has 27 main layers. Counting all the minor layers composing the main layers, the network is about 100 layers deep. The detailed description of the network can be found in [12].

Training results We completed the of training the data set with established GoogLeNet [12] model with 240000 iterations. With the preliminary results in mind, we are tuning the layer by layer parameters as well as determining the feature extraction interruption in a trained model and apply further algorithms to classify unseen data with higher precision.

Facial Feature Extraction After training with Caffe using our data set, we extracted parameters (filters) for each layer, thus extracting the feature information afterwards. Figure Fig. 3a is one example from the CK+ dataset of label sad. Figure Fig. 3b shows the filters used for convolution layer 1. Figure Fig. 3c Fig. 3d with their corresponding extract features (filter output). In both figure sets, the left is the original photo, the middle is the first 36 outputs of the first convolution layer, and the right are the final features,

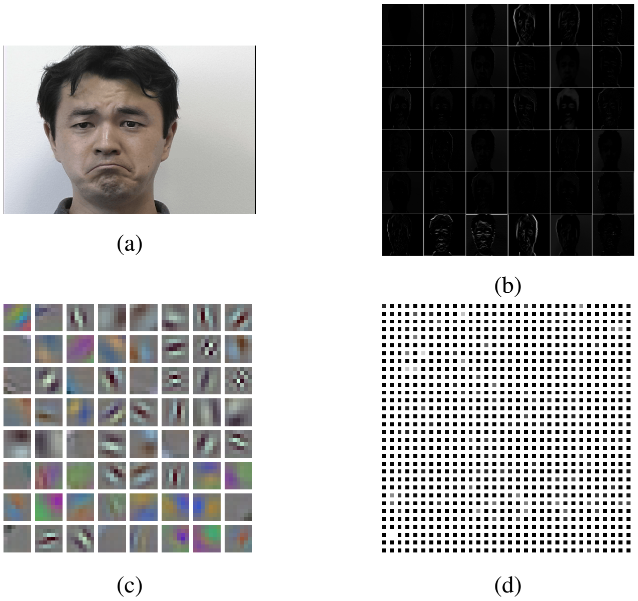


Figure 3: Neural Net Layers. (a) Dataset photo. (b) First convolution layer visualization. (c) Feature extraction after first pooling. (d) Feature extraction after final layer

which are the output of the very last polling layer. We will use these features to do further learning tasks.

4.3. Restyling

4.3.1 Image Preparation

After reading the color image and the registered depth information from Kinect, both images were first cropped based on the OpenCV ROI detected when reading the images. The face center position as well as its approximated size is being used to eliminate most unwanted areas in both RGB and depth.

Morphological Image Processing The depth image in particular is noisy and contains a lot of holes, so a morphological image processing method is used to fill out the holes first. Using slight image dilation in the gray scaled depth image, small areas missing depth information is filled out smoothly. ROI cropping and morphological image processing results are seen in Fig. 4.

Filtering After morphological image processing, the depth map still suffers from noise due to imperfection of the depth camera in Kinect. The noise will cause the uneven surfaces when performing normal map calculation. Different filters of different sizes were tested during the experiment. The final choice was a bilateral filter.

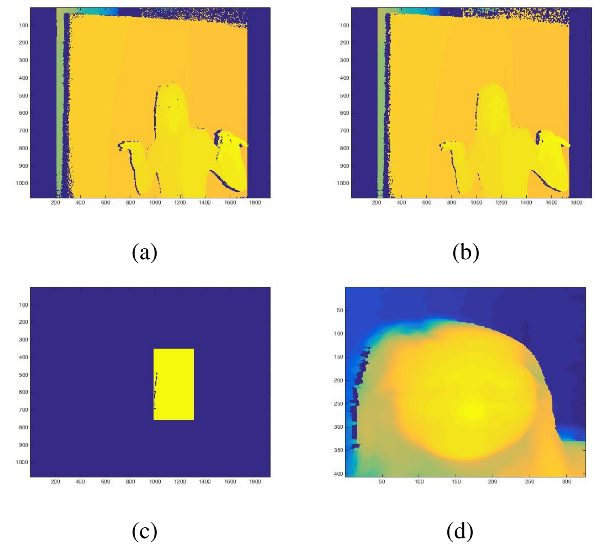


Figure 4: (a)Raw depth image. (b) After morphological image processing. (c) Cropped binary mask ROI in yellow (d) Cropped depth map ROI

$$g(i,j) = \frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$w(i,j,k,l) = exp(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{||f(i,j) - f(k,l)||^2}{2\sigma_r^2})$$

Where $f(i,j)$ is one pixel value of the image, (k,l) is the surrounding kernel(window of size of choise) pixel image, and σ_d, σ_r are the spatial and range and parameters controlling the smoothness intake of the pixel intensity relation and the spatial relation.
Originally, median and gaussian filters were tested due to their efficiency, but since median filters caused chunky depth map and gaussian filters will blur out smaller details of the face, an edge-preserving bilateral filter of $w_{size} = 8, \sigma_r = \sigma_d = 50$ was chosen to smooth out the depth information as seen in Fig. 5.

Thresholding The second step of finer cropping the ROI is using the processed depth information to crop our ROI. Using a distance threshold of 2000(mm) combined with the previous face tracking coordinates, we are able to segment out the face in RGB and depth images.

4.3.2 Normal Map Calculation

In order to render more realistic lighting with the face, a normal map calculation has to be performed in advance to determine the landscape of the facial area. The normal map

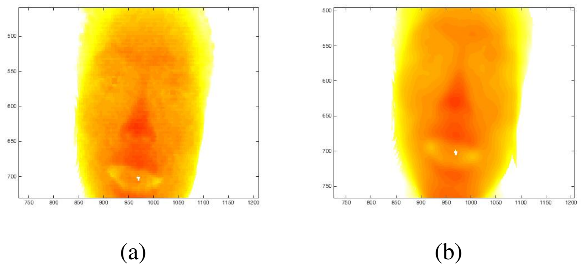
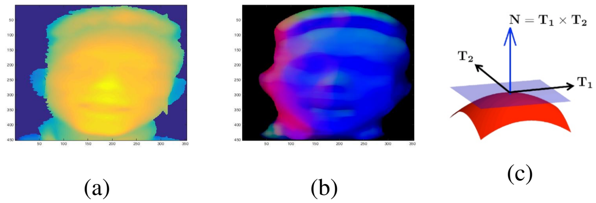

(a) (b)

Figure 5: Bilateral filtering. (a) Unfiltered depth data. (b) Bilateral-filtered depth data.



(a) (b) (c)

Figure 6: Normal map calculation. (a) Depth data. (b) Corresponding normal map(R:x, G:y, B:z). (c) Normal vector calculation

can help determine how the light interact with human skin. The calculation of normal map is essentially the cross product of a small window's x and y direction vector as depicted in Fig. 6. Normal vector V_{normal} at the depth map pixel $D(x, y)$ was calculated as

$$dir_x = (D(y, x + 1) - D(y, x - 1))/2$$

$$dir_y = (D(y + 1, x) - D(y - 1, x))/2$$

$$dir_z = \frac{1}{\sqrt{dir_x^2 + dir_y^2 + 1}}$$

$$V_{normal} = \begin{bmatrix} \frac{dir_x}{dir_z} \\ \frac{dir_y}{dir_z} \\ \frac{1}{dir_z} \end{bmatrix}$$

4.3.3 Relighting

Normal map of the face is used to determine the lighting of a particular pixel. A Lambertian reflectance model is used to calculate the pixel color and intensity in relighting. Assuming the skin surface is diffusive, the model provides quick calculation of inner product of the pixel's normal vector with respect to the vector pointing from the pixel toward the light source. The equation of a Lambertian reflectance model is as below

$$I_D = I_{org}\mathbf{L} \cdot \mathbf{N} C I_L = I_{org}|L||N|cos(\alpha)CI_L$$

where I_{org} is the original pixel value, I_D is the resulted pixel value, L and N are the light vector and normal vector, C and I_L are the color and intensity of the incoming light, and α is the angle between L and N. The relighting effects can be seen in Fig. 7.

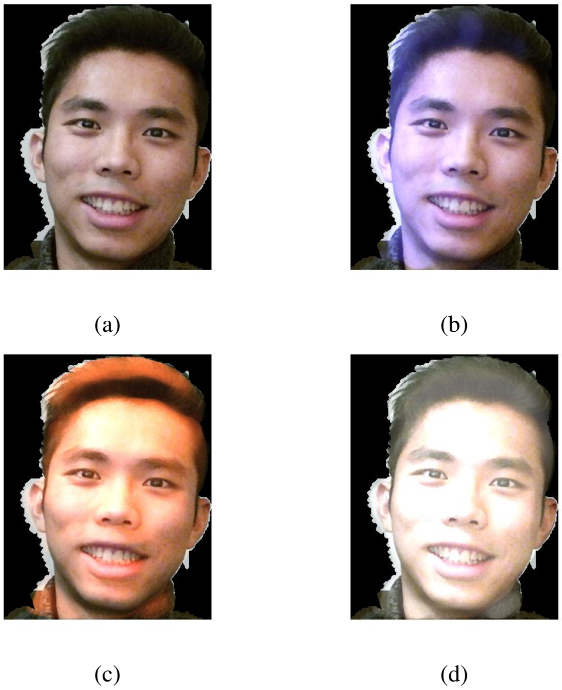
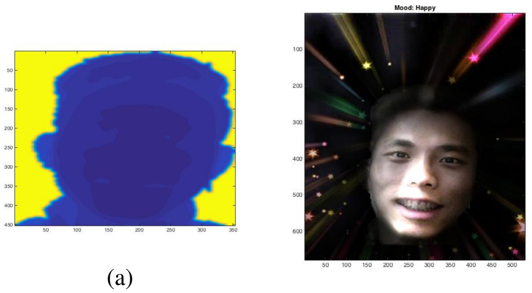

(a) (b) (c) (d)

Figure 7: Relighting. (a) Original plane photo. (b) Relit from -x direction with blue light. (c) Relit from +y direction with green light. (d) Relit in +z direction with white light.

4.3.4 Masking and Background Composition

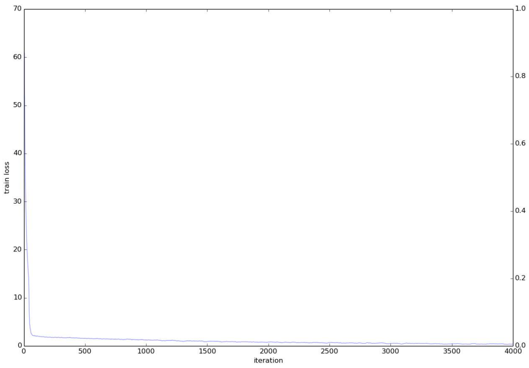
The final step of the process is combining the relit face with a particular background according to the label. Before compositing the two images into one, a mask of face and inverse masking of background has to be applied. A binary mask is produced using thresholded depth map. Erosion is then performed on the mask to enlarge the region. A large gaussian filter create smooth gradient along the edges. The mask is applied as intensity reduction for the face in order to eliminate imperfection of the edge of the face in previous steps. And the inverse version is applied to the background to prevent from direct edge overlapping which will cause an odd visual perception. The mask is depicted as Fig. 8



(a)

(b)

Figure 8: (a) A mask produced using depth thresholding, erosion, and gaussian filtering. (b) An example output of final relighting and masking.



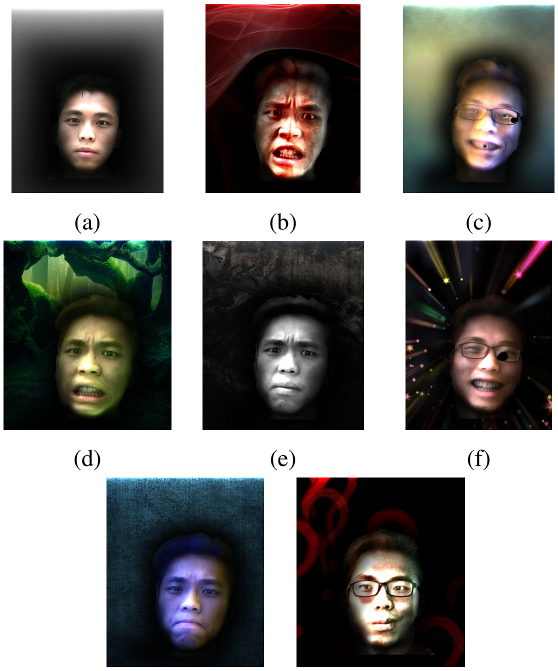
(a)

Figure 9: Training loss curve of Googlenet in Caffe of the first 4000 iterations

5. Results

5.1. CNN Training Accuracy

We ran the Googlenet model for 240000 iterations with an average loss of 0.005 as depicted in Fig. 9. The average loss is approximately 0.05 after the iterations. The overall model accuracy is around 60% tested with 80+ labeled data different from the training data. Expressions such as neutral, anger, happy, and sad were more precise than contempt, suprise, and fear. Since the intr-class variation for expression recognition is high due to difference among individual translations of the same emotion. The dataset used was considerably small of only 1600+ images. Since the CK+ dataset also contains subtle expression of nearly neutral, those were not used since if labeled 'neutral', the number of data will largely overpower the others in quantity.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 10: Final example images of (a) neutral (b) angry (c) contempt (d) disgust (e) fear (f) happy (g) sad (h) surprise

5.2. Rendered Images

The rendered images can be seen in Fig. 10, picked from correct labeled events.

5.3. Image Quality Analysis

Since I also varied the distance from the Kinect to test the depth perception quality, large holes that were not filled completely can be observed in some of the pictures in Fig. 10 due to the close proximity of the face with Kinect. The default range of Kinect depth detection range is from $50cm$ to $4.5m$, which limits the system but at the same time provides great thresholding for image processing. On the other hand, since the Kinect RGB camera ISP already has auto-white balancing and auto-exposure, color and intensity can be left raw at this point. Under lower lighting condition, pixel noise will rise but would not severely effect face-tracking and relighting and filtering eliminated part of the noise.

6. Future Work

The system currently are dealing with only front-facing portrait mode, but did not test its capability in side-way facial images. If newer techniques can compensate the incomplete face recognition or deploy a multi-camera system, a real-time tracking and editing system can be built.

This project, considered as a prototype, proved the feasibility of automatic visual effects, but is not yet optimized for real-time application since bilateral filtering takes approximately 10 seconds per images and neural network evaluation takes around 1 second per frame. If algorithm and hardware are optimized as well as lowering the recognition sampling rate, a real time version of similar system can be built for live streaming events.
The relighting currently adds addition light sources to an existing photo, but did not do exposure compensation before applying new effects. In extreme lighting conditions, input faces will have a high dynamic range, which can be delighted using bright and dark area detection and compensation can be done with virtual lightings accordingly. Similar to the work done by [3], the system still functions if using an ordinary camera instead, since we can use a predefined morphable 3D face model and the face geometry to create the same lighting effect.

7. Conclusion

The report have demonstrated that the concept of automatic special effect and rerendering is possible combining existing image processing algorithms, hardwares, and the emerging machine learning tools. Different from plane color filters, the system takes in priors of human facial structure as well as expression to perform effects closely related to characters.
The automatic restyling system consists of mainly three components, the hardware interface for image acquisition, image processing for rendering and editing, and machine learning for expression recognition. For image acquisition, the Kinect camera modules quality are good enough to produce encouraging results, while there were minor issues regarding the supporting platforms, there are existing open sourcewares that eliminate the barrier. For image processing and rendering, well known filters and morpholigical image processing methods provide efficient and effective tools to fulfill the needs. Although the depth information from Kinect is not as good in resolution as the color images, it still provides sufficient structural information to be used to relight the face. Finally, expression recognition using convolutional neural network is proven to have higher accuracy ones the dataset is sufficient, several models is still being tested for the best recognition accuracy and remains a popular research area as for now.
With the growing market of live streaming and social media platforms with quick and easy editing, there is not doubt that incorporating artificial intelligent special effect editors will be an useful tool in the market someday.

References

[1] Matthias Ziegler, Andreas Engelhardt, Stefan Mller, Joachim Keinert, Frederik Zilly, Siegfried Foessel. *Multi-camera system for depth based visual effects and* compositing CVMP, 2015.

[2] Lingzhu Xiang, ., Florian Echtler, ., Christian Kerl, ., Thiemo Wiedemeyer, ., Lars, ., hanyazou, ., ? Alistair, . (2016). libfreenect2: Release 0.2 [Data set]. Zenodo. http://doi.org/10.5281/zenodo.50641

[3] Yang Wang, Lei Zhang, Zicheng Liu et.al *Face Relighting from a Single Image under Arbitrary Unknown Lighting Conditions.* IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 31, Issue: 11, Nov. 2009)

[4] Zhen Wen, Zicheng Liu, Thomas S. Huang. *Face Relighting with Radiance Environment Maps.* CVPR, 2003.

[5] Lembit Valgma. *3D Reconstruction Using Kinect v2 Camera.* Bachelor's thesis (12 ECTP). 2016

[6] Jan Smisek, Michal Jancosek, and Tomas Pajdla. *3D with Kinect.* Chapter 1, Consumer Depth Cameras for Computer Vision. Springer ACVPR. 2016

[7] Fasel, B. *Robust Face Analysis Using Convolutional Neural Networks.* Object Recognition Supported by User Interaction for Service Robots (2001): 1-48.

[8] Yu, Zhiding, and Cha Zhang. *Image Based Static Facial Expression Recognition with Multiple Deep Network Learning - Microsoft Research.* Microsoft Research. IEEE, Nov. 2015.

[9] Vinyals, Oriol, Blundell, Charles, Lillicrap, Timothy, Kavukcuoglu, Koray, and Wierstra, Daan. *Matching networks for one shot learning*. arXiv preprint arXiv:1606.04080, 2016.

[10] Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. *The Extended Cohn-Kanade Dataset (CK): A Complete Dataset for Action Unit and Emotion-specified Expression.* 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops (2010): n. pag. Web.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. *Caffe: Convolutional Architecture for Fast Feature Embedding.* In Proceedings of the 22nd ACM international conference on Multimedia (MM '14). ACM, New York, NY, USA. (2014): 675-678.

[12] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. *Going Deeper with Convolutions.*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): n. pag. Web.