COMP 4801 Final Year Project
Final Report

Project Code : FYP17004
Project Name : Real-time Coherent Video Style Transfer
Student : Gao Chang 3035141748
Supervisor : Prof. Yizhou Yu
Second Examiner : Dr. Dirk Schnieders
Real-time Coherent Video Style Transfer

Gao Chang, Gu Derun
Department of Computer Science
The University of Hong Kong

Abstract

With the increasing popularity of deep learning and computer vision, the field of image style transfer using convolutional neural networks has gained interests of researchers globally. Although there have been mature solutions for style transfer on images, they usually suffer from high temporal inconsistency when applied to videos. Some video style transfer models have recently been proposed to improve temporal consistency, yet they fail to guarantee fast processing speed, nice perceptual style quality and high temporal consistency at the same time. In this project, we propose a novel real-time video style transfer model, ReCoNet, as a solution to this problem. ReCoNet is a feed-forward convolutional neural network which can generate temporally coherent style transfer videos in real-time speed. In addition to the loss functions used in previous video style transfer models, it also adopts a novel luminance warping constraint to increase stylization stability under illumination effects and a novel feature-map-level temporal loss to further enhance temporal consistency on traceable objects. Experimental results indicate that ReCoNet exhibits outstanding performance both qualitatively and quantitatively. A corresponding computer application has also been developed to stylize streaming videos using ReCoNet models pre-trained on different style images.
Contents

1 Introduction 5

2 Project Background and Literature Review 11
  2.1 Origin of Image Style Transfer 11
  2.2 Neural Style Transfer 13
  2.3 Video Style Transfer 19

3 Methodology 22
  3.1 Motivations 23
    3.1.1 Feature-map-level Temporal Loss 23
    3.1.2 Luminance Difference 24
  3.2 Network Architecture 26
  3.3 Loss functions 28
    3.3.1 Perceptual Losses 29
    3.3.2 Output-level Temporal loss 30
    3.3.3 Feature-map-level Temporal loss 31
    3.3.4 Summary 32

4 Experiments and Results 32
  4.1 Implementation Details 32
  4.2 Comparison to Methods in the Literature 36
    4.2.1 Quantitative Analysis 36
    4.2.2 Qualitative Analysis 37
  4.3 Ablation Study 42
    4.3.1 Temporal Loss on Different Levels 42
1 Introduction

As a popular art form, painting has attracted interests of people for thousands of years. Artists use this specific tool to represent the actual image of the world or their illusion with painted contents, and render them with their distinctive artistic styles. However, the algorithmic basis of the interplay between the content and style of an artwork was not quantitatively analyzed until recent decades [1]. For the past twenty years, scientist have been working to develop methods to separate fantastic artistic styles from the paintings and append them to other ordinary images. This process is usually referred to as style transfer, where the contents of the target image and the artistic style of the artwork should be both preserved. Figure [1] demonstrates how the style of Udnie painted by Francis Picabia in 1913 could be appended to a scene in the movie Sintel using Johnson et al’s model [2], where both the color choices and perceptual styles of the original image are transferred to those of the painting Udnie with few changes to the original object contents.

Among existent style transfer methods, Non-photorealistic Rendering (NPR) [3, 4, 5] has been widely studied and firmly developed. Although NPR methods have achieved inspiring image transfer results, they still suffer from high dependence on specific styles they utilize [5, 6], which restrict their reproductive ability on different artistic styles. Inspired by the power and generality of neural networks,
scientists found it possible to reproduce different artistic styles with convolutional neural networks. Gatys et al. [7, 6] first developed a neural algorithm for automatic image style transfer, which uses a convolutional neural network to refine a random noise to a stylized image iteratively constrained by a content loss and a style loss. This method inspired many later image style transfer models [2, 8, 9, 10, 11, 12, 13, 14]. One of the most successful successors is the perceptual losses model proposed by Johnson et al. [2], using a feed-forward convolutional neural network to stylize images and a pre-trained VGG network [15] to compute perceptual losses. Although Johnson et al.’s model and some of other image style transfer models have nicely reproduced the artistic style and achieved
near real-time inference speed (at least 24 FPS), one of the most critical issues in their stylization results is high temporal inconsistency. Temporal inconsistency, or sometimes referred to as temporal incoherence, can be observed visually as flickering between consecutive stylized frames and inconsistent stylization of moving objects [16]. Figure 2(a) and 2(b) demonstrate temporal inconsistency in video style transfer.

Figure 2: Temporal inconsistency in video style transfer. (a) The style target image Edtaonisl (Francis Picabia, 1913) and two consecutive video frames downloaded from Videvo.net [17] (b) Style transfer results by Chen et al [16] (c) Style transfer results by ReCoNet. The circled regions show that ReCoNet can better suppress temporal inconsistency while Chen et al’s results exhibit noticeable flickering and inconsistent color.
To mitigate this effect, Anderson et al \cite{18} and Ruder et al \cite{19} proposed the *temporal loss* optimization method guided by optical flows and occlusion masks. Although their methods can generate smooth and coherent stylized videos, it generally takes several minutes to process each video frame due to optimization on the fly. Some recent models \cite{16, 20, 21, 22} improved the speed of video style transfer using optical flows and occlusion masks explicitly or implicitly, yet they failed to guarantee real-time processing speed, nice perceptual style quality, and coherent stylization at the same time. Now a new problem arises: is there a better way to achieve high temporal coherence and rich perceptual style of the stylized videos while maintaining real-time inference speed?

As a research product of this project, a real-time coherent video style transfer network, *ReCoNet*, is proposed as a solution to the aforementioned problems. ReCoNet is a feed-forward neural network which can generate coherent stylized videos with artistic styles perceptually similar to the style target in real-time speed. It stylizes videos frame by frame through an encoder and a decoder, and uses a VGG loss network \cite{2, 15} to capture the perceptual style of the transfer target. It also incorporates optical flows and occlusion masks as guidance in its temporal loss to maintain temporal consistency between consecutive frames, and the effects can be observed in Figure 2(c). In the inference stage, ReCoNet can
run far above the real-time standard on modern GPUs due to its lightweight and feed-forward network design.

In addition to the new network design, we also propose a new loss constraint and a new loss function to further improve temporal consistency. First, we find that the brightness constancy assumption \[23\] in optical flow estimation may not strictly hold in real-world videos and animations, and there exist luminance differences on traceable pixels between consecutive image frames. Such luminance differences cannot be captured by temporal losses purely based on optical flows. To consider the luminance difference, we introduce a novel luminance warping constraint in the temporal loss at the output of the decoder in our network. Second, from stylization results of previous methods \[16, 20, 21, 19, 22\], we have also observed instability such as different color appearances of the same moving object in consecutive frames. With the intuition that the same object should possess the same features in high-level feature maps, we apply a novel feature-map-level temporal loss to the output of the encoder in our network. This further improves temporal consistency of our model.

Moreover, a demonstrative application has also been developed to perform video style transfer on devices with a modern GPU and a web-cam. The application can display both the actual video streamed by the web camera and its stylization result at the same time, providing multiple transformation styles to select.
In summary, there exist the following contributions in this project:

- A novel feed-forward neural network model named ReCoNet is proposed for video style transfer. It highly incorporates perceptual style and temporal consistency in its stylization results and can achieve an inference speed more than 200 frames per second (FPS) on a single modern GPU. ReCoNet also possesses the generality to reproduce various artistic styles on videos.

- A novel luminance warping constraint is introduced to the output-level temporal loss in ReCoNet to specifically consider luminance changes of traceable pixels in input videos. This constraint can improve stylizing stability of areas under illumination effects and help suppress overall temporal inconsistency.

- A novel feature-map-level temporal loss is introduced to ReCoNet to penalize variations in high-level features of the same object in consecutive frames. This loss can improve stylization stability of traceable objects in the output videos.

- An application using ReCoNet has been developed to stylize streaming videos using webcams. Multiple stylization targets are provided using pre-trained ReCoNet models.
The project background and literature review will be first introduced in Section 2. Detailed motivations, the network architecture, and the loss functions will be presented in Section 3. In Section 4, both quantitative and qualitative experimental results will be reported and analyzed, with ablation study on the two novel losses in ReCoNet. The application for streaming stylization using ReCoNet is demonstrated in Section 5.

2 Project Background and Literature Review

In this section, the history of image style transfer and video style transfer will be introduced, with specific examinations of some popular yet somewhat problematic models for image and video stylizations. This section focuses primarily on what is style transfer and how to perform style transfer, especially on the artistic styles in painting, the neural networks used for style transfer (iterative and feed-forward), the perceptual losses, temporal consistency, optical flows, and occlusion masks.

2.1 Origin of Image Style Transfer

For thousands of years, painting has been a popular art form representing realistic or fantasy scenes. A painting is generally made to reproduce the existent
or imaginary objects, including natural items, beautiful patterns, and meaningful shapes, with some consistent or inconsistent artistic styles, including color, strokes, shape patterns, and textures. In the past, reproducing a painting usually required high craftsmanship and costly manual work with limited technological support. Even though the technology nowadays supports almost identical copying of an artwork, the artistic styles inside are hard to be reproduced on other scenes only with high-quality copying.

Since around 1990s, the algorithmic theory of how painting depicts and portrays scenes into artistic works has gained interests of computer scientists. More specifically, they began to research on how to extract the artistic styles in the artworks and reproduce such styles on other scenes systematically [1]. If such algorithmic methods could be obtained, then everyone can use such methods to turn photographs or other pictures into art masterpieces. These methods are usually referred to as image style transfer, or image stylization. The image to be transferred into other styles is called content image, and the image containing artistic styles to be reproduced is called style image.

Non-photorealistic Rendering (NPR) is typical one of the early fields that have been firmly developed as a solution to image style transfer [3 4 5]. However, NPR methods usually require specific designs on specific artistic styles, without much generalization ability of capturing and reproducing different styles with the
same sets of settings or hyper-parameters. Moreover, NPR methods are heavily dependent on the kinds of artistic styles. They reproduce well only in some certain kinds of styles such as oil paintings and animations [1]. Scientists then turn their interests to methods with much greater generalization ability and less requirements on the artistic styles.

2.2 Neural Style Transfer

With the increasing popularity of neural network in 2010s, researchers began to realize that the neural network is a brilliant tool to learn complicated and underlying knowledge of things representable by digital inputs. Since the training and prediction of neural network could be accelerated by specifically designed modern devices such as graphics processing unit (GPU) and field-programmable gate array (FPGA), neural networks became practical to be used by individual researchers and even the general public, which has boosted the advancements in neural network and deep learning (machine learning with deep neural networks) in recent years.

Since images can be digitalized into two-dimensional data and are extremely common and widely used in daily life, the study of images using neural network is one of the major topic in deep learning. As spatial data are usually critical in images, the process of convolution broadly used in computer vision has been
introduced to neural network to process image data and extract the underlying information. Such neural networks are named as convolutional neural network (CNN) [24]. CNN has demonstrated its strong ability in many image-related areas such as object detection [25, 26], image classification [27, 15], image segmentation [28, 29], etc. Intuitively, as paintings are or can be scanned into digital images, the convolutional neural networks should possess the ability to capture the artistic styles in the artworks and reproduce them on other images.

Figure 3: Feature reconstruction results of each layer in the model proposed by Gatys et al [7, 6]
Inspired by the VGG convolutional neural network design [15], Gatys et al [7,6] first developed a neural algorithm that uses CNN to conduct automatic image style transfer. The neural algorithm starts with a random noise image as the input image and passes it through the neural network iteratively. The neural network utilizes the layers “conv1_1, conv2_1, conv3_1, conv4_1, conv5_1” of the original pre-trained VGG network [15] and fixes the network parameters. Unlike image prediction tasks where the network usually directly outputs a predicted label or image processing tasks where the network usually directly generate a processed image, the neural algorithm generates a large number of feature maps of the input image at each network layer. These feature maps contain the perceptual information of the input image, with low-level information at layers close to the input and high-level information at layers close to the end of the network. Such perceptual information extraction and reconstruction effects are demonstrated in Figure 3. A content loss function and a style loss function are computed on these feature maps and then back-propagated to the input image pixels. In this way, the neural algorithm updates the input noise image iteratively until the loss function converges, and the updated noise image will finally look perceptually similar to both the content image and the style image. The back-propagation process is shown in Figure 4.
As the core part of the neural algorithm, the content and style loss functions serve as difference purposes. The content loss function aims to minimize the difference between the feature maps of the input noise image and the original content image. The feature maps of the content image are pre-computed through the fixed-weight style transfer network, while the feature maps of the input noise image are computed for each iteration since it is also iteratively updated. An L2 norm is then adopted as the criteria to minimize the distance of those two feature maps, and when it converges, the updated noise image should possess similar perceptual information as the content image. The style loss is similar to
the content loss and is designed to minimize the difference between the feature maps of the input noise image and the style transfer target image. However, instead of directly minimizing the L2 norm, the style loss uses the Gramian matrix to compare the feature correlations of the two feature map groups. Theoretically, matching the Gramian matrix of the feature maps of input noise image and the style target is equivalent to minimize the maximum mean discrepancy of the two perceptual information distributions [1]. Therefore, the Gramian matrix possesses the ability to reproduce perceptual artistic styles on the updated input noise.

This pioneering algorithm suddenly attracted the attention of both researchers and engineers. Many neural style transfer algorithms [2, 8, 9, 10, 11, 12, 13, 14] have later been proposed to improve Gatys et al [7, 6]’s model as its iterative updates are extremely time-consuming and its style transfer ability is still rather primitive. Many computer and mobile softwares have also been developed and released for style transfer using pictures stored in computers or mobile phones, such as Prisma [30], Ostagram [31], and Artisto [32]. These works have boosted the advancement in neural style transfer greatly.

One of the most successful succeeder of the neural algorithm is the perceptual losses model proposed by Johnson et al [2], as shown in Figure 5. Instead of iteratively refining a random noise, the perceptual losses model uses a feed-forward
image transform network which takes a content image as the input and directly outputs its stylized output, and a pre-trained VGG network \[15\] to compute the perceptual losses (a content loss, a style loss, and a total variation regularizer). In the training stage, the network parameters for the VGG network is also fixed to extract the feature maps of the content image, style image, and the stylized output, then the perceptual losses computed on those feature maps will be used to back-propagate and update the parameters of the image transform network. In the inference stage, only the image transform network will be used to directly stylize an input image without any additional computation. It turns out that Johnson et al’s model and some of other similar feed-forward image style transfer models have nicely reproduced the artistic style with rich perceptual information and achieved near real-time inference speed (at least 24 FPS) in the inference stage, making style transfer more robust and fast for practical usage.
2.3 Video Style Transfer

Although these image style transfer models have achieved nice perceptual style quality and near real-time inference speed, severe flickering artifacts can be observed when applying this method frame by frame to videos since temporal consistency is not considered. To mitigate such temporal inconsistency, Anderson et al [18] and Ruder et al [19] introduced a temporal loss function in video stylization as an explicit consideration of temporal consistency. The temporal loss utilizes optical flows and occlusion masks and is iteratively optimized for each frame until the loss converges: the optical flows indicate the direction and distance of movement for each pixel from the previous frame to the current frame, and the occlusion masks indicate whether a pixel a traceable or not in two consecutive image frames; since traceable pixels should preserve their traceability and temporal motion in their stylized results, the optical flows and occlusion masks computed or obtained from the original consecutive input frames can be used as a constraint on the stylized consecutive input frames to maintain temporal information. However, it generally takes several minutes for Anderson et al’s and Ruder et al’s models to process each video frame, which is not applicable for real-time usage. Although Ruder et al [22] later accelerated the inference speed, their video stylization still runs far below the real-time standard.
To obtain a temporally consistent and fast video style transfer method, some real-time or near real-time models have recently been developed. Chen et al. [16] proposed a recurrent model which requires feature maps of the previous frame in addition to the input consecutive frames, and conducts explicit optical flows warping on feature maps in both training and inference stages (Figure 6(a)). Since this model requires optical flow estimation by FlowNetS [33] in the inference stage, its inference speed barely reaches real-time level and its ability to reduce temporal inconsistency is susceptible to errors in optical flow estimation. Gupta et al. [20] also proposed a recurrent model which takes an additional
stylized previous frame as the input. Although their model performs similarly to Chen et al.’s model in terms of the ability to maintain temporal consistency, it suffers from transparency issues and still barely reaches real-time inference speed. Adopting a feed-forward network design, Huang et al. [21] proposed a model similar to the perceptual losses model [2] with an additional temporal loss (Figure 6(b)). This model is faster since it neither estimates optical flows and occlusion masks nor uses information of previous frames in the inference stage. However, Huang et al.’s model calculates the content loss from a deeper VGG loss network layer `relu4_2`, which is hard to capture low-level features of the content image. Strokes and textures are also weakened in their stylization results due to a low weight ratio between perceptual losses and the temporal loss.

Observing the strengths and weaknesses of these models, several improvements are adopted in ReCoNet. Compared with Chen et al. [16]’s model, ReCoNet does not estimate optical flows and occlusion masks but involves ground-truth optical flows only in loss calculation in the training stage. This can avoid optical flow prediction errors and accelerate inference speed. Meanwhile, ReCoNet can render style patterns and textures much more conspicuously than Huang et al. [21]’s model, which could only generate minor visual patterns and strokes besides color adjustment. With a lightweight and feed-forward network de-
sign, ReCoNet can run faster than all aforementioned video stylization models [16, 20, 21, 19, 22, 18] during the inference stage.

3 Methodology

Figure 7: The pipeline of ReCoNet. $I_t$, $F_t$, $O_t$ denote the input image, encoded feature maps, and stylized output image at time frame $t$. $M_t$ and $W_t$ denote the occlusion mask and the optical flow between time frames $t - 1$ and $t$. $Style$ denotes the artistic style image. The dashed box represents the results from the previous frame, which will only be used in the training stage. Red arrows and texts denote the loss functions for training.

The training pipeline of ReCoNet is demonstrated in Figure 7. There are three major modules in ReCoNet: an encoder which converts input image frames to encoded feature maps, a decoder which generates stylized images from feature maps, and a VGG-16 [15] loss network to compute the perceptual losses. Addi-
tionally, a multi-level temporal loss is added to both the output of encoder and the output of decoder to reduce temporal incoherence. In the inference stage, neither loss networks nor historical information will be used. The encoder and the decoder will directly stylize videos frame by frame.

3.1 Motivations

There are two novel loss functions/constraints introduced to ReCoNet: a feature-map-level temporal loss appended to the output of the encoder, and a luminance warping constraint added to the temporal loss at the output of the decoder. Below are the motivations for the two losses.

3.1.1 Feature-map-level Temporal Loss

Based on the intuition that the same object should preserve the same representation in high-level feature maps, a new feature-map-level temporal loss is proposed for feature-map-level consistency. Although warping frames directly at the output level may not be accurate due to illumination effects and possible errors in optical flows, the same method is proved to be suitable at the feature-map level [16]. Therefore, the (down-scaled) ground-truth optical flows and occlusion masks are used to calculate feature-map-level temporal loss between the warped feature maps and the current ones. Experiments in Section 4.3.1 show that the
introduction of the feature-map-level temporal loss can improve stylization consistency of ReCoNet on the same object in consecutive frames.

3.1.2 Luminance Difference

In real-world videos, the same object in consecutive frames may alter their luminance and chromaticity (color appearances) due to illumination effects. In such cases, the video data does not satisfy the brightness constancy constraint [23] assumption, and warping optical flow directly on the frames will ignore luminance and chromaticity changes in traceable pixels [33, 34, 35].

Also in animations, many datasets, such as MPI Sintel [36], use the albedo pass to generate ground-truth optical flows and later render the scenes with illuminations to obtain the clean or final pass, including smooth shading, self shadowing, darkening in cavities, darkening to objects close to a surface, specular reflections, inter-reflections, and mirroring effects. Such illumination post-processing may also lead to differences on luminance and color chromaticity.

To further examine the difference in luminance and chromaticity, we calculated and averaged the absolute value of temporal warping error $I_t - W_t(I_{t-1})$ over MPI Sintel Dataset and 50 real-world videos download from Videvo.net [17], where $W$ is the forward optical flow and $I$ is the input image frame. We used FlowNet2 [35] to estimate the optical flows and the method of Sundaram et
al [37] to generate the occlusion masks for downloaded videos. Figure 8 shows the histograms of temporal warping error in both RGB and XYZ color space.
Two conclusions are drawn based on the results. First, RGB channels share similar warping error distributions, and there is no bias of changes in value among color channels. Second, despite changes in relative luminance channel Y, the chromaticity channels X and Z in XYZ color space also contribute to the total inter-frame difference. However, since no exact guideline of chromaticity mapping in a particular style can be determined, only the luminance difference is considered in the temporal loss.

Grounded on these findings, a novel luminance constraint in the temporal loss at the output stylized image is proposed to encourage the stylized frames to have the similar luminance changes to the input frames. Experiments in Section 4.3.2 show that this new constraint can reduce unstable color changes under illumination effects and significantly improve overall temporal coherence of the stylized frames.

3.2 Network Architecture

ReCoNet adopts a feed-forward convolution design. Compared to feed-forward networks in literature [21, 2], the whole network in ReCoNet is separated to an encoder and a decoder for different purposes. The encoder is designed to encode image frames to feature maps where perceptual information is aggregated, and the feature-map-level temporal loss is computed on its output. The decoder
<table>
<thead>
<tr>
<th>Layer</th>
<th>Layer Size</th>
<th>Stride</th>
<th>Output Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encoder</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td></td>
<td></td>
<td>3 × 640 × 360</td>
</tr>
<tr>
<td>Conv + InsNorm + ReLU</td>
<td>48 × 9 × 9</td>
<td>1</td>
<td>48 × 640 × 360</td>
</tr>
<tr>
<td>Conv + InsNorm + ReLU</td>
<td>96 × 3 × 3</td>
<td>2</td>
<td>96 × 320 × 180</td>
</tr>
<tr>
<td>Conv + InsNorm + ReLU</td>
<td>192 × 3 × 3</td>
<td>2</td>
<td>192 × 160 × 90</td>
</tr>
<tr>
<td>(Res + InsNorm + ReLU) × 4</td>
<td>192 × 3 × 3</td>
<td>1</td>
<td>192 × 160 × 90</td>
</tr>
<tr>
<td><strong>Decoder</strong></td>
<td></td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>Up-sample</td>
<td></td>
<td></td>
<td>192 × 320 × 180</td>
</tr>
<tr>
<td>Conv + InsNorm + ReLU</td>
<td>96 × 3 × 3</td>
<td>1</td>
<td>96 × 320 × 180</td>
</tr>
<tr>
<td>Up-sample</td>
<td></td>
<td>1/2</td>
<td>96 × 640 × 360</td>
</tr>
<tr>
<td>Conv + InsNorm + ReLU</td>
<td>48 × 3 × 3</td>
<td>1</td>
<td>48 × 640 × 360</td>
</tr>
<tr>
<td>Conv + Tanh</td>
<td>3 × 9 × 9</td>
<td>1</td>
<td>3 × 640 × 360</td>
</tr>
</tbody>
</table>

Table 1: Network layer specification. Layer sizes and output sizes are denoted as channel × height × width. **Conv, Res, InsNorm, ReLU, Tanh** denote convolutional layer, residual block [38], instance normalization layer [39], ReLU activation layer [40], and Tanh activation layer respectively.

is designed to decode feature maps to a stylized image where the output-level temporal loss is computed on its output. Table [1] shows the encoder and decoder design in ReCoNet. There are three convolutional layers and four residual blocks [38] in the encoder, and two up-sampling convolutional layers with a final convolutional layer in the decoder. For the upconvolution processes in the decoder, upsample layers with convolutional layers are adopted instead of traditional deconvolutional layers, which can reduce checkerboard artifacts [41] in the stylized frames. Instance normalization [39] is performed after each convo-
olution process to achieve better stylization quality. The padding of convolutional layers is set to reflection padding.

The loss network used in ReCoNet is a VGG-16 network \([15]\) pre-trained on the ImageNet dataset \([42]\). For each iteration, the VGG-16 loss network processes each of the input image frame, output image frame and style target independently and generates their feature maps, where the perceptual losses are computed.

### 3.3 Loss functions

The loss functions used in ReCoNet consist of a group of perceptual losses and a multi-level temporal loss. The perceptual losses design is inherited from and similar to the perceptual losses model \([2]\). The multi-level temporal loss design targets temporal consistency at both high-level feature maps and the final stylized output. At the feature-map level, a strict optical flow warping is used to maintain temporal consistency of traceable pixels in high-level features. At the output level, an optical flow warping with a luminance constraint is adopted to reproduce both the movements and luminance changes of traceable pixels.

A two-frame synergic training mechanism \([21]\) is used in the training process. For each training iteration, the network generates feature maps and stylized output of the first image frame and the second image frame in two separate runs. Afterwards, the temporal losses are calculated using the feature maps and styl-
ized output of both frames, and the perceptual losses are computed on each frame independently and summed up. Note again that in the inference stage, only one input frame will be processed by the network in a single run.

3.3.1 Perceptual Losses

The content loss $L_{content}(t)$, the style loss $L_{style}(t)$ and the total variation regularizer $L_{tv}(t)$ [2] are computed as the perceptual losses at each time frame $t$.

The content loss utilizes the feature maps at $j = relu3.3$ layer in the VGG-16 loss network, to penalize the perceptual difference between $I_t$ and $O_t$:

$$L_{content}(t) = \frac{1}{D_j} \| \phi_j(O_t) - \phi_j(I_t) \|^2$$  \hspace{1cm} (1)

where $D_j = C_j \times H_j \times W_j$ is the multiplication of channel size $C_j$, image height $H_j$ and image width $W_j$ at layer $j$. The style loss utilizes the feature maps at $J = [relu1.2, relu2.2, relu3.3, relu4.3]$ layers in the VGG-16 loss network, to penalize the perceptual different between $S$ and $O_t$ in Frobenius norm:

$$L_{style}(t) = \sum_{j \in J} \| G_j^\phi(O_t) - G_j^\phi(S) \|^2_F$$  \hspace{1cm} (2)

where $G_j^\phi$ is the channel-wise Gram matrix for each feature maps $\phi_j(x)$ [7]. Each entry $m, n \in C_j$ of $G_j^\phi$ corresponds to the inner product of $(c_m, c_n)$-th
channel:

$$G^\phi_j(x)_{c_m,c_n} = \frac{1}{D_j} \sum_{h,w} \phi_j(x)_{c_m,h,w} \phi_j(x)_{c_n,h,w}$$  \(3\)

where \(D_j = C_j \times H_j \times W_j\) is the multiplication of channel size \(C_j\), image height \(H_j\) and image width \(W_j\) at layer \(j\). A total variation regularizer \(L_{tv}(t)\) is also added to pursue spatial smoothness and restrain checkerboard artifacts in the stylized output \(O_t\): \([2]\):

$$L_{tv}(t) = \sum_{h,w} (\|O_{t,(h+1),w} - O_{t,h,w}\|^2 + \|O_{t,h+1,w} - O_{t,h,w}\|^2)^{\frac{1}{2}}$$  \(4\)

where \(O_{t,h,w}\) is the \((h,w)\)-th pixel on the stylized output image \(O_t\).

### 3.3.2 Output-level Temporal loss

Since temporal losses in previous works \([16, 20, 21, 19, 22]\) ignore changes in luminance of traceable pixels, the relative luminance \(Y = 0.2126R + 0.7152G + 0.0722B\), same as \(Y\) in XYZ color space, is added as a warping constraint for all channels in RGB color space in the output-level temporal loss:

$$L_{temp,o}(t-1, t) = \sum_c \frac{1}{D} M_t \| (O_t - W_t(O_{t-1}))_c - (I_t - W_t(I_{t-1}))_Y \|^2$$  \(5\)

where \(c \in [R, G, B]\) is each of the RGB channels of the image, \(Y\) the relative luminance channel, \(O_{t-1}\) and \(O_t\) the stylized images for previous and current input
frames respectively, $I_{t-1}$ and $I_t$ the previous and current input frames respectively, $W_t$ the ground-truth forward optical flow, $M_t$ the ground-truth forward occlusion mask (1 at traceable pixels or 0 at untraceable pixels), $D = H \times W$ the multiplication of height $H$ and width $W$ of the input/output image. The relative luminance warping constraint is applied to each RGB channel equally based on the “no bias” finding in Section 3.1. Section 4.3.2 further discusses different choices of the luminance warping constraint and the output-level temporal loss.

### 3.3.3 Feature-map-level Temporal loss

The feature-map-level temporal loss is adopted to penalize temporal inconsistency on the encoded feature maps between two consecutive input image frames:

$$L_{\text{temp},f}(t-1, t) = \frac{1}{D} M_t \| F_t - W_t(F_{t-1}) \|^2$$

(6)

where $F_{t-1}$ and $F_t$ are the feature maps outputted by the encoder for previous and current input frames respectively, $W_t$ and $M_t$ the ground-truth forward optical flow and occlusion mask downscaled to the size of feature maps, $D = C \times H \times W$ the multiplication of channel size $C$, image height $H$ and image width $W$ of the encoded feature maps $F$. Downscaled optical flows and occlusion masks are used to simulate temporal motions in high-level features.
3.3.4 Summary

The final loss function for the two-frame synergic training of ReCoNet is:

\[ \mathcal{L}(t-1, t) = \sum_{i \in \{t-1, t\}} \left( \alpha \mathcal{L}_{\text{content}}(i) + \beta \mathcal{L}_{\text{style}}(i) + \gamma \mathcal{L}_{\text{tv}}(i) \right) + \lambda_f \mathcal{L}_{\text{temp,f}}(t-1, t) + \lambda_o \mathcal{L}_{\text{temp,o}}(t-1, t) \]  (7)

where \( \alpha, \beta, \gamma, \lambda_f \) and \( \lambda_o \) are training hyper-parameters.

4 Experiments and Results

In this section, implementation details of ReCoNet in both the training stage and the inference stage, quantitative and qualitative comparison between ReCoNet and other video style transfer models, and ablation study of the designs in ReConet are demonstrated at length.

4.1 Implementation Details

We choose Monkaa and FlyingThings3D in the Scene Flow datasets [43] as the training dataset, and MPI Sintel dataset [36] as the testing dataset. The Scene Flow datasets provide optical flows and motion boundaries for each consecutive frames, from which the occlusion masks can also be obtained using the method
provided by Sundaram et al [37]. Monkaa dataset is generated from the animation movie Monkaa and contains around 8640 frames, resembling MPI Sintel dataset. FlyingThings3D dataset is a large dataset of everyday objects flying along random 3D trajectories which contains around 20150 frames, resembling animated and real-world complex scenes. Same as the verification process of previous works [16, 20, 21], we use MPI Sintel dataset to verify the temporal consistency and perceptual styles of video stylization results. All input frames in training and testing are resized to $640 \times 360$.

ReCoNet models are trained with a batch size of 2 for 30,000 steps, roughly two epochs over the training dataset. Consecutive frames are paired up for the two-frame synergic training, and random horizontal flipping is conducted on each pair. The frame pairs are shuffled in training process to achieve randomness in the training data. The Adam optimizer [44] is used to perform gradient descent, with a learning rate of $10^{-3}$. The the default training hyper-parameters of ReCoNet are $\alpha = 1, \beta = 10, \gamma = 10^{-3}, \lambda_f = 10^7, \lambda_o = 2 \times 10^3$. Figure 9 shows an example set of loss curves for the training process of the style Edtaonisl (Francis Picabia, 1913).

For coding and computation acceleration, we implement the video style transfer pipeline on PyTorch 0.3 [45] with cuDNN 7 [46]. All tensor calculations in the training and inference stages, including neural network propagation and
back-propagation, optical flow and occlusion mask warping, and loss function computations, are performed on a single GTX 1080 Ti GPU.

To further verify the generalization capacity of ReCoNet on real-world videos, 50 videos were downloaded from Videvo.net [17] as the validation dataset. Figure 10 demonstrates the style transfer results of four different styles on three consecutive video frames. It can be observed that the color, strokes, textures and visual patterns of the style target can be successfully reproduced by ReCoNet, and the stylized frames are visually coherent.
Figure 10: Video style transfer results using ReCoNet. The first column contains two groups of three consecutive image frames in videos downloaded from Videvo.net [17]. Each video frames are followed by two style target images and their corresponding stylized videos. The style images are *Mosaic, Dream, Autoportrait* (Picasso, 1907), and *Candy*. 

---

**Mosaic**

**Dream**

**Autoportrait**

**Candy**
4.2 Comparison to Methods in the Literature

Both quantitative and qualitative analysis are conducted to fully examine Re-CoNet and other video style transfer networks.

4.2.1 Quantitative Analysis

To quantitatively analyze the ability of video style transfer models to reduce temporal inconsistency, the temporal error $e_{stab}$ is adopted as the validation criteria. $e_{stab}$ is the square root of output-level temporal error over one whole scene, to verify the temporal consistency of traceable pixels in the stylized output:

$$e_{stab} = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T} \frac{1}{D} M_t \| O_t - W_t(O_{t-1}) \|^2}$$ (8)

where $T$ is the total number of frames, and other variables are identical to those in the output-level temporal loss.

<table>
<thead>
<tr>
<th>Model</th>
<th>Alley-2</th>
<th>Ambush-5</th>
<th>Bandage-2</th>
<th>Market-6</th>
<th>Temple-2</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al [16]</td>
<td>0.0934</td>
<td>0.1352</td>
<td>0.0715</td>
<td>0.1030</td>
<td>0.1094</td>
<td>22.5</td>
</tr>
<tr>
<td>ReCoNet</td>
<td>0.0846</td>
<td>0.0819</td>
<td>0.0662</td>
<td>0.0862</td>
<td>0.0831</td>
<td>235.3</td>
</tr>
<tr>
<td>Huang et al [21]</td>
<td>0.0439</td>
<td>0.0675</td>
<td>0.0304</td>
<td>0.0553</td>
<td>0.0513</td>
<td>216.8</td>
</tr>
<tr>
<td>Ruder et al [19]</td>
<td>0.0252</td>
<td>0.0512</td>
<td>0.0195</td>
<td>0.0407</td>
<td>0.0361</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 2: Temporal error $e_{stab}$ and average FPS in the inference stage with style Candy on different models. Five scenes from MPI Sintel Dataset are selected for validation.
Table 2 shows the temporal error $e_{stab}$ of four video style transfer models on five scenes in MPI Sintel Dataset with style *Candy*. From the results, we find that Ruder et al [19]’s model is not suitable for real-time usage due to its low inference speed, although it has the lowest temporal error among all models in our comparison. Also, ReCoNet achieves lower temporal error than Chen et al [16]’s model, primarily because of the introduction of the multi-level temporal loss. Although the temporal error of ReCoNet is higher than Huang et al [21]’s model, ReCoNet is capable of capturing strokes and minor textures in the style image while Huang et al’s model is hard to. Please refer to the qualitative analysis below for details.

It is also worth mentioning that ReCoNet and Huang et al’s model achieve far better inference speed than the others models. This is because, compared with Chen et al’s model and other recurrent models [20, 22], feed-forward models are easier to be accelerated with parallelism since each iteration does not need to wait for the previous iteration to be fully processed.

### 4.2.2 Qualitative Analysis

The qualitative analysis is conducted between ReCoNet and other real-time models proposed by Chen et al [16] and Huang et al [21].
Figure 11: Qualitative comparison of style transfer results in the literature. (a) Style transfer results between Huang et al [21]’s model and ReCoNet on image frames. (b) Style transfer results between Chen et al [16]’s model and ReCoNet on consecutive image frames with zoom-ins of flickering regions.

Figure 11(a) shows the stylization comparison between Huang et al ’s model and ReCoNet. Although Huang et al ’s model achieves lower temporal error quantitatively and is able to reproduce the color information and major textures...
in the style image, it fails to learn much about the perceptual strokes, visual patterns, and minor textures. As shown in the two examples in the figure, there are two reasons that may account for the weak perceptual styles in Huang et al’s model: First, they use a low weight ratio between perceptual losses and temporal loss to maintain temporal coherence, which brings visible reduction to the quality of output style. However, in ReCoNet, the introduction of the new temporal losses makes it possible to maintain temporal consistency with a much larger perceptual to temporal losses ratio, leading to better preserved perceptual information. As shown in the first example, the stylized image generated by ReCoNet reproduces the distinct color blocks in the Composition style much better than Huang et al’s result, especially on the uneven sand surfaces and the sea wave. Second, Huang et al’s model chooses a deeper layer relu4_2 in the VGG loss network to compute the content loss, which is difficult to capture low-level features such as edges. In the second example, although sharp bold contours are characteristic in the Girl image, their model fails to clearly reproduce such style. Unlike Huang et al’s model, as shown in Figure 2 and 11(b), Chen et al’s work can well maintain the perceptual information of both the content image and the style image. However, noticeable inconsistency in their stylized results can be found as shown in the zoom-in regions, which can also be validated by its relatively high temporal errors in the previous quantitative comparison.
A user study has been conducted to further compare the video style transfer results of ReCoNet with these two models. We compared the stylization results of ReCoNet with Chen et al.’s model on style Candy, Edtaonisl (Francis Picabia, 1913), Tilt-shift, and Mosaic; and with Huang et al.’s model on style Composition, Udnie (Francis Picabia, 1913), Dream, and Oriental. For each comparison, we used a different video clip downloaded from Videvo.net [17], containing 80 to 200 frames with a frame rate of 24 FPS and a resolution of $640 \times 340$. 50 participants ranging from 18 to 35 years old were invited to conduct the user study. Participants must promise before answering the questions that they have no color vision deficiency including and not limited to color blindness and color amblyopia. They were asked three identical questions for each comparison:

1. Which video perceptually resembles the style image more? (regarding the color, strokes, textures, and other visual patterns)

2. Which video is more temporally consistent? (such as fewer flickering artifacts, consistent color and style of the same object)

3. Which video is preferable overall?

The first question was designed to compare the ability of video style transfer models to reproduce perceptual style. The second question was set to compare the ability of video style transfer models to maintain temporal consistency. The third question was asked in a subjective way to learn whether users generally
<table>
<thead>
<tr>
<th>Group 1, Style</th>
<th>Candy</th>
<th>Group 5, Style</th>
<th>Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>ReCoNet</td>
<td>64</td>
<td>162</td>
<td>152</td>
</tr>
<tr>
<td>Same</td>
<td>72</td>
<td>23</td>
<td>25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 2, Style</th>
<th>Edtaonisl</th>
<th>Group 6, Style</th>
<th>Udnie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>ReCoNet</td>
<td>6</td>
<td>44</td>
<td>37</td>
</tr>
<tr>
<td>Chen et al</td>
<td>32</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Same</td>
<td>12</td>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 3, Style</th>
<th>Tilt-shift</th>
<th>Group 7, Style</th>
<th>Dream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>ReCoNet</td>
<td>37</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>Chen et al</td>
<td>9</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Same</td>
<td>4</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group 4, Style</th>
<th>Mosaic</th>
<th>Group 8, Style</th>
<th>Oriental</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models</td>
<td>Q1</td>
<td>Q2</td>
<td>Q3</td>
</tr>
<tr>
<td>ReCoNet</td>
<td>14</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>Chen et al</td>
<td>8</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Same</td>
<td>28</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total</th>
<th>Models</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Models</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReCoNet</td>
<td>64</td>
<td>162</td>
<td>152</td>
<td>ReCoNet</td>
<td>164</td>
<td>42</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>Same</td>
<td>72</td>
<td>23</td>
<td>25</td>
<td>Same</td>
<td>14</td>
<td>67</td>
<td>43</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12: User study voting results

prefer our model or not. For each comparison, we put the style image, content video, and the two stylized videos using ReCoNet and one of the other two models in the same video window for easy comparison, and required participants to view them at least twice. A blind comparison is adopted, which means that participants were not told which model generated which video. They were given three voting options for each question: the first video is better; the second video
is better; the two models perform similarly, or it is hard to support one against the other (denoted as “same” in the result table). Figure 12 shows the detailed voting results. Compared with Chen et al ’s model, ReCoNet achieves much better temporal consistency while maintaining good perceptual styles. Compared with Huang et al ’s model, the style transfer results of ReCoNet are much better in perceptual styles and the overall feeling although its temporal consistency is slightly worse. This validates our previous qualitative analysis.

4.3 Ablation Study

The ablation study of ReCoNet mainly focuses on the two novel network designs: the multi-level temporal loss and the relative luminance warping constraint.

4.3.1 Temporal Loss on Different Levels

<table>
<thead>
<tr>
<th>Loss Levels</th>
<th>Alley-2</th>
<th>Ambush-5</th>
<th>Bandage-2</th>
<th>Market-6</th>
<th>Temple-2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature-map only</td>
<td>0.1028</td>
<td>0.1041</td>
<td>0.0752</td>
<td>0.1062</td>
<td>0.0991</td>
<td>0.0975</td>
</tr>
<tr>
<td>Output only</td>
<td>0.0854</td>
<td>0.0840</td>
<td>0.0672</td>
<td>0.0868</td>
<td>0.0820</td>
<td>0.0813</td>
</tr>
<tr>
<td>Both</td>
<td>0.0846</td>
<td>0.0819</td>
<td>0.0662</td>
<td>0.0862</td>
<td>0.0831</td>
<td>0.0804</td>
</tr>
</tbody>
</table>

Table 3: Temporal error $e_{stab}$ with style Candy for different temporal loss settings in ReCoNet. Five scenes from MPI Sintel Dataset are selected for validation.

To study whether the multi-level temporal loss does help suppress temporal inconsistency and better maintain the perceptual styles, we implement ReCoNet on
Figure 13: Temporal inconsistency in traceable objects. (a) The style target and two consecutive frames in MPI Sintel Dataset. (b) Stylized frames generated without feature-map-level temporal loss. (c) Stylized frames generated with feature-map-level temporal loss. A specific traceable region is circled for loss comparison.

Candy style with three different temporal loss settings: feature-map-level temporal loss only, output-level temporal loss only, and feature-map-level temporal loss plus output-level temporal loss.

Table 3 shows the temporal error $e_{stab}$ of these settings on five scenes in MPI Sintel Dataset. As the results shows, the temporal error is greatly reduced with the output-level temporal loss, while the feature-map-level temporal loss also
improves temporal consistency on average. Figure 13 also demonstrates a visual example of object appearance inconsistency without feature-map-level temporal loss. When only using output-level temporal loss, the exactly same object may alter its color subject to changes in surrounding environment. With the feature-map-level temporal loss, features are better preserved for the same object, leading to a more temporally coherent stylized output.

4.3.2 Luminance Difference

To validate our choice of the relative luminance warping constraint in the output-level temporal loss, we compare three different approaches taking or not taking luminance difference into consideration at the output level:

1. A relative luminance warping constraint on each RGB channel (Formula 5):

$$L_{temp}^o = \frac{1}{D} M_t (\|(O_t - W_t(O_{t-1}))_Y - (I_t - W_t(I_{t-1}))_Y\|_2 + \|(O_t - W_t(O_{t-1}))_{X,Z}\|_2)$$

where $X, Y, Z$ are the XYZ channels;

2. Change color space of output-level temporal loss into XYZ color space, then add a relative luminance warping constraint to Y channel: $L_{temp}^o = \frac{1}{D} M_t (\|(O_t - W_t(O_{t-1}))_Y - (I_t - W_t(I_{t-1}))_Y\|_2 + \|(O_t - W_t(O_{t-1}))_{X,Z}\|_2)$

3. No luminance constraint: $L_{temp}^o = \frac{1}{D} M_t \|(O_t - W_t(O_{t-1}))_{R,G,B}\|_2$. 

44
Figure 14: Style transfer results using three different approaches described in Section 4.3.2 to target luminance difference. The style target is Candy, and the validation scenes are same as Table 3 for temporal error $e_{\text{stab}}$ calculation. The luminance-wise and the total temporal error maps show the absolute value of temporal errors in the relative luminance channel and in all color channels respectively.

Other variables in approach 2 and 3 are identical to those in Formula 5. As demonstrated in Figure 14, all of the three approaches can reproduce pleasant perceptual styles of Candy despite some variations in coloring. However, the first approach has a more similar luminance-wise temporal error map to the input frames compared with the other two approaches, especially at the circled illuminated region in the figure. This shows the first approach can better preserve luminance changes between consecutive frames as those in the input, and
Therefore results in more natural stylizing outputs. Moreover, the total temporal error map of the first approach is also closer to zero than the results of other two approaches, which implies that the first approach is able to generate more stable stylized results. This finding is also supported numerically by a much lower overall temporal error produced by the first approach in the validation scenes. Based on both qualitative and quantitative analysis, we can conclude that the first approach - adding a relative luminance warping constraint to all RGB channels - can generate smoother luminance changes on areas with illumination effects and achieve better overall temporal coherence in the stylized videos.

5 Streaming Stylization Application

To demonstrate the ability of ReCoNet to stylize streaming videos, we have implemented a python application with graphical user interface (GUI) in PyQt [47] and opencv [48] plus the inference pipeline of ReCoNet.

Figure 15 shows the GUI of the application. The top row contains two streaming windows, left for displaying the original video input and right for displaying the corresponding stylization results. The FPS for inferencing (one divided by the time duration for a video frame to be processed by ReCoNet) and displaying (one divided by time duration from a video frame is captured by the web-cam
to both the video frame and its stylization result is displayed on the GUI) is also displayed over the streaming windows. The bottom row contains multiple style target images for users to choose. Once clicked, the application will first load the corresponding pre-trained ReCoNet model, then begin to stylize input video frames and display them at the stylization streaming window.

![Streaming Video Here](image1.jpg) ![Stylized Streaming Video Here](image2.jpg)

Figure 15: The GUI of the streaming stylization application

Using the same inference pipeline and hardware, the inference FPS of the application could reach around 180 FPS, which is slightly lower than the quantitative testing result in Section 4.2.1. This is because preloaded datasets are used during the quantitative testing, and therefore it will take less time for the GPU to load and update image tensors. The display FPS of the application also depends on
the frame rate of the web-cam, the speed of data transmission, and the speed of rendering images on the qt widget. For an around 30 FPS web-cam and a modern desktop computer, the display FPS is usually around 20 FPS. Note that in the application, a video frame is fetched from the web-cam only when the previous frame has been fully processed and displayed in the GUI. Some speed-up techniques may also be implemented to improve the display FPS such as using threads and buffers to store video frames captured by the web-cam in full speed and preloading them to the GPU.

6 Conclusion

In this project, a novel feed-forward convolutional neural network ReCoNet is proposed for video style transfer. ReCoNet is able to generate temporally coherent stylized videos in real-time processing speed while maintaining artistic styles perceptually similar to the style targets. A novel luminance warping constraint in the output-level temporal loss and a novel feature-map level temporal loss are proposed in ReCoNet for better stylization stability under illumination effects and on traceable objects respectively. A streaming stylization application using ReCoNet has also been developed, which can display both the streaming video captured by one web camera and its stylization result at the same time, with multiple styles supported. In future work, the possibility of incorporating both
chromaticity and luminance difference in optical flow warping results for better video style transfer method can be further examined. Also, it also deserves more investigations on using smaller networks for coherent style transfer in real-time or near real-time speed on mobile devices without GPU.
Acknowledgement

I would like to acknowledge the great contributions from my group mate Mr. Derun Gu and my project co-operator Mr. Fangjun Zhang (who comes from EEE department and has worked with us under the same project). I also want to express my gratitude towards Mr. Ruochen Zhang, Dr. Dirk Schnieders, and Dr. Sameer Singh for offering facilities and making suggestions to the project. Last but not least, I and grateful and thankful that our supervisor, Professor Yizhou Yu, has supervised, guided and helped us in many aspects throughout this project. I sincerely appreciate all the efforts, favors, feedbacks and advices from them.
References


