TOUCH EVENT DETECTION AND TEXTURE ANALYSIS FOR VIDEO COMPRESSION

by

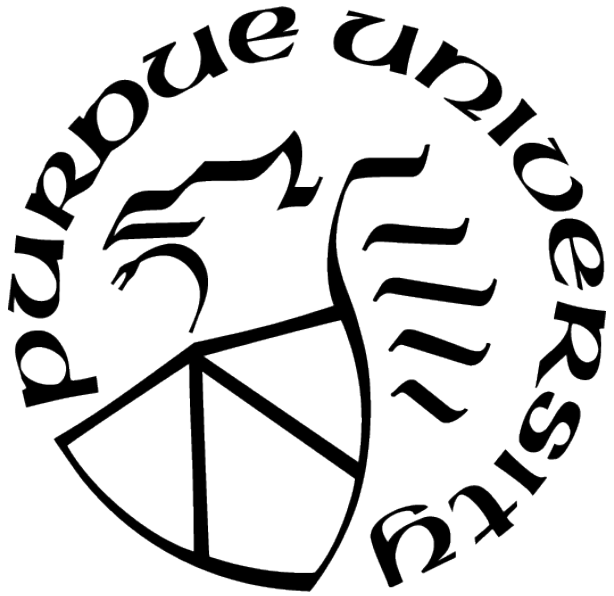
Qingshuang Chen

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



School of Electrical and Computer Engineering

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Fengqing Maggie Zhu, Chair

School of Electrical and Computer Engineering

Dr. Stanley Chan

School of Electrical and Computer Engineering

Dr. Edward J. Delp

School of Electrical and Computer Engineering

Dr. Amy Reibman

School of Electrical and Computer Engineering

Approved by:

Dr. Dimitrios Peroulis

This thesis is dedicated to my supportive husband and my parents.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude toward my major professor, Professor Fengqing Maggie Zhu. Professor Zhu has always provided me with many suggestions on my research and specific tasks. My Ph.D. study could not be fulfilled without her efforts. I especially thank her for her patience and effort going through many revisions of my manuscripts. I would also like to express my sincere gratitude toward Professor Edward J. Delp. Professor Delp has provided me the opportunity to work in the Video and Image Processing Laboratory (VIPER lab) and his guidance for me to become an independent researcher. I value his advises very much and I have learned many valuable insights that can not be easily learned elsewhere.

I am also grateful to my committee members, Professor Amy Reibman and Professor Stanly Chan for their advice to complete my study, and for their efforts and time that put on my research and thesis. I would like to thank Professor Amanda Seidl for the research opportunity of touch event detection project.

I would like to thank Di Chen for being a great teammate for video codec project. I would like to thank Rana Abu-Zhaya for providing with the background knowledge of touch detection project, and supporting me with the experiment data. I would also like to thank for my former roommate, Ruiting Shao, I am grateful for her friendship and support during my Ph.D. study and life. I would like to extend my gratitude to all my former and current VIPER lab members for their friendships and supports: Dr. Albert Parra Pozo, Dr. Neeraj J. Gadgil, Dr. Khalid Tahboub,Dr. Joonsoo Kim, Dr. Yu Wang, Blanca, Dr. Chichen Fu, Dr. Shaobo Fang, Dr. Javier Ribera Prat, Dr. David Joon Ho, Dr. Jeehyun Choe, Dr. Dahjung Chung, Dr. Yuhao Chen, Dr. David Gera, Dr.Soonam Lee, Dr. Daniel Mas Montserrat,Dr. Shuo Han, Dr. Sri Kalyan Yarlagadda, Blanca Delgado, Jiaju Yue, He Li, Chang Liu, Sriram Baireddy, Enyu Cai, Alain Chen, Jiaqi Guo, Hanxiang (Hans) Hao, Jiangpeng He, Janos Horvth, Han Hu, Runyu Mao, Zeman Shao, Changye Yang, Yifan Zhao, Justin Yang.

I would like to thank my husband, Jiaju Yue. He supports me, encourages me and backs me up. He always clean up my way to let me focus on my study. I am grateful for his love and supports.

Last, but not least, I would like to express my sincere appreciation to my father Yunxiao Chen and my mother Xuejian Song. They encouraged me to explore new opportunities in life and career and supported me for my whole life. This journey would not have been possible without their endless love and supports.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

p	pixel
f	frame
fps	frame per second
w	weight
c	score

ABBREVIATIONS

ML	Machine Learning
DNN	Deep Neural Network
CNN	Convolutional Neural Network
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
UGC	User Generated Content
AV1	AOMedia Video 1
H.264/AVC	H.264/Advanced Video Coding
H.265/HEVC	H.265/High-Efficiency Video Coding
GOP	Group of Pictures
MS-SSIM	Multiscale SSIM
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
QP	Quantizatin Parameter
QoE	Quality of Experience
SSIM	Structural Similarity Index
UEQ	UnEqual Quality
VMAF	Video Multi-Method Assessment Fusion
AOM	Alliance of Open Media
PM	Pixel-level method
BM	Block-level method
PMR	Pixel-level method with refinement

ABSTRACT

Touch event detection investigates the interaction between two people from video recordings. We are interested in a particular type of interaction which occurs between a caregiver and an infant, as touch is a key social and emotional signal used by caregivers when interacting with their children. We propose an automatic touch event detection and recognition method to determine the potential timing when the caregiver touches the infant, and classify the event into six touch types based on which body part of the infant has been touched. We leverage deep learning based human pose estimation and person segmentation to analyze the spatial relationship between the caregivers' hands and the infant. We demonstrate promising performance on touch event detection and classification, showing great potential for reducing human effort when generating groundtruth annotation.

Recently, artificial intelligence powered techniques have shown great potential to increase the efficiency of video compression. In this thesis, we describe a texture analysis pre-processing method that leverages deep learning based scene understanding to extract semantic areas for the improvement of subsequent video coder. Our proposed method generates a pixel-level texture mask by combining the semantic segmentation with simple post-processing strategy. Our approach is integrated into a switchable texture-based video coding method. We demonstrate that for many standard and user generated test sequences, the proposed method achieves significant data rate reduction without noticeable visual artifacts.

1. INTRODUCTION

1.1 Touch Event Detection

Touch is a key social and emotional signal used by caregivers when interacting with their children [1]–[13]. Touch is present in an enormous amount of caregiver-infant interactions and its presence has been found to impact infants' attention, arousal levels, behavioral, and emotional states [7], [8], [14], [15], as well as to reduce infants' stress [2].

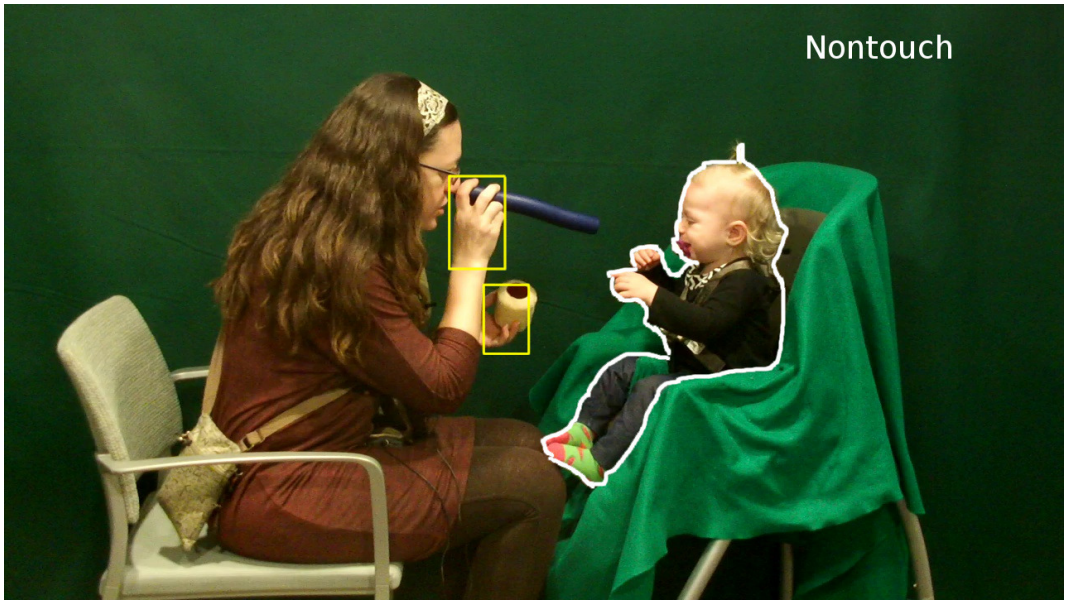
The benefits of touch on human development extend to special populations as well. Low birthweight infants who receive more nurturing touch from their mothers were found to have higher scores for the security of their attachments than infants who receive less touch[16]. This can impact physiology. For example, compared to infants who did not receive any specific therapy, low birthweight infants who received touch therapy in the form of "Kangaroo Care" (skin-to-skin contact between the mother and the infant) were more socially alert, and had higher scores on measures of cognitive and motor development at 6 months of age [17]. Moreover, low birthweight infants who received more nurturing touch from their mothers were less likely to show depression and anxiety at 2 years of age [18]. The positive impact of caregiver touch on low birthweight and preterm infants extends even further into development. Studies that extend beyond infancy into childhood show that early human tactile contact leads to lower stress, better sleep, more cognitive control, and a better mother-child relationship [19]. Furthermore, the use of touch therapies such as "Kangaroo care" have a positive impact not only on the development of preterm low birthweight infants, but also on their mothers' well-being and their sensitivity towards their infants [17].

Recent work suggests that touch may not only be generally helpful to infant development by fostering attachment [1], [16] and mother-child reciprocity [19], but may also be specifically helpful to some crucial tasks in development, namely learning the input language. Speech to infants (and to adults) does not contain clear reliable cues to word boundaries [20]. Thus, in order to learn the mapping between words and their referents the infant must first find words in the continuous flow of human speech and remember these word units. Data from a variety of paradigms suggests that infants do not begin to segment the speech stream until 6 to 7 months of age, and even then their abilities are still quite tenuous [21]. However, recent

data suggests that touch may be helpful for very young infants in the task of segmenting the speech stream into words [22]. In this study, 4.5-month-old who received consistent touch cues from an experimenter were better able to segment the speech they were familiarized with than infants who did not receive such tactile stimulation [22]. This suggests a role for touch in the formation of the protolexicon the lexicon of things that sound like words that infants compose before they form a true lexicon (mental dictionary) –that is adult-like in nature.

However, touch would not be useful to language-acquiring children, if they did not receive it from their caregivers in real life interactions in a way that would aid in language acquisition. Recent work suggests that caregivers do in fact provide their infants with touches that are informative both about the beginnings and ends of words in continuous speech and also about the meanings of words, at least in certain contexts. Specifically, Abu-Zhaya, Seidl, and Cristia [23] recorded caregivers interacting with their infants in a book-reading situation and found that caregiver touch is used in a way that might be helpful to two crucial language learning tasks: segmenting the speech stream into words and mapping words to their referents. They found that when interacting with their infants, mothers naturally accompanied their speech with touch cues. Moreover, most of the touch cues (73 percent) that mothers provided to their infants were accompanied with speech. These touches that were accompanied by speech were found to be longer and had twice as many beats as the touches that were not accompanied by speech. An examination of the speech that was accompanied by touch compared to the speech that was not accompanied by touch, and more specifically body-part and animal words, revealed that words that were spoken with touch had a significantly higher fundamental frequency than words that were not accompanied by touch. Moreover, most mothers' touches were found to be temporally aligned with their speech such that the onsets/offsets of touches and the onsets/offsets of words were often aligned providing potential multimodal word segmentation cues. Further, the majority of the touches produced during the production of body-part words were found to be semantically congruent, i.e., when mothers talked about their infants' body-parts, they were very likely to also touch those same body parts that were evoked by their speech. For example, the researchers found that a mother was more like to say the word "foot" when touching her

infant's foot than another object or body part. Thus, this congruency could be helpful to the infant in forming her word-to-world mappings in the wild. Figure 2.20 is an example of such touch events.



(a)

(b)

Figure 1.1. (a) An example without touch event, (b) An example where a potential touch event has occurred.

Recently, researchers examining the use of touch in mother-infant interactions have employed a micro-genetic approach by frame-by-frame annotation of touch cues yielding a detailed examination of maternal touches during different types of interactions [23]. Using this micro-genetic approach to examine videotaped human interactions, and annotating touch events in this detailed fashion allows researchers to examine the richness of mother-infant interactions and the way language is presented to the infant in alignment with various other cues.

However this micro-genetic coding is extremely time consuming. For example, Abu-Zhaya et al. [23]. used ELAN [24] to annotate the touch events. Using their detailed coding, it could take as much as 15 hours to annotate a series of 120 touch events in a 5 minute caregiver-infant interaction. This was so time consuming because they developed a scheme according to which each touch event is annotated on three different tiers marking the duration of the touch event, its location, its type and the number of beats that are involved in that touch event. Their videos were annotated by teams of two research assistants who watched the videos frame by frame and annotated each touch event as a team. Both research assistants had to reach consensus on the different components of the touch event before the annotation was complete. Not only is annotating these video interactions extremely time consuming, but research assistants have to be trained for several hours before they can even begin annotating the videos. Further, and in order to achieve accurate and consistent annotation of events, each member of each team has to be attentive and detail-oriented, and both members of each team have to be dedicated to working with one another in order to analyze the data properly.

Hence given the importance of human touch on human development it would be helpful for researchers studying human behavior to have a tool that can easily quantify both the quantity and quality of human touch that infants receive. Having an automatic system that is capable of detecting touch events would greatly reduce the amount of time spent on manually annotating these events, and would allow researchers to devise more efficient plans for data analyses.

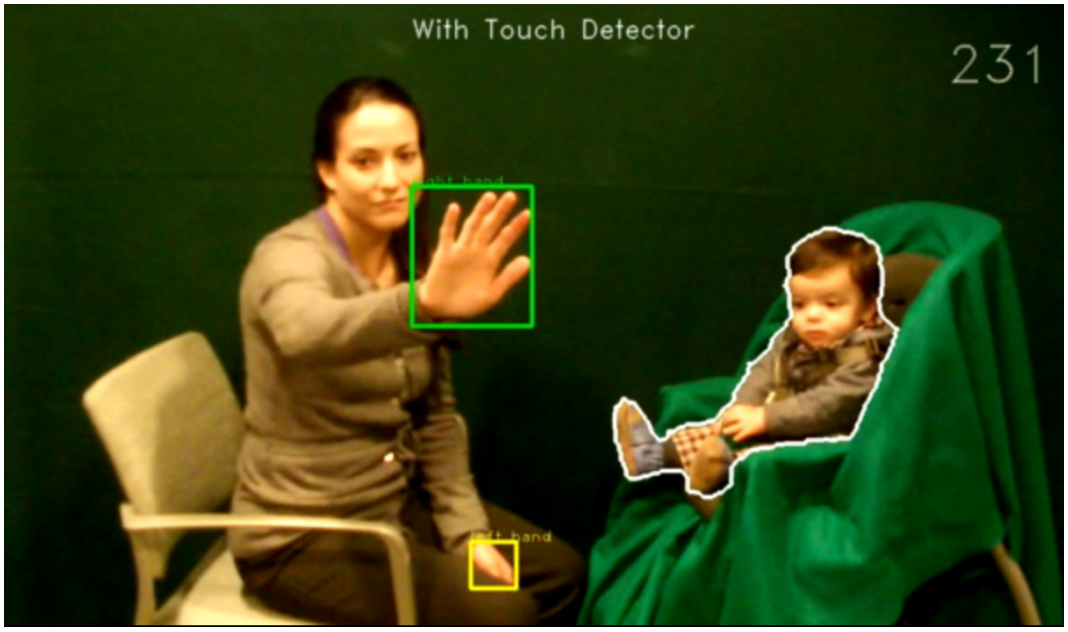
This tool could be broadly applied. Developing such a tool and easily quantifying caregiver touch to infants would be helpful for researchers interested in looking at the usefulness

of touch in language learning as well as to researchers interested in examining the frequency of touch provided to infants who are at risk for receiving lower levels of maternal touch. Further, the creation of such an automatic tool might be helpful for medical teams working with special populations and caregivers who have children with special needs. Given the enormous benefits that caregiver touch has on the development of low birthweight preterm infants [16]–[19], caregivers might benefit from a tool that monitors the amount of touch they are providing to their infants to help them achieve higher frequencies of touch in the Neonate Intensive Care Unit and beyond. Further, and due to findings showing that not only are the duration and frequency of touch important to child development, but also the specific functions of touch [16], it is necessary to identify the different functions of touch events – e.g. nurturing touch, soothing touch and aggressive touch, and alert caregivers on their touch pattern in order to promote better outcomes for their infants.

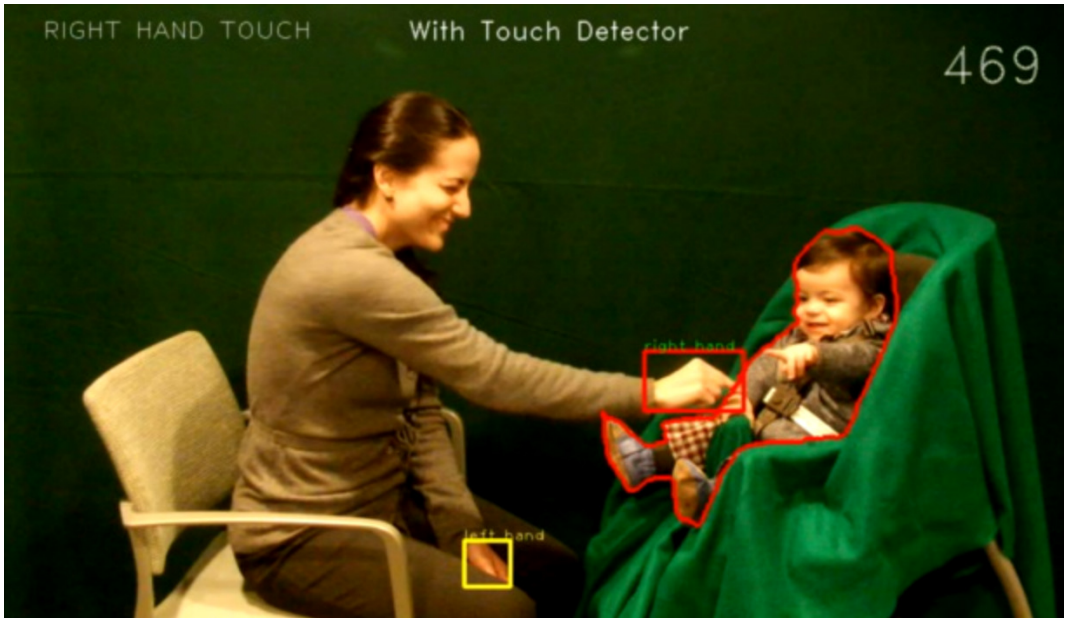
An automatic system that can detect the frequency and function of human touch can be broadly applicable to other situations in child development. Various methodological problems make it difficult for researchers to estimate and confirm the prevalence of child abuse in daycare and school settings [25]. However, methods like covert video surveillance have been effective in identifying serious physical and emotional violence that would not have been detected otherwise [26]; hence, developing an automatic system that can identify touch events quickly and classify them based on their locations, types and functions can enable a quicker and more accurate detection of incidents of child abuse. Moreover, such a system can be applied to an even wider range of human interactions detecting physical and sexual abuse in the workplace and in night-clubs.

In this thesis we are interested in detecting a particular interaction between the caregiver and the infant, namely the touch event, and detecting the moment when a touch occurs. A touch event is defined as the time when the infant is touched by the hands of the caregiver. Thus successfully tracking the hands of the caregiver and clearly detecting the outline of the infant are crucial in our touch event detection. Essentially a touch will occur when the segmented contours of the hands and the infant merge. An example of this is shown in Figure 1.2. We present two methods to track the position of the hands using a tracking based method and a learning based method. We propose two Grab-Cut [27] based infant

segmentation methods to obtain the contour of the infant. We propose a touch detection method by analyzing the merging contours of the caregivers hands and the infants contour. We also propose an improved automatic touch event recognition method to determine the potential time interval when the caregiver touches the infant. In addition to label the touch events, we also classify them into six touch types based on which body part of infant has been touched. CNN based human pose estimation and person segmentation are used to analyze the spatial relationship between the caregivers hands and the infants. We demonstrate promising results for touch detection and show great potential of reducing human effort in manually generating precise touch annotations.



(a)



(b)

Figure 1.2. (a) An example of a frame without a touch event, (b) An example where a touch event has occurred by detecting merging contours.

1.2 Introduction for Texture Analysis in Video Compression

In recent years, Internet traffic has been dominated by a wide range of applications involving video, including video on demand (VOD), live streaming, ultra-low latency real-time communications, *etc.* With ever increasing demands in resolution (*e.g.*, 4K, 8K, gigapixel [28], high speed [29]), and fidelity, (*e.g.*, high dynamic range [30], and higher bit precision or bit depth [31]), more efficient video compression is imperative for content transmission and storage, by which networked video services can be successfully deployed. Fundamentally, video compression systems devise appropriate algorithms to minimize the end-to-end reconstruction distortion (or maximize the quality of experience (QoE)), under a given bit rate budget. This is a classical rate-distortion (R-D) optimization problem. In the past, the majority of effort had been focused on the development and standardization of video coding tools for optimized R-D performance, such as the intra/inter prediction, transform, entropy coding, *etc.*, resulting in a number of popular standards and recommendation specifications (*e.g.*, ISO/IEC MPEG series [32]–[38], ITU-T H.26x series [36]–[40], AVS series [41]–[43], as well as the AV1 [44], [45] from the Alliance of Open Media (AOM)[46]). All these standards have been widely deployed in the market and enabled advanced and high-performing services to both enterprises and consumers. They have been adopted to cover all major video scenarios from VOD, to live streaming, to ultra-low latency interactive real-time communications, used for applications such as telemedicine, distance learning, video conferencing, broadcasting, e-commerce, online gaming, short video platforms, *etc.* Meanwhile, the system R-D efficiency can also be improved from pre-processing and post-processing, individually and jointly, for content adaptive encoding (CAE). Notable examples include saliency detection for subsequent region-wise quantization control and adaptive filters to alleviate compression distortions [47]–[49].

Modern video codecs such as HEVC [37] and AV1 [50] use hybrid coding techniques consisting of motion compensation and 2D transform to remove spatial and temporal redundancy. However, efficient exploitation of statistical dependencies measured by a mean squared error (MSE) does not always produce the best psychovisual result. Some regions in the frame, e.g., texture, are "perceptually insignificant" where an observer does not notice

any difference without observing the original video sequence, but are costly to encode. Texture based approaches have been shown to improve the coding efficiency for "perceptually insignificant" regions [51]–[54]. We explored similar ideas in our previous work [55], where we do not encode the texture regions, but instead these regions are reconstructed at the decoder based on a motion model. A Convolutioanl Neural Networks (CNN) based texture analyzer was developed to identify the texture regions in a frame and generates block-based texture masks. The displacement of the entire texture region is modeled by a set of motion parameters. At the decoder, instead of performing motion compensation prediction to reconstruct blocks in the texture region, the texture blocks are warped from the reference frames towards the current frame using the motion parameters.

While the proposed approach in [55] can achieve a data rate saving of 1% to 13% compared to the baseline when implemented using AV1 with satisfactory visual quality, the block-based texture masks cannot always accurately represent the texture regions. The block-based texture masks can be seamlessly integrated into AV1 since the common coding units are blocks. However, it can sometimes cause noticeable visual artifacts when an identified texture block consist of small structural region. In addition, the smallest texture block size in [55] was 32×32 in order to avoid detecting small moving objects, but at the same time limits the size of identified texture regions and reduces potential data rate savings. To illustrate this, an example is shown in Figure 1.3 where part of the bow of the white boat and the person's head are identified as texture region in [55] since the majority of that block is texture. The bow of the white boat and the person's head show flickering artifacts since they have different motion trajectory than the river. There is also some texture regions in the river not identified due to the large block size used in the texture analyzer. The methods proposed in this paper address both of these issues.

In this thesis, we introduce a modified switchable texture mode in AV1 and show that the proposed method can achieve significant data rate reductions with improved visual quality. Our previous work [55] using block-based texture analyzer has shown data rate savings in texture regions. However, block-based texture mask cannot accurately represents texture regions and may cause coding artifacts. Therefore, in this thesis, a pixel-level texture mask generation is described in this section to obtain more accurate texture masks. We

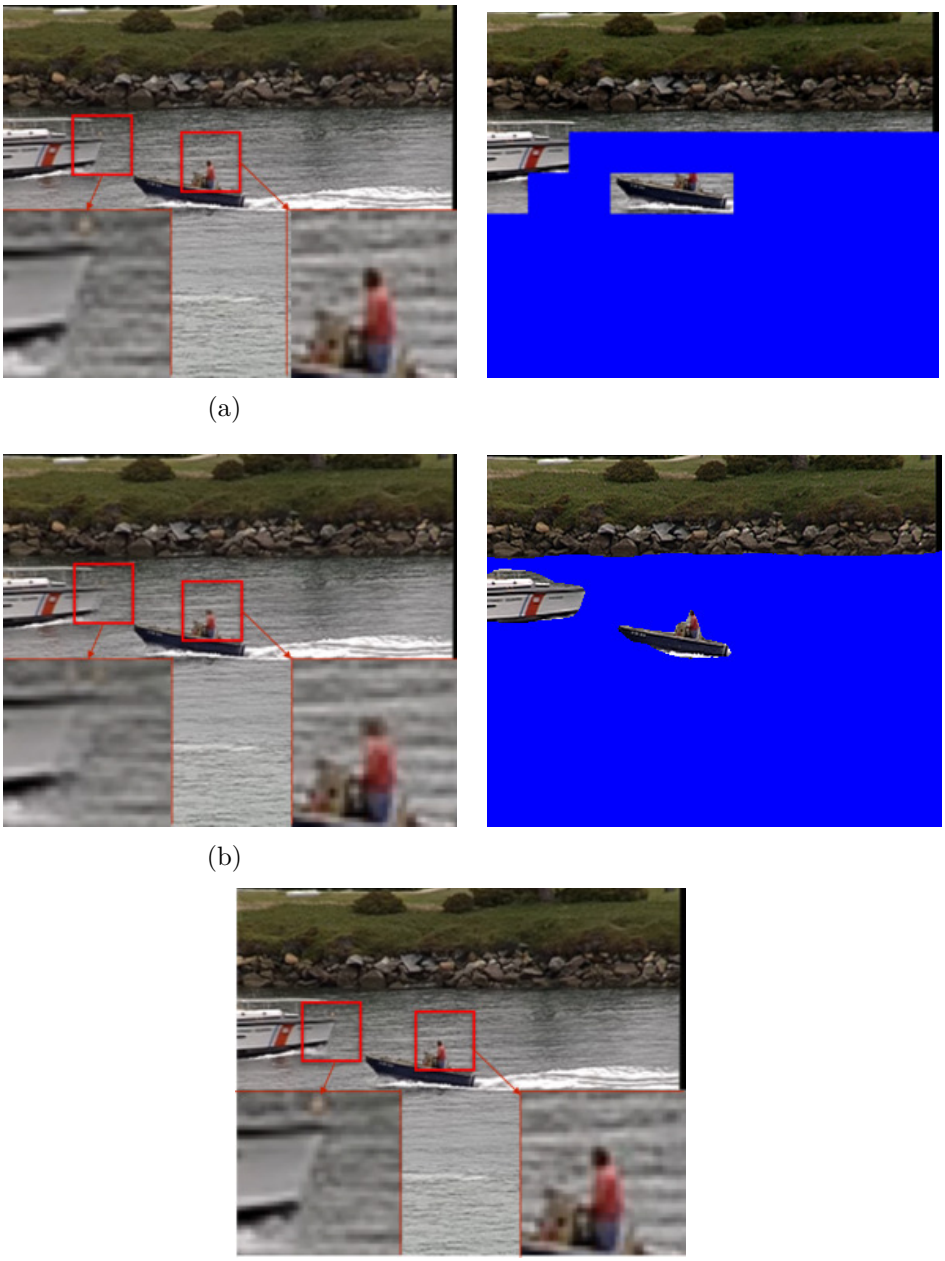


Figure 1.3. (a) is the reconstructed frame by AV1 original codec. (b) is the block-based mask. (c) is the reconstructed frame using block-based mask. (d) is the pixel-level mask. (e) is the reconstructed frame using pixel-level mask.

incorporate semantic scene segmentation into video compression by generating pixel-level texture segmentation masks to represent "perceptually insignificant" regions in a frame and use motion models to reconstruct the texture regions at the decoder to improve the coding efficiency. First, we use a semantic scene segmentation method described in Section 3.3.1 to generate masks for different semantic classes. Then in Section 3.3.2, we describe how several semantic classes with similar texture are grouped into four texture classes to produce a single pixel-level segmentation mask for each texture class.

1.3 Thesis Statement and Contributions

Touch Event Detection

In this thesis, we propose an automatic touch event detection and recognition method to determine the potential timing when the caregiver touches the infant, and classify the event into six touch types based on which body part of the infant has been touched. We leverage deep learning based human pose estimation and person segmentation to analyze the spatial relationship between the caregivers' hands and the infant. We demonstrate promising performance on touch event detection and classification, showing great potential for reducing human effort when generating groundtruth annotation.

The challenges of this project includes:

1. The dataset is collected in a challenging scenario where the caregiver is mostly in the profile view. Also as the caregiver is free to interact with the infant, hand occlusion and hand vanishing are likely to happen.

2. Some touch events are very difficult to distinguished even for human. The proposed method aims at rejecting non-touch frames with high confidence and narrow down the potential touch frames for human annotator to refine.

The main contributions of this thesis are listed as follows:

1. We propose a 2D hand tracking method using color and motion features with occlusion handling.

2. We propose a long-term 2D hand tracking method with re-initialization capability using a hand detection network and a human pose estimation network.

3. We propose a challenging hand tracking dataset with annotations for long sequences in which the subjects are mostly in profile view. The dataset also contains difficult scenarios such as hand interaction, hand occlusions and hand vanishing.

4. We propose an automatic touch event detection and recognition system to determine the potential timing when the caregiver touches the infant, and classify the event into six touch types based on which body part of the infant has been touched.

Texture Analysis for Video Compression

In this thesis, we describe a texture analysis pre-processing method that leverages deep learning based scene understanding to extract semantic areas for the improvement of subsequent video coder. Our proposed method generates a pixel-level texture mask by combining the semantic segmentation with simple post-processing strategy. Our approach is integrated into a switchable texture-based video coding method and implementing using the AV1 codec [44]. We demonstrate that for many standard and user generated test sequences, the proposed method achieves significant data rate reduction without noticeable visual artifacts.

The challenges of this project includes:

1. Lack of annotated texture datasets. Most texture related datasets contain still images with pure texture. Also, there is no existing image/video datasets with pixel-level annotated for texture regions.

2. The accuracy of the texture analysis is important to video compression performance. False positive in texture analysis will lead to compression artifacts, and false negative in texture analysis lead to inefficient compression.

3. Lack of proper visual quality metric to analyze the compression results objectively.

The main contributions of this thesis are listed as follows:

1. We combine semantic segmentation with optional post-processing steps to generate pixel-level segmentation mask for texture regions in a video frame.

2. The proposed texture analysis method is integrated into a switchable texture-based video coding method.

3. We show that for many standard test sets and user generated test sequences, the proposed method achieves significant data rate reductions with improved visual quality.

1.4 Publications Resulting From This Thesis

Conferences:

1. **Q. Chen**, F. Zhu, "CNN based touch interaction detection for infant speech development", *IEEE International Conference on Multimedia Information Processing and Retrieval*, pp.20-25, 2019.

2. **Q. Chen**, F. Zhu, "Long Term Hand Tracking With Proposal Selection," *IEEE International Conference on Multimedia and Expo Workshops*, pp. 1-6, 2018.

3. **Q. Chen**, H. Li, R. Abu-Zhaya, A. Seidl, F. Zhu, E. J. Delp, "Touch event recognition for human interaction", *Electronic Imaging*, pp. 1-6, 2016.

4. D. Chen, **Q. Chen**, F. Zhu, "Pixel-level texture segmentation based AV1 video compression", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.1622-1626, 2019.

Journal:

1. D. Ding, Z. Ma, D. Chen, **Q. Chen**, Z. Liu, F. Zhu, "Advances in video compression system using deep neural network: a review and case studies", *Proceedings of the IEEE*,pp. 1-27, 2021.

2. TOUCH EVENT DETECTION

2.1 Overview

There is a growing need to understand the content of videos, such as human action recognition. Using automated methods to analyze video contents are of great interest due to the high expense and intense labor required to perform these tasks manually. Video action recognition datasets like UCF101 [56], Hollywood2 [57] have enabled improved performance for these tasks. These datasets [56], [57] consist of small video clips and each video clip has a label for type of actions. A considerable amount of literature has been published on classifying actions in video clips and assign labels to each video clip [58]–[60]. However, recognizing pairwise human interaction and making frame level decision is still an open problem.

In this thesis, we are interested in a particular type of human interaction known as touch, as touch is a key social and emotional signal used by caregivers when interacting with their children. Touch event is defined as the time when the hands of the caregiver has physical contact with infant in the context of our work. Essentially, a touch will occur when the segmented regions of the hands and the infant overlap. A touch event is further categorized into one of the six touch types (head, arm, hand, torso, leg, foot) based on which body part of infant has been touched. Thus, successfully tracking the hands of the caregiver and clearly detecting the outline of the infant are crucial in our touch event detection.

Our method for touch detection contains three parts, the hand tracker of the caregiver, the infant's contour detector and the touch event detector. Figure 2.1 shows block diagram of the analysis system.

Two method of hand tracking are described in Section 3. A tracking based method is described in Section 3.2 and a learning based method using deep neural networks is described in Section 3.3. The hand tracking method in Section 3.2 takes advantage of analyzing the motion of the hand inspired from [61], and deals with occlusions occurred while tracking two hands. Section 3.3 extends the idea of using hand detection and the locations of body joints to refine tracking results. A hand model is trained using the Faster-RCNN [62].

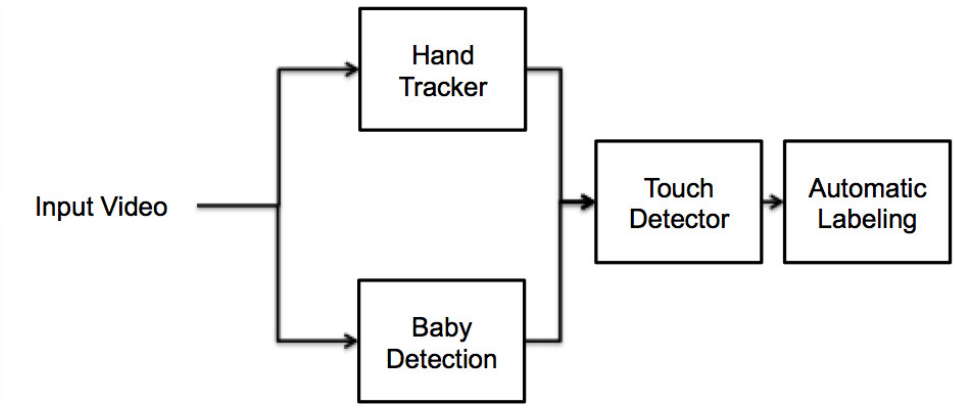


Figure 2.1. Block diagram of our touch event detection system.

The motion and the localization of the human hand is important in characterizing human actions in dynamic scenes. Hand tracking is used in this thesis to obtain the spatial information of caregiver's hands with respect to the location of the infant. Hand tracking is very challenging due to large variation in appearance and movement compared to other body parts. This chapter represents two hand localization method, a tracking based method is described in Section 3.2 and a learning based method using deep neural networks is described in Section 3.3. The hand tracking method in Section 3.2 takes advantage of analyzing the motion of the hand inspired from [61], and deals with occlusions occurred while tracking two hands. Section 3.3 extends the idea of using hand detection and the locations of body joints to refine tracking results. A hand model is trained using the Faster-RCNN [62]. We use a heatmap based method called stacked hourglass network [63] to predict the keypoints of human body. The hand position in the tracker is updated and corrected if need to based on hand detection and body keypoints results.

2.2 Hand Tracking with Occlusion Handling

Hand tracking is an active and open research problem as it plays an important role in other vision tasks including human computer interaction [64], human action recognition

[65] and sign language recognition [66]. Due to the high number of degrees of freedom in a hand and high occlusion of hands with other objects, many researchers rely on other sensors such as depth cue [67], [68] or use meanings, e.g., wearing colored glove on the hand [69], to obtain accurate hand models. However, for many other applications, there exists only visible imaging data. There has been much work in 2D hand tracking based on videos captured from RGB cameras. In [61], a Camshift [70] method is described that reduces tracking failures when the hand moves across other large skin-like areas by classifying velocities. It generally works when the velocity of the hand differs greatly from other skin-like regions and fails to track the hand when there are other skin regions moving along with the hand at similar speed. Hand detection [71] uses hand shape, context, skin colors and deformable part model [72] to detect the hand in static images. Another method [65] combines hand detection with hand tracking and uses an upper body model to refine the hand detection. Recent developments of hand detection in egocentric view have heightened the need for localizing hands in RGB videos captured from wearable devices [73] and in driving scenario [74]. Furthermore, tracking hand positions in long sequences is of special interest [75], [76].

The motion and the localization of the human hand is important in characterizing human actions in dynamic scenes. Hand tracking is used in this thesis to obtain the spatial information of caregiver's hands with respect to the location of the infant. Hand tracking is very challenging due to large variation in appearance and movement compared to other body parts. This chapter represents two hand localization method, a tracking based method is described in Section 3.2 and a learning based method using deep neural networks is described in Section 3.3. The hand tracking method in Section 3.2 takes advantage of analyzing the motion of the hand inspired from [61], and deals with occlusions occurred while tracking two hands. Section 3.3 extends the idea of using hand detection and the locations of body joints to refine tracking results. A hand model is trained using the Faster-RCNN [62]. We use a heatmap based method called stacked hourglass network [63] to predict the keypoints of human body. The hand position in the tracker is updated and corrected if need to based on hand detection and body keypoints results.

The hand tracker uses color and motion features to track the position of both hands in each frame. The hand positions are initialized by manually selecting two bounding boxes containing each hand of the caregiver respectively in the first frame. Each bounding box is considered as a separate independent tracker. Figure 2.2 is the block diagram for the proposed method.

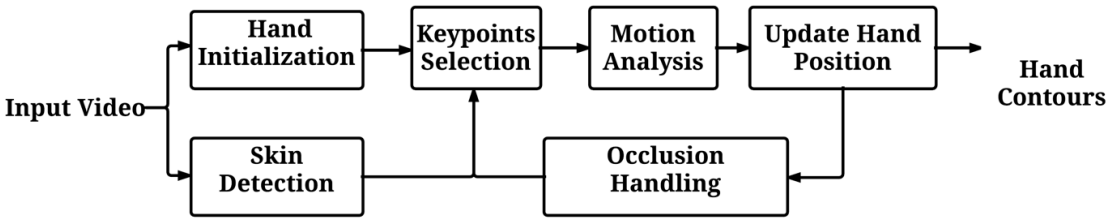


Figure 2.2. Block Diagram of the Proposed Hand Tracking Method.

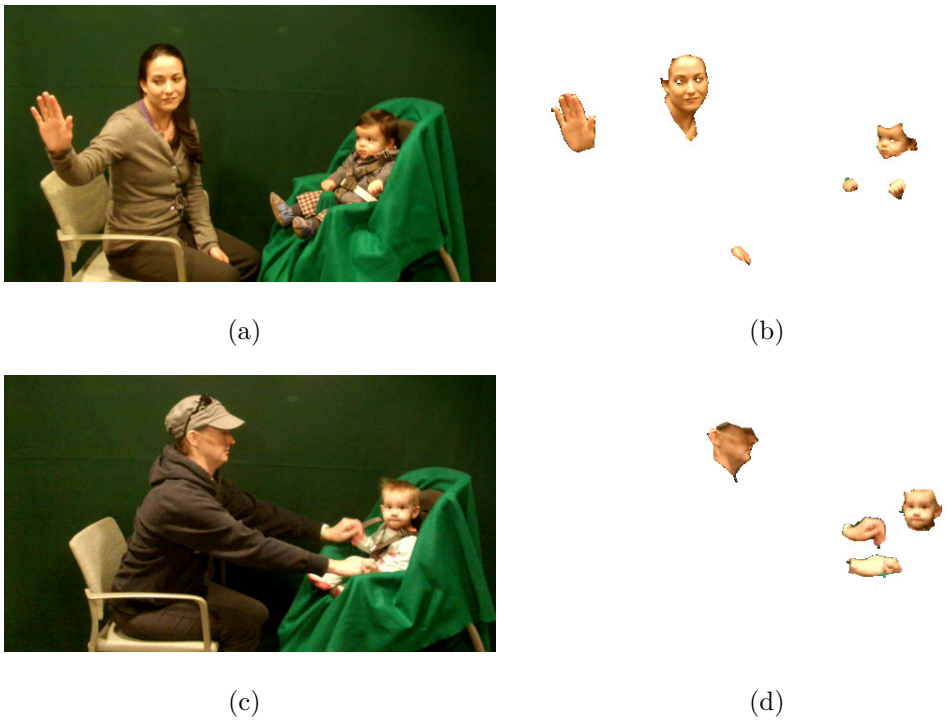
2.2.1 Skin Detection

Color is an important feature in many tracking scenarios, because the color differences of the interested objects usually within small variances. Thus color cue provides relevance information about the target in consecutive frames. Especially for handing tracking, the color of hands are likely to be human skin color. Skin detection is used to obtain the skin regions the frame, and we assume that the interested points to track on hand have similar color as the skin.

A pixel-based skin detection method is used to obtain the skin mask of the hands [77], which classify each pixel as skin or non-skin individually. [77] suggested to use the linear quantization on each histogram since too many color bins lead to over-fitting while too few bins results in poor accuracy. For the color model, each RGB color space is quantized from the original 256 bins to 16 bins and is mapped into 1D 16^3-bin histogram. The sum of this histogram is then normalized to one.

Since the lighting condition in our dataset is much brighter than existing public skin datasets [78], [79], we use our own data to train the skin color model. The skin color model

is trained using 240 frames from videos with different pairs of caregivers and infants. All skin regions are manually segmented as ground truth skin pixels in every training frame, the remaining pixels are treated as non-skin pixels. Figure 2.3 shows examples of skin mask ground truth in our dataset.



(a) (b)

(c) (d)

Figure 2.3. Examples skin masks ground truth in our dataset, (a) and (c) are the original images, (b) and (d) are the corresponding skin mask ground truth.

The probability of the skin class and non-skin class are defined as follows[77]:

$$P(RGB|skin) = \frac{s(RGB)}{T_s} \tag{2.1}$$

$$P(RGB|nonskin) = \frac{n(RGB)}{T_n} \tag{2.2}$$

where $s(RGB)$ represents the number of skin pixels in the histogram and $n(RGB)$ represents the non-skin pixel counts in the histogram. T_s and T_n are the total counts contained in the

skin and non-skin histograms, respectively. A color pixel is considered as skin pixel when it satisfies:

$$\frac{P(RGB|skin)}{P(RGB|nonskin)} \geqslant \Theta \tag{2.3}$$

where

$$\Theta = C \cdot \frac{T_n}{T_s} \tag{2.4}$$

Θ is a threshold which can be adjusted for trade-off between correct detections and false positives. C is an adaptive parameter. The value of C varies with different skin tones or lighting conditions.

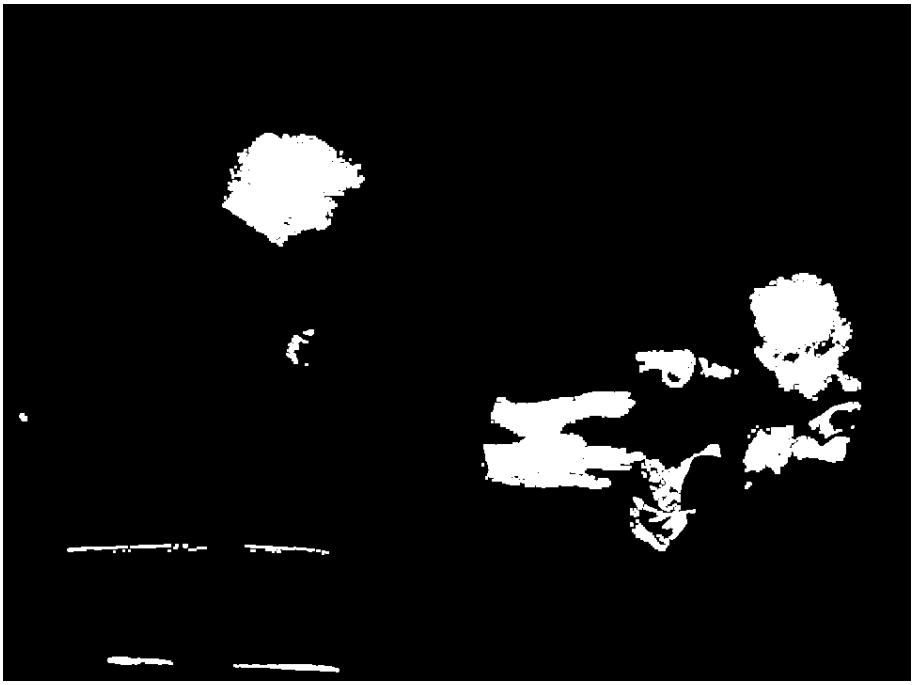
We divide 240 images to four subsets and 60 image each, then perform a four-fold cross validation on the performance of the skin detection. Table 2.1 shows the cross validation results. And we use the second color model with the highest precision score in our system.

Table 2.1. Skin Detection Results

Training Set	Testing Set	Precision	Recall
Subsets 1,2,3	4	89.04%	68.22%
Subsets 1,2,4	3	90.80%	71.37%
Subsets 1,3,4	2	85.01%	74.55%
Subsets 2,3,4	1	86.70%	77.66%

In our experiments described below we empirically choose C (and hence Θ) such that the detected skin regions compared favorably to the training data skin masks. We then used a morphological opening with window size 3 and a constant structuring element on the output of the pixel-based skin classifier to reduce isolated and small skin regions. Figure 2.4(b) shows the skin detection result of a sample frame in Figure 2.4(a)

(a)



(b)

Figure 2.4. (a) Original image, (b) Skin color detection, white pixels represent skin region, black pixels represent non-skin region.

2.2.2 Hand Motion Analysis

Motion Prediction using Optical Flow

The motion of the hand is analyzed using optical flow [80] and the Lucas-Kanade registration method on the keypoints that represent the hand **luct81**. The optical flow estimation is based on the relationship between the searched motion field at time t and the image at t. The most widely used assumption is that pixel intensity remains constant between consecutive frames. The pixel intensity at the point (x, y) in the image at time I is denoted $I(x, y, t)$, with the assumption that the pixel intensity does not change, we can write

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.5}$$

Where (dx, dy) is the displacement in the next frame after dt time. Using chain rule for differentiation, we get:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \tag{2.6}$$

The image gradients are f_x and f_y :

$$f_x = \frac{\partial I}{\partial x}; f_y = \frac{\partial I}{\partial y} \tag{2.7}$$

If we let:

$$u = \frac{dx}{dt}; v = \frac{dy}{dt} \tag{2.8}$$

In order to solve the optical flow equation 2.6with two unknowns u and v, we use the Lucas-Kanade registration method **luct81**. The Lucas-Kanade method takes the assumption that neighboring pixels have similar motion, it takes a 3 by 3 pixel window around the point and all the 9 points have the same motion. The least square is used to obtain a solution for two unknowns using 9 equations. The final solution is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{\mathrm{i}} f_{x_{\mathrm{i}}}^2 & \sum_{\mathrm{i}} f_{x_{\mathrm{i}}} f_{y_{\mathrm{i}}} \\ \sum_{\mathrm{i}} f_{x_{\mathrm{i}}} f_{y_{\mathrm{i}}} & \sum_{\mathrm{i}} f_{y_{\mathrm{i}}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_{\mathrm{i}} f_{x_{\mathrm{i}}} f_{t_{\mathrm{i}}} \\ -\sum_{\mathrm{i}} f_{y_{\mathrm{i}}} f_{t_{\mathrm{i}}} \end{bmatrix} \tag{2.9}$$

We find the Harris corners [81] inside the hand bounding box and the points on the contour of the hand skin mask and use them as keypoints. The Harris corners are obtained by finding the difference in intensity for a displacement in all directions. A point is considered as a corner point that shifting a window in any direction should yield large change in intensity in that window. The skin mask is obtained using our skin model. By obtaining the optical flow of keypoints using the iterative Lucas-Kanade technique with a pyramid **luct81**, [82], the predicted positions of the keypoints on the next frame can be found. The pyramid structure uses a 3 level pyramid from coarse to fine. Figure 2.5 illustrates motion analysis for keypoints and the trajectories for 10 frames.

Figure 2.5. Zoomed in view of the trajectories of detected keypoints in 10 consecutive frames.

Keypoints Update

The hand bounding box is updated based on the remaining keypoints after discarding outliers using the velocity and skin color tests described below. Where the velocity [61] is defined as the Euclidean distance between the keypoints' location in the previous frame and the predicted position in the current frame.

The average velocity v_{avg} is the arithmetic average of the velocity of all keypoints. The velocity test is:

$$|v - v_{avg}| \leqslant 2v_{std} \tag{2.10}$$

where v is the velocity and v_{std} is the standard deviation of the velocity. Any keypoint that falls outside of the above range are considered an outlier and is discarded. The skin color test checks whether the predicted position of keypoints are on the current skin mask. If not, these keypoints are discarded because we assume the hand is in the skin regions. The new hand bounding box is the smallest rectangle that contains all the remaining keypoints. New keypoints used for motion analysis are the Harris corners and the contour points inside the new bounding box.

2.2.3 Occlusion Handling

Occlusion handling is a challenging problem in object tracking. It is even more difficult in hand tracking, because the hand has a non-rigid shape. In our work, the caregivers are allowed to freely move their hands. Thus, the caregivers often move their two hands together, for example a handclap. The problem then becomes how to keep track of two hands respectively after they separate.

We propose a method to handle occlusion by using the merge and split concept [83]. Figure 2.6 shows the flow chart of this method. We define a hand flag that indicates whether the two hands are together. In the hand initialization step, we manually select two hands and set the hand flag to 1. The hand tracker described above provides the hand position and the centroid of each hand can easily be obtained. When two hands are approaching, the centroids of hands are also getting closer. A merge happens when the centroids of two hands become one point. When it occurs, the hand flag is set to 0. In our experiments the threshold for the merge and split is set to 50 pixels and was empirically determined.

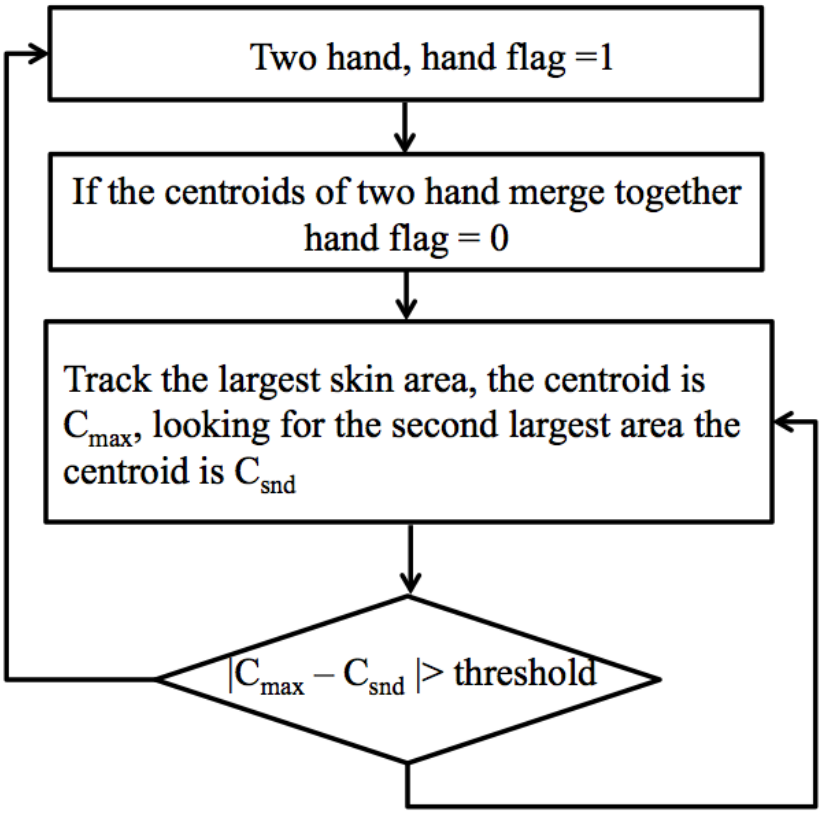
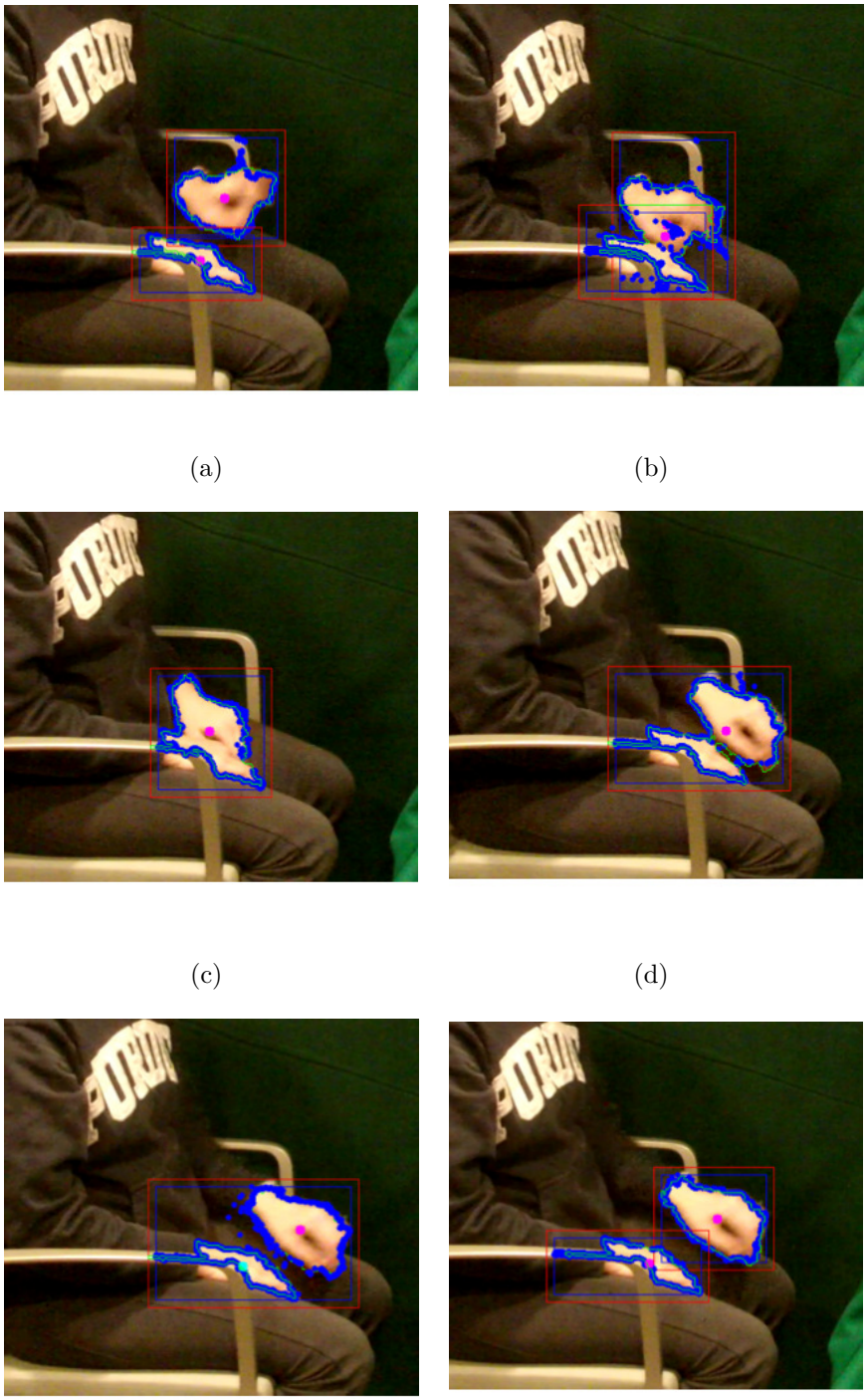


Figure 2.6. Flowchart of the merge and split method.

After the two hands merge together, the independent trackers are tracking the same region. The hand trackers not only track the contour points for largest skin region, but also search for the second largest skin region. Once the Euclidean distance between the centroid of the largest skin region and the centroid of the second largest skin region is greater than the threshold, a split occurs. The one tracker tracks the largest skin region and the other tracks the second largest skin region. The hand flag is then set back to 1.



(a) (b)

(c) (d)

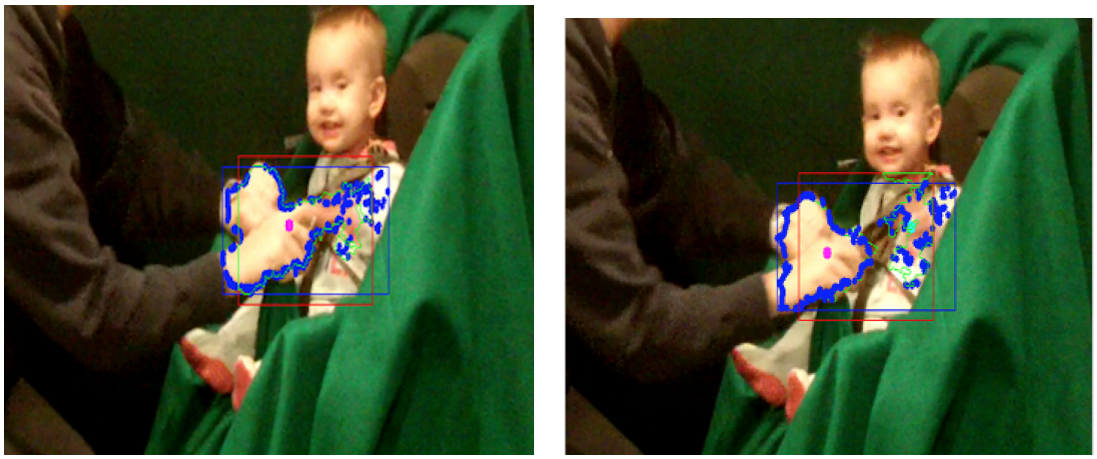
(e) (f)

Figure 2.7. Examples frames of an example of two tracker occulsion occurs and succefully sperater by merge and split.

The two hands merge and split method proposed above works for most of occlusion cases we have observed because hands represent large skin regions in the scene. However, the hand tracker fails in some special cases. For example, during the splitting of two merged hands, one hand may be fully occluded by objects other than the hand. Figure 2.8 and 2.9 illustrates two example failing cases.

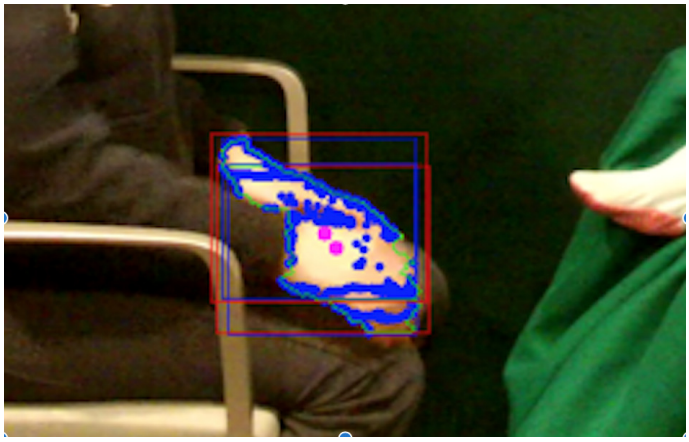
Figure 2.8 happens when the a non-hand skin region is large enough for splitting after a merge. This type of errors would happen in the situations that the caregiver is interacting with the infant. And this tracker is not able to distinguish the whether the skin region belongs to the caregiveri hands or other skin regions like the infant's hands.

Figure 2.9 shows another failing case that one hand is highly occluded and the split is not happening while the two hand are actually separating. Incorrect decision of the splitting time will lead the tracker fails in subsequent frames.

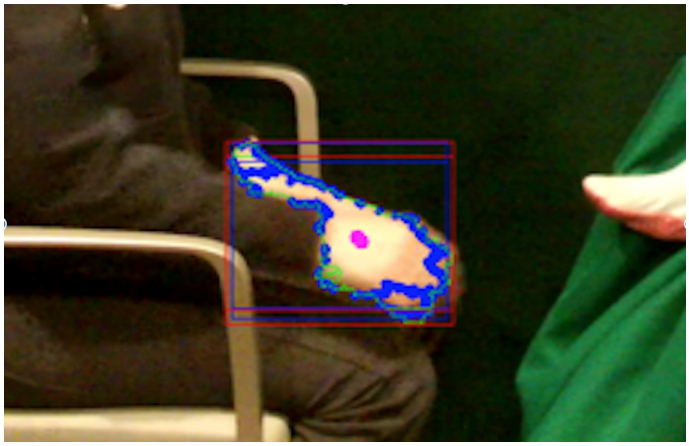


(a) (b)

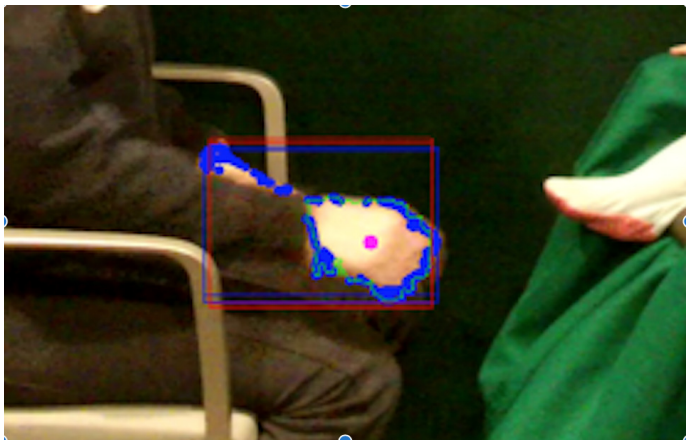
Figure 2.8. Example of failing cases using occlusion handling.



(a)



(b)



(c)

Figure 2.9. Example of failing cases using occlusion handling.

2.3 Long Term Hand Tracking

A considerable amount of literature has been published on adapting hand tracking with most popular tracking frameworks, for example Camshift [61]. These methods works well under the condition that hand trajectories are simple and without much occlusion. The major shortcoming of these methods is that the tracker cannot be re-initialized if it fails. However, due to the high number of degrees of freedom in a hand and the frequent interactions between the caregiver and the infant, the previous describes hand tracker would drift if the tracking error propagated from frame to frame. Thus, it is important to design a hand tracker that can accurately re-initialize the hand positions.

Several attempts have been made to integrate hand detection into tracking schema. A hand detection method which selects hand proposals from hand shape, context and skin color based on the deformable parts model [72] in static images was introduced in [71]. The same method was used in [65] with an upper body detection in tracking to refine the hand positions in videos. Hand models trained using neural networks were proposed in [84], [85] to solve different vision tasks. Automatic human pose estimation also shifted from classical methods like graphical models to deep neural networks. DeepPose [86] is one of the earliest deep-learning based method to estimate human poses which uses a convolutional architecture to directly regress the coordinates of joints. In [87], a structural heat map is predicted to characterize the probabilities of joints at different locations through multiple resolutions.

Our method for long term hand tracking extends the idea of using hand detection and the locations of body joints to refine tracking results. A hand model is trained using the Faster-RCNN [62]. In addition, we use a stacked hourglass network [63] to predict the keypoints of human body. The hand position in the tracker is updated and corrected if need to based on hand detection and body keypoints results.

Initialization

The hand positions and an approximate center of the human body in the first frame are manually initialized to yield an accurate starting point. No additional manual input is needed to track the hand position in the rest of the video. We assume that both hands are

visible in the first frame and we provide a simple interface that allows users to manually draw hand bounding boxes using a mouse with a click and drag action. Each hand bounding box is treated as an independent hand tracker. The center of the body can be determined directly by clicking on the approximate center of the person. It is used as an input to the human pose estimator.

2.3.1 Hand Tracker

The hand tracker uses color and motion features to select and track keypoints in each frame. Due to the lack of feature points in small hand bounding boxes, instead of using SIFT features [88] to detect keypoints, our tracker uses contour points and Harris corners [81] to preserve temporal information. Contour points are generated using a pixel-based skin detection method [77]. Previously, we used merge and split concept [83] to handle occlusion [89]. However, tracker re-initialization was not addressed which became problematic for longer videos if the track cannot be recovered and errors can propagate from frame to frame. In this method, we can recover the tracker in subsequent frames based on the information from the hand detector and wrist position estimator which will be explained in Section 2.3.4.

2.3.2 Hand Detection

Overview of Object Detection Methods

Object recognition is one of the hottest research topics in computer vision. Object recognition contains the problems of detecting, localizing and classifying generic objects in static images. It is a challenge problem because the high variations of the appearance in one object category, for example the view point variation. PASCAL VOC (Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes)[90] ran challenges evaluating performance on object class recognition from 2005-2012. It contains 54,000 objects in 22,000 images and a total of 20 object classes. Since 2010, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [91] has been used as a standard benchmark for evaluating large-scale object recognition performance. It contains 1.2 million training images, 50,000

validation images and 100,000 test images of 1000 object classes. We review some popular object detection methods that are worth mentioning.

Deformable Part Models: Deformable part models(DPM) [72] uses Histogram of Oriented Gradients (HOG) as feature and SVM as classifier. DPM computes HOG feature of the whole image at multiple resolutions. Each object model contains root filter and part filters. The filter score of each sub-window of the feature pyramid is compute, and it is the dot product of filter and feature vector. The final score is sum of filter scores minus deformation costs. DPM reach 49% average precision for person detection on PASCAL VOC 2007. It was the state of art object detection method before the deep neural networks. Figure 2.10 illustrates the flowchart of the deformable part models. Figure 2.11 illustrates an example of the feature pyramid.

A DPM based hand detection method is proposed by [71]. This hand detection using DPM for hand shape and hand context, and use a super-pixel based non-maximum suppression to obtain the final hand hypotheses.

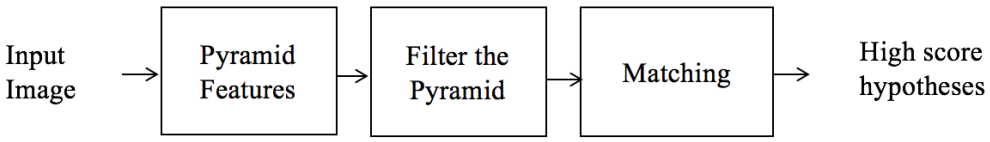


Figure 2.10. Flowchart of the deformable part models.

R-CNN: Regions with CNN features [92] show that a Convolutional Neural Networks can lead higher object detection performance on PASCAL VOC as compared to systems based on HOG-like features. R-CNN use a process called Selective Search to generate bounding boxes of different sizes. A modified version of AlexNet [93] is used as backbone network to pass those bounding boxes into network and SVM is used as object classifier. The final step of R-CNN is to refine the bounding box size using a linear regression model. Figure 2.12 shows the R-CNN detection system.

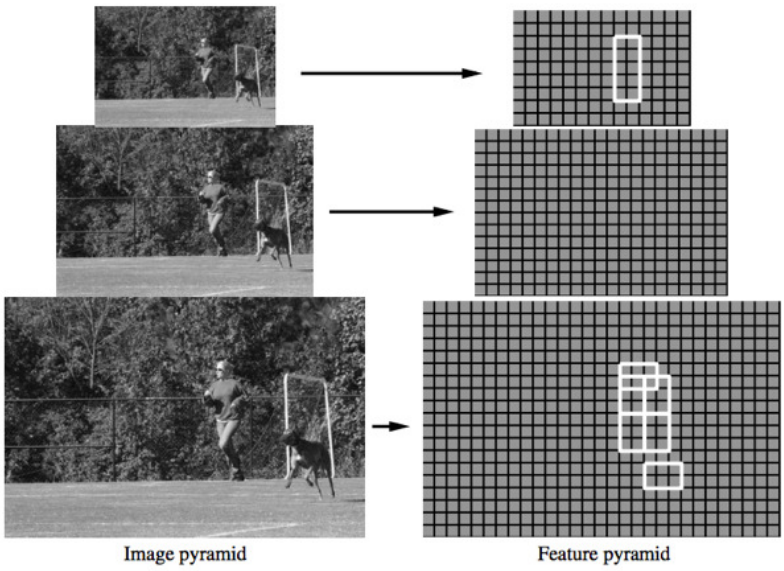


Figure 2.11. A feature pyramid and an instantiation of a person model within that pyramid. The part filters are placed at twice the spatial resolution of the placement of the root. [72]

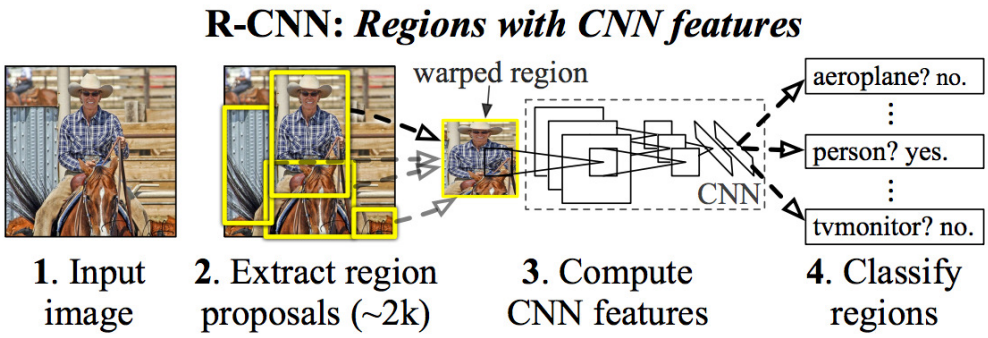


Figure 2.12. R-CNN object detection system overview. [92]

Fast R-CNN: R-CNN works quite slow because it requires a forward pass of the CNN for every proposed bounding box(around 2000 bounding boxes for one image). Additionally, R-CNN need to train the feature generation network, the classifier that predict class and the regression model separately. In Fast R-CNN [94], Region of Interest(RoI) Pooling is introduced which shares the forward pass for an image across its subregions. Faster R-CCN

also used a single network to extract image features, classify objects and regress bounding boxes. Figure 2.13 shows the Fast R-CNN detection system.

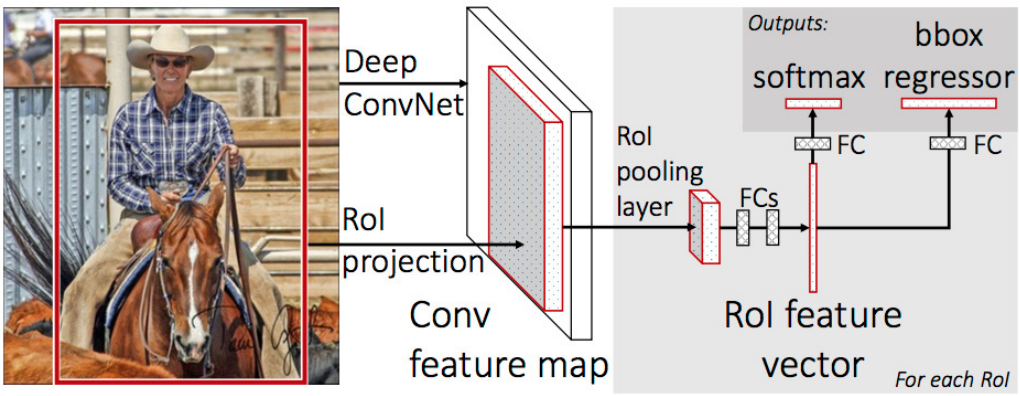


Figure 2.13. Fast R-CNN object detection system overview. [94]

Faster R-CNN: In Faster R-CNN [62], the proposal selection process Selective Search has been replaced. A single CNN is used for region proposals and classification. The bounding boxes are generated by Region Proposal Network which is a fully convolutional network on top of the features of the CNN. The Region Proposal Network passes a sliding window over the CNN feature and output k potential bounding boxes with confidence scores, where k is the number of anchor boxes.

Hand Detection in Our System

We use the Faster-RCNN, which is a powerful object detection network [62], to generate hand hypotheses in each frame. The network consists of a feature extractor, a region proposal network and a softmax classifier. We choose Zeiler and Fergus (ZF) [95] as the feature extractor when train the hand model. Then the region proposal network provides hand proposals and the softmax classifier gives a confidence score ranging from 0 to 1 for each detected hand. A comprehensive hand dataset collected from several different public image datasets [71] is used to train the hand model. This dataset contains 13,050 annotated hand instances, about 4,170 of them are high quality hand instances where the annotated bounding

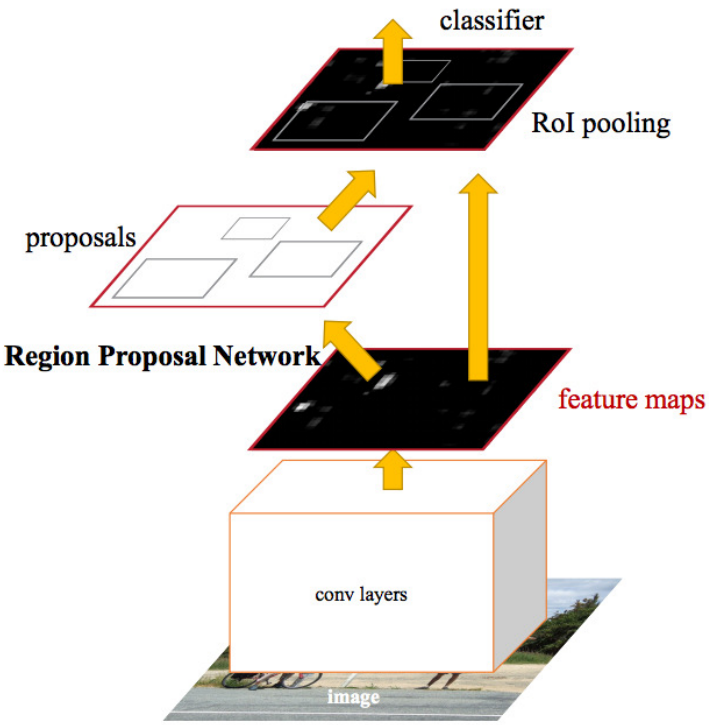


Figure 2.14. Faster R-CNN object detection system overview. [62]

boxes contain more than 1,500 pixels. Figure 2.15 illustrates example hand detection results using Faster-RCNN.

(a) (b)

Figure 2.15. Examples of hand detection results with confidence score greater than 0.9 on the Oxford hand dataset [71]

2.3.3 Human Pose Estimation

Overview of Human Pose Estimation

A key step toward understand people in images and videos is accurate pose estimation [63]. A good pose estimation system need to be robust to occlusion and invariant to changes in appearance due to clothing or lighting condition. Early researches focus on part based models[96]–[98] or pictorial structures [99], [100]. Recent human pose estimations has shifted from classical methods like graphical models to deep neural networks.

DeepPose [86] is one of the earliest deep-learning based method to estimate human poses which uses a convolutional architecture to directly regress the coordinates of joints. [86] also propose a cascade of DNN-based pose predictor which refines the joint predictions by using higher resolution sub-images. In [87], a structural heat map is predicted to characterize the probabilities of joints at different locations through multiple resolutions. They uses a deep ConvNet and a graphical model.

The stacked hour glasses [63] combing features across different resolutions without using any graphical model or any pictorial structures. The hour glass is constructed in a bottom-up, top down fashion. The final architecture contains two hour glasses stacked end-to-end with intermediate loss. The first hourglass predicts an initial set of heatmaps upon which apply a loss. Then, the second hourglass processes these high level features again across all scales to further capture higher order spatial relationships, which is critical to the final performance. Figure 2.16 shows the architecture of stacked hour glasses.

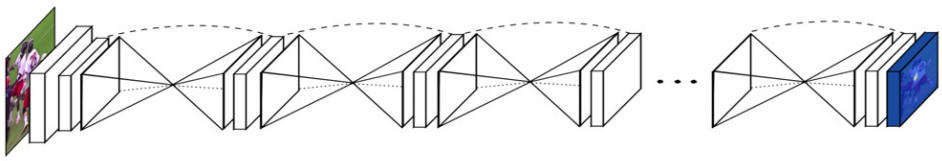


Figure 2.16. Stacked hour glasses architecture [63]. Each box is a residual module

Human Pose Estimation in Our System

Human pose estimation is used to obtain the possible location of joints, in particular locations of wrists. We assume the actual hand positions are near the wrist if the human pose estimator is accurate. The Stacked Hourglass Network is used because it has shown good performance in locating elbow and wrist [63] . The network consists of multiple repeated encoder-decoder architectures that capture information across all scales.

We use a pre-trained model that was trained on the MPII Human Pose dataset [101]. The MPII images contain center and scale annotations, thus we need to determine an approximate center and scale S for the human body as inputs to the human pose estimator. During the initialization step, we manually select the center of the body and two hand bounding boxes B_{i} where $\mathrm{i} = 1, 2$. The width and height for each hand bounding box are denoted as $B_{\mathrm{i}} = \{w_{\mathrm{i}}, h_{\mathrm{i}}\}$. We assume a linear relationship between the scale S and the selected hand area, as stated in Equation 2.11.

$$S = a\sqrt{0.5 \sum_{\mathrm{i}=1,2} (w_{\mathrm{i}} \times h_{\mathrm{i}})} + b \tag{2.11}$$

We tested the human pose estimator with the initialization step on a subset of the MPII test images to obtain proper values for coefficients a and b, which are set to 0.06 and 1.5 accordingly for the image resolution of 1280×720. Scale S need to be adjusted linearly with the image resolution.

2.3.4 Combining Multiple Proposals

The hand tracker provides one hand hypothesis for each tracker based on temporal information, while the hand detector gives proposals that are greater than a certain confidence score. The human pose estimator provides possible locations of wrists.

To detect and re-initialize the current hand track position, we look for instances when both the hand detector and the human pose estimator return a strong hand hypothesis that is not in agreement with the current tracker position. Instead of using a high recall random forest classifier [76] as the hand detector, we set the threshold of the confidence score for the hand detector to 0.9 to maintain a high precision to enlarge the probability that the

re-initialization happens at the true hand position. The hand position is re-initialized when the predicted wrist is inside a valid hand detection proposal. i.e., a confidence score greater than 0.9. If there are multiple valid hand detection proposals, we pick the one with the highest confidence score. Figure **??** shows the re-initialization process which selects the best candidates from several hand proposals with joints information.

2.4 Graph-Based Method For Infant Segmentation

Graph-based image segmentation techniques generally represent the problem in terms of a graph $G = (V, E)$ where each node $v\mathrm{i} \in V$ corresponds to a pixel in the image, and the edges in E connect certain pairs of neighboring pixels. A weight is associated with each edge based on some property of the pixels that it connects, such as their image intensities. Depending on the method, there may or may not be an edge connecting each pair of vertices. A graph-partitioning method attempts to organize nodes into groups such that the intra-group similarity is high and the inter-group similarity is low. A *Cut* which partitions the graph or subgraph into two disjoint sets A and $B = V - A$ is sometimes defined as a total weight of the removed edges:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

A key challenge is to find the minimum cut.

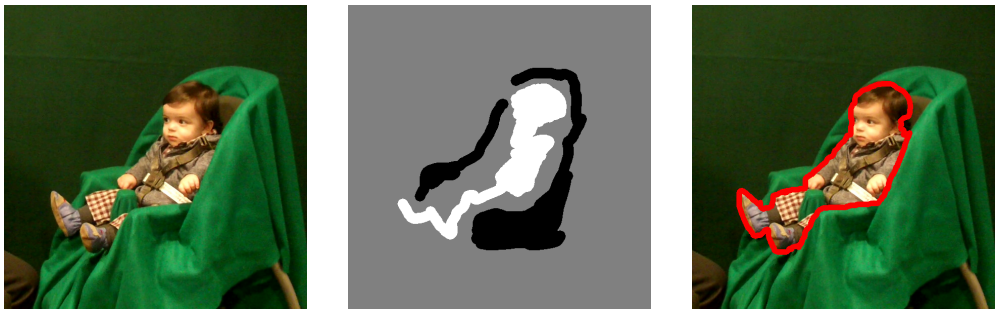
2.4.1 Pixel-wise Grab-Cut Infant Segmentation

We use Grab-Cut [27] to detect the contour of the infant in every frame of the video sequence. Grab-cut segmentation is an iterative method based on Graph-Cut[102], which is described by the Gibbs energy:

$$E(x) = \sum_{\mathrm{i} \in I} D(x_{\mathrm{i}}) + \lambda \sum_{\mathrm{i} \in I, \mathrm{j} \in N_{\mathrm{i}}} V(x_{\mathrm{i}}, x_{\mathrm{j}}) \tag{2.12}$$

where i is a pixel that belongs to image I, N_{i} is the neighboring pixels of i, x_{i} takes on the value of 0 for sure background, 1 for sure foreground, 2 for probably background, and 3

for probably foreground. $D(x_i)$ is the data term, and $V(x_i, x_j)$ is the smoothing term. The data term $D(x_i)$ is modeled by a Gaussian Mixture Model (GMM), where we estimate the probability distribution of the background and the foreground. A mask is generated by the user that marks the foreground as RGB color white $(255, 255, 255)$, background as RGB color black $(0, 0, 0)$, and the unknown region as a RGB color different than black and white. Based on this initial graph, the Grab-Cut method finds a minimum cost to the energy function. A zoomed in view of the original image and its mask image is shown in Figure 2.17(a) and 2.17(b). The infant segmentation can be seen in Figure 2.17(c).



(a) (b) (c)

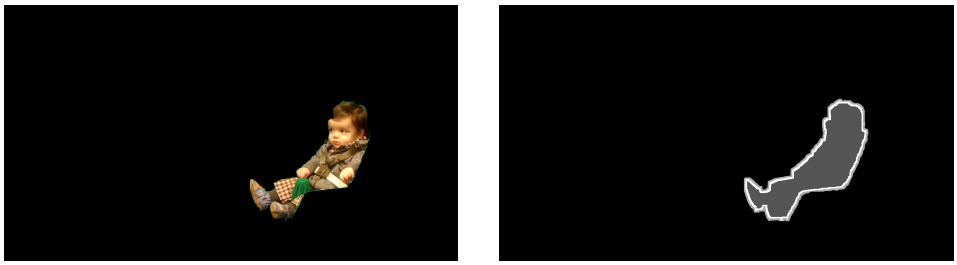
Figure 2.17. (a) Original image (b) Infant mask (c) Infant Contour.

Background and Foreground Updates

As the initial mask is obtained from the first frame of the video, the mask need to be update for later frames. Morphological operations, erosion and dilation, are conducted on the result of previous frame to generate the mask denoting the sure foreground region, sure background region, probable foreground region, and probable background region of the current frame. The kernel for the erosion and dilation are decided based on the addition of the average velocity from the motion analysis to a fixed given base kernel size.

The result of the previous frame is first converted into a binary image, and processed by a morphological closing with kernel size 3×3 to fill the interstitials and smooth the borders. Then erosion and dilation operations for the current frame are conducted to generate the probable foreground and probable background regions. The intersection of the sure background from the current mask and the the region after dilation is set to probable foreground.

And the sure foreground is refined by the intersection of the current sure foreground and the region after erosion operation. Figure 2.18(b) shows the erosion and dilation operations on the infant segmentation using the GrabCut result from the previous frame shown in Figure 2.18(a).



(a) (b)

Figure 2.18. Morphological Operations on Infant Segmentation

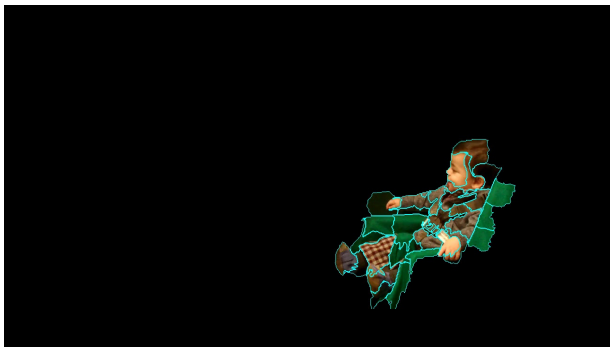
2.4.2 Superpixel Grab-Cut Based Infant Segmentation

Instead of doing a pixel-wise Grab-Cut in previous method, we are using a simple linear iterative clustering (SLIC)[103] as the initial segmentation tool to obtain the superpixels for the frame. The SLIC method computes a local clustering of pixels in 5D space consisting of L, a, b values from CIELAB color space and x, y pixel coordinates. The sure foreground in the mask is obtained using joint locations of the infant from the human pose estimation in section 3.3.3. We connect the joints to obtain a skeleton mask as the sure foreground. The sure background mask is manually initialized as described in previous method. The intersection of the sure background and the sure foreground will be set to probable foreground. Figure 2.19 shows the infant segmentation result using this method.

The super-pixel Grab-Cut method with updated mask using joint information is more robust to the movement of the infant. Because the foreground mask is update in each frame using the pose estimation results for the infant.

(a)

(b)



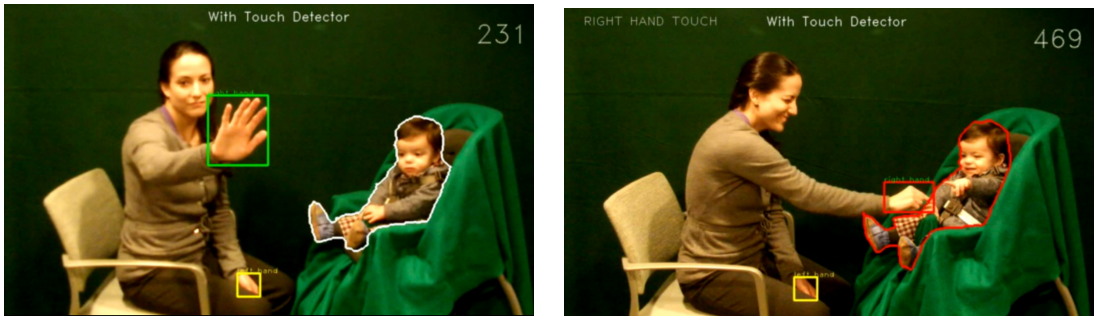
(c)

Figure 2.19. (a) Original image with skeleton (b) Infant mask (c) Infant Contour.

2.5 Proposed Touch Detector Method

After obtaining the caregiver's hands position and infant's contour from the hand tracker and baby detector in each frame, a touch event can be defined as the time period during which contours merge. We examine each hand of the caregiver with the infant's contour separately. Once a hand touches the infant, the contour of the hand and the contour of the baby will merge into one contour. When this situation occurs, we can declare that a touch

event has occurred. The same method is used to detect whether a touch event has occurred for the other hand. Either hand touching the baby will result in a touch event detection and labeling that frame as a touch event. Figure 2.20(a) shows a frame without a touch event, and Figure 2.20(b) shows an example of a potential touch event by detecting the merging contours of the hands and the infant.



(a) (b)

Figure 2.20. (a) An example of a frame without a touch event, (b) An example where a potential touch event has occurred by detecting merging contours.

2.6 Improved Touch Detection Method

The proposed improved touch detection method performs two tasks: (1) identify a frame is touch or non-touch by checking whether the hand segments of the caregiver overlap with the infant segment, (2) classify a detected touch frame into six different touch type classes based on the spatial relationship between keypoints on caregivers' hand and infant body parts. The overview block diagram is shown in Fig. 2.21. Hand segment updates are introduced in Section 2.6.1, infant segmentation is described in Section 2.6.2, and touch detection decision is described in Section 2.6.3 .

2.6.1 Hand Location

The pose estimator [104] we used is trained on Microsoft COCO [105] dataset and a foot dataset [104] with a total of 25 joints. After obtaining wrist and elbow position from the pose estimator, the hand joints detector [106] is applied by assuming hand is located at an

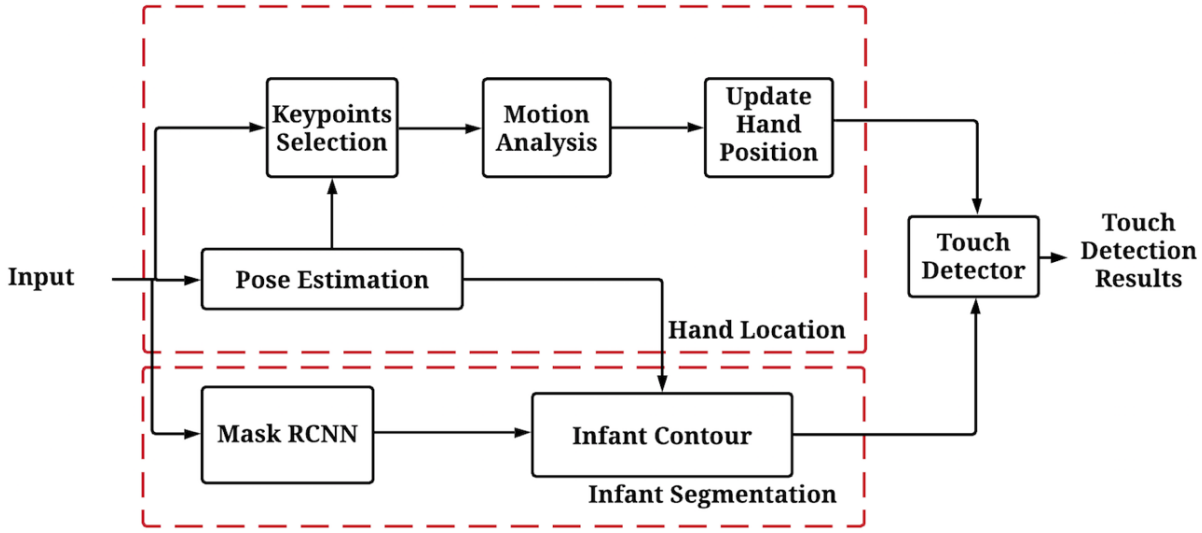


Figure 2.21. Improved touch detection block diagram

extend region of forearm in the same direction, we denote as $I_{crop} \in \mathbb{R}^{w \times h \times 3}$. The hand joints detector $h(\cdot)$ maps cropped hand region I_{crop} to N joints locations $\mathbf{x_n}$ associated with a score c_n, where N is 21 in this model. An example of human pose estimation and hand joints estimation are shown in Figure 2.22, where joints information for infant will be used in Section 2.6.2 and Section 2.6.3. We say a hand joint is detected if Equation 2.13 is equal to one, where $1(\cdot)$ is an indicator function, α and β are empirically set to be 0.5 and 10, respectively.

$$Confidence = 1((\sum_{n \in [1...N]} 1(c_n > \alpha)) > \beta) \tag{2.13}$$

We extend the tightest bounding box enclosing all hand joints ten pixels in horizontal and vertical direction, and use the extended rectangular as the hand bounding box. If the detected hand is not confident or the pose estimation does not provide wrist or elbow position, then a feature based tracking method is enabled to continue updating hand position. The hand tracker uses color and motion features as keypoints to track in each frame. Due to the lack of feature points in small hand bounding boxes, instead of using SIFT features [88] to detect keypoints, our tracker uses contour points and Harris corners [81] to preserve temporal information. Contour points are generated using a pixel-based skin detection method [77]. Similarly, the final hand bounding box is the extended rectangular enclosing all keypoints.

The hand tracker is disabled when a confident hand joints prediction is available. The final hand segment is obtained by applying skin detection inside the hand bounding box.

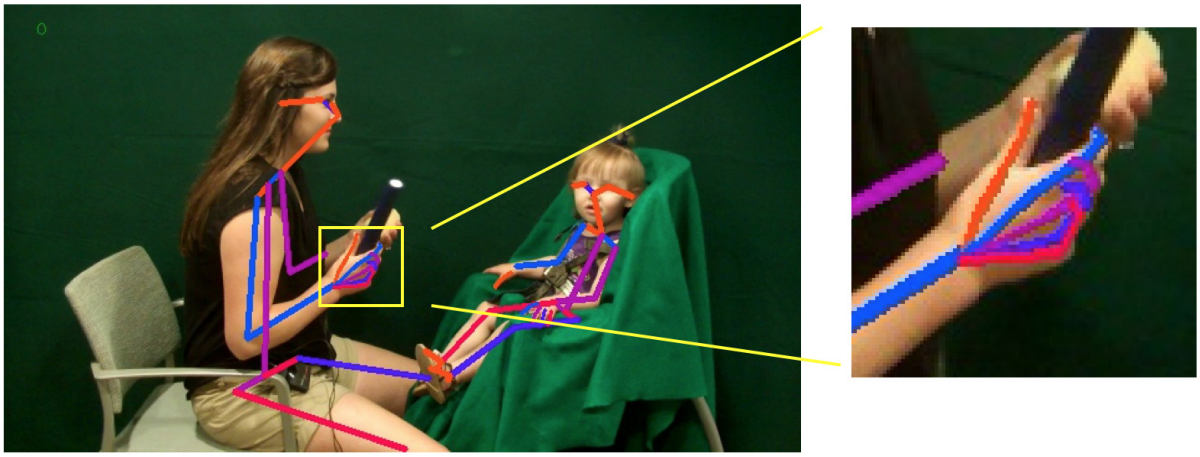
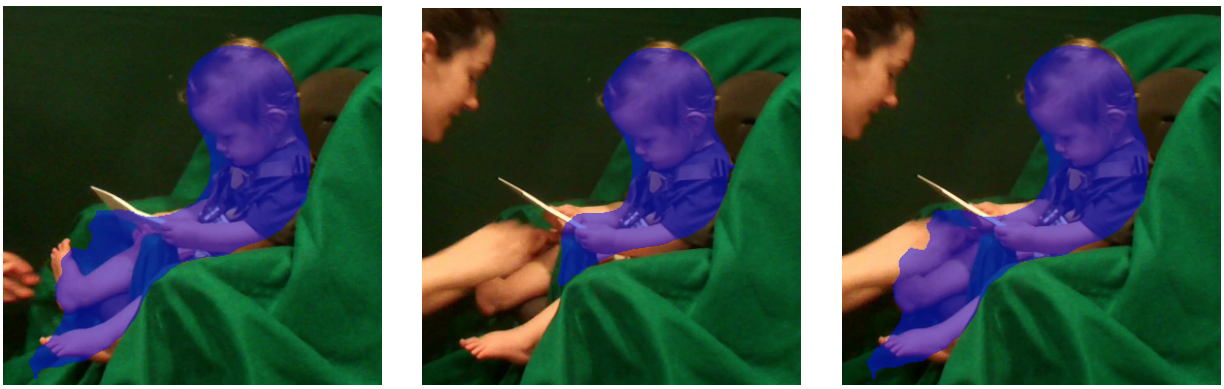


Figure 2.22. Human pose estimation and hand joints estimation

2.6.2 Infant Segmentation

Mask-RCNN [107] is trained on Microsoft COCO [105] and is used to generate infant segmentation. Infant body parts may be occluded by caregiver or other objects during their interaction, and Mask-RCNN tends to exclude the occluded region, an example is shown in Figure 2.23(b). However, excluding occluded region is not consistence with the way we detect touch event. Thus, we proposed a temporal refinement to recovery the occluded region. We use the confidence score of infant joints to assess the confidence of an infant segmentation by assuming when parts are missing in the infant segmentation, the confident score of occluded joints is also low. Equation 2.13 is used to determine whether a infant segmentation is confident, where α and β are empirically set to be 0.3 and 20 respectively, and c_n is the confidence score for a infant joint here. An invalid infant segmentation at t_1 is recovered by using a confident segmentation from previous frame at t_0 as shown in Figure 2.23(c).



(a) (b) (c)

Figure 2.23. Infant segmentation temporal refinement, (a) valid infant segmentation at time t_0 (b) occluded segmentation at time t_1 (c) infant segmentation recovered by temporal refinement at time t_1.

2.6.3 Touch Detection

The proposed method makes decision to label a frame as "touch" or "non-touch" first, and then assigns a touch type label L_{i} to detected "touch" frame based which part of infant body has been touched, where $L_{\mathrm{i}} \in \{\text{“}head\text{”}, \text{“}hand\text{”}, \text{“}torso\text{”}, \text{“}arm\text{”}, \text{“}leg\text{”}, \text{“}foot\text{”}\}$ and i is the index. Whether touch occurs in a given frame is determined by checking if caregiver's hand segments, obtained from Section 2.6.1, overlap with the infant segmentation from Section 2.6.2.

To classify touch type, we analyze the spatial relationship between caregiver's hands and infant body parts. We define six infant body parts corresponding to six touch type labels, and each part contains a set of limbs, where limbs are pairs of adjacent joint points belongs to that part. For example, left elbow and left wrist are a pair of adjacent joint points, they form the limb left forearm and belongs to the part "arm". We use a straight line connecting from one joint to another to fit the body limbs, and using a set of points to represent the fitted line, they are linearly spaced in 0.1 pixel in horizontal direction. Then for a given frame I, there are sets S_{i} for $\mathrm{i} \in [1 \ldots 6]$ contains fitted points for each part respectively to

represent infant body parts. We evaluate the Euclidean distance of joint points of caregiver's hands to infant body parts. For each hand point $\mathbf{x_n}$, we get a label z_n using Equation 2.14.

$$z_n =_{\mathrm{i}} \|\mathbf{x_n} - \mathbf{x_p}\|_2^2, \forall \mathbf{x_p} \in S_{\mathrm{i}} \tag{2.14}$$

The final touch type is determined as the majority vote of z_n, where $n \in [1 \ldots N]$, and N is the total number of caregiver's hand points.

2.7 Experiments

2.7.1 Hand Tracking Experiments

Dataset and Evaluation Metric

The proposed hand tracker with automated re-initialization is tested on a public dataset and on our own dataset. The public dataset [108] contains three sequences, one sequence is 2000 frames and the other two sequences are around 400 frames. Furthermore, only a single person appears in each video and his two hands are visible in frontal view for most of the frames.

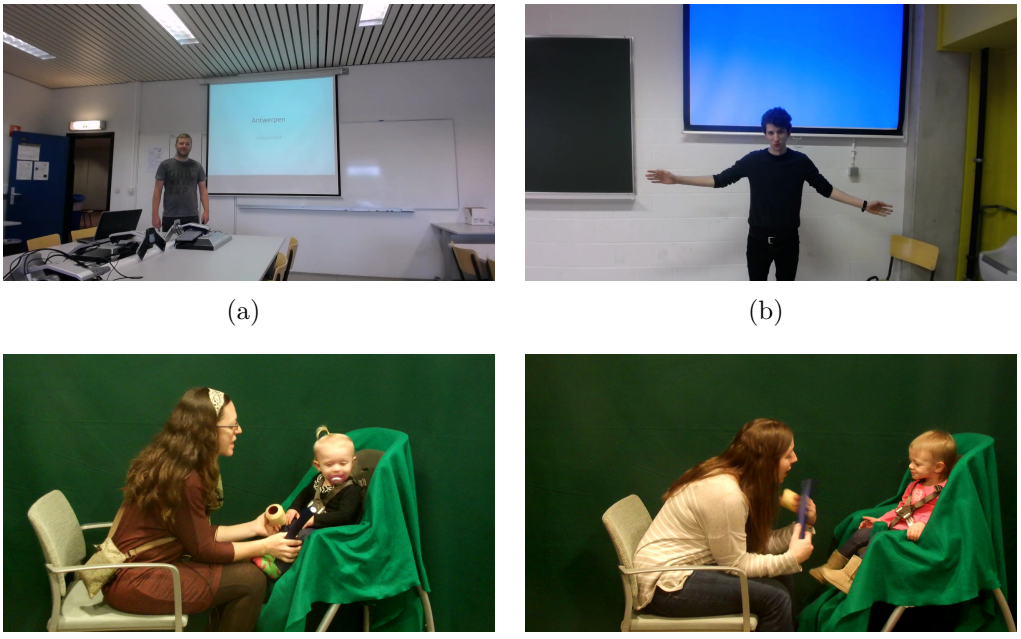
Our dataset contains five long video sequences (2500 frames each) with a more challenging setting where the caregiver is mostly in the profile view. An infant is also present in the video, adding more complexity to the dataset for tracking hands. In addition, the caregiver is free to interact with the infant, hand occlusion and hand vanishing are common in these videos. These videos were acquired from different pairs of caregivers and infants at different times and dates under the same recording settings. The recorded videos have a resolution of 1280×720 at 30 fps. In each image, two hands of the caregiver are annotated. If the hand can be perceived clearly by human, the annotations consist of a axis-aligned bounding rectangle for each hand. If the hand is not observed, we annotate the hand is not visible on that frame. Figure 2.29 shows sample frames from both dataset.

The intersection over union score is used for evaluation, a score lower than 0.5 is considered as a tracker error.

$$IoUscore = \frac{Area(box_{tracker} \cap box_{groundtruth})}{Area(box_{tracker} \cup box_{groundtruth})} \tag{2.15}$$

We also compute the F-measure for comparing tracking accuracy among different methods.

$$F\text{-}measure = \frac{2TP}{2TP + FP + FN} \tag{2.16}$$



(a) (b)

(c) (d)

Figure 2.24. Examples frames from two datasets, (a) and (b) are from [108], (c) and (d) are from our dataset.

Experimental Comparison Results

We compared the proposed hand tracking method to an image based method [71] and a semiautomatic method [109] which allows a user to manually correct the hand position when tracker fails throughout the video. Our results, shown in Table 2.2, improved significantly compared to the hand detection method. Our method also performed comparably to the semiautomatic method with manual hand re-initialization for the two shorter videos, i.e., dataset1 and dataset2 and showed improved performance for dataset3 which contains a longer video.

We also tested the proposed hand tracking method on a more challenging dataset of our own which contains 5 videos of more than 2000 frames each. In this dataset, the hands of the caregiver is often occluded when interacting with the infant. Another challenge of this dataset is that only the side view of the caregiver is captured where there are limited views of the hands and one hand is mostly occluded by the other hand posing significant difficulty

Table 2.2. Tracking Results from dataset [108]

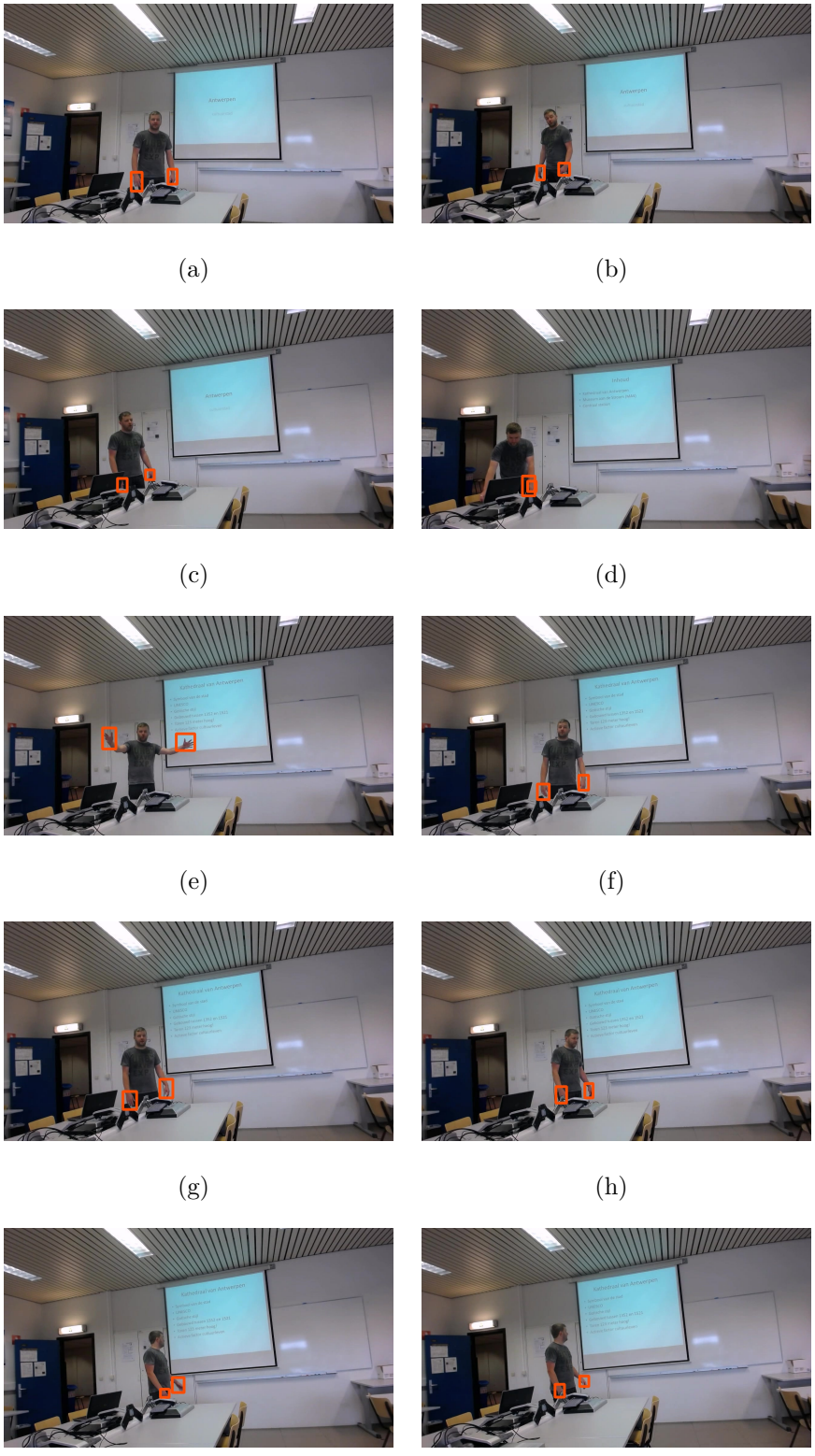
Name	Length	Mittal etc. [71]	Beugher etc. [109]	Ours
Dataset1	403	85.0%	95.76% (manual)	94.95%
Dataset2	492	46.5%	92.75% (manual)	89.41%
Dataset3	2000	-	88.31% (manual)	90.54%

is tracking that hand. The performance on our dataset is shown in Table 2.5. Results shown in column 3 of Table 2.5 was obtained by using the hand tracking method described in [89].

Table 2.3. Tracking Results from our dataset

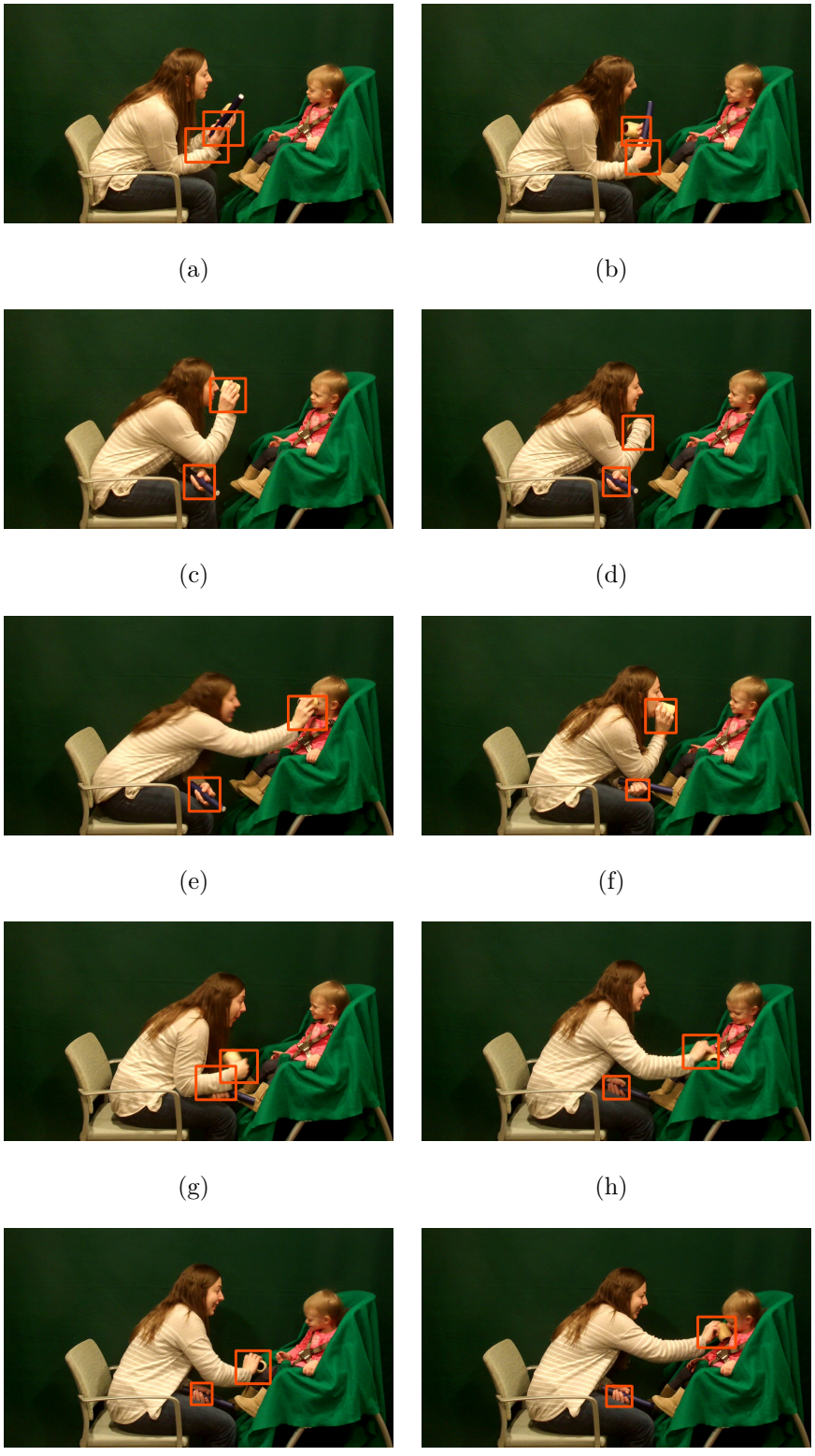
Name	Length	Chen [89]	Ours
Sequence1	2500	20.41 %	75.77%
Sequence2	2500	29.36 %	80.70%
Sequence3	2500	29.45 %	76.24%
Sequence4	2500	26.37 %	72.41%
Sequence5	2500	4.61 %	82.96%
Avg	-	22.04 %	77.62 %

Our method significantly outperformed the method described in [89], where tracking errors are more prominent in longer videos without re-initialization . Sample tracking results for both datasets are shown in Figure 2.25 and Figure 2.26.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 2.25. Sample tracking results for the public dataset [109].



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 2.26. Sample tracking results for our dataset.

2.7.2 Touch Detection Experiments

Proposed Touch Detection System

To test the performance of our method, we recorded the interactions between a caregiver and an infant in a lab setting. In these experiments, the caregivers were asked to interact with the infant as they would normally do during playtime. The infant was secured in a high chair and the caregiver sat on a chair facing the infant. The lab where the experiments were conducted had a green wall as background and the high chair was also covered by a green blanket. A RGB camera and a clip-on wireless microphone were used to record video and audio data. The video sequences were acquired from different pairs of caregivers and infants at different times and dates while under the same recording settings. The videos were recorded at a resolution of 1280×720 and with a frame rate of 30 fps.

The videos were processed using our automatic touch detector. In our experiments the threshold C in the skin detector was set to 0.0001 and the threshold for the merge and split for the hand occlusion is set to 50 pixels. We compared and illustrated the results with the ground-truth data annotated by a trained analyst for an example video sequence. Figure 2.27 shows a sample result of the performance comparison. The green bars and yellow bars indicate potential touch events detected using the hand tracking with occlusion handling and the long-term hand tracking respectively. The blue bars are touch events noted by a trained analyst. The automatic touch detector successfully captured all touch events, but included one false alarm. This was mainly due to the lack of precise hand contour detection for some frames in the video and difficulty in dealing with occlusions due to the camera view. The current automatic touch detector cannot differentiate different types of touch, resulting in one touch event detected instead of three consecutive touches (resting, grabbing, moving) as indicated by the trained analyst between 11.2 - 14.2 seconds.

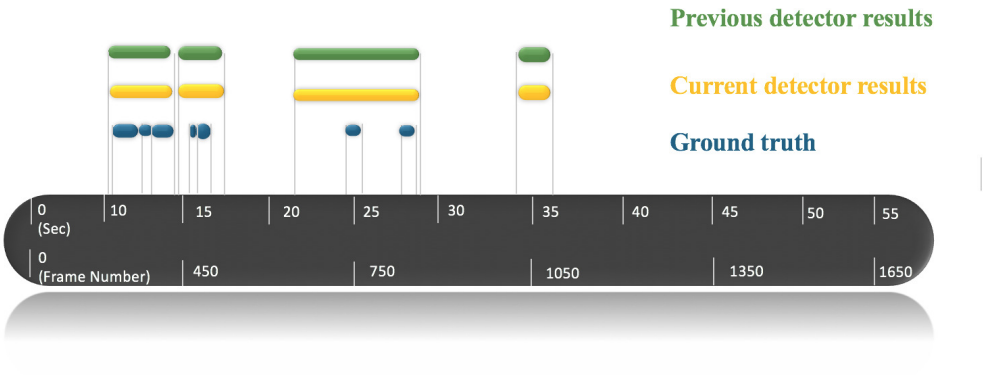


Figure 2.27. The comparison of touch event detection with a trained analyst using two hand tracking methods

Improved Touch Detection System

Dataset

We evaluate the performance of our method on a testing set which records the interactions between a caregiver and an infant in a lab setting. Our testing dataset contains five 2500 frames video sequences and two long video sequences (more than 9000 frames each). The video sequences were acquired from different pairs of caregivers and infants at different times and dates while under the same recording settings. In these experiments, the caregivers were asked to interact with the infant as they would normally do during playtime. The infant was secured in a high chair and the caregiver sat on a chair facing the infant. The lab where the experiments were conducted had a green wall as background and the high chair was also covered by a green blanket. A RGB camera and a clip-on wireless microphone were used to record video and audio data. The videos were recorded at a resolution of 1280×720 at 30 fps.

The precise labels for each video in the testing set are annotated by trained analyst, and this information is used as groundtruth for our evaluation. The testing set contains $25,043$ out of $31,374$ non-touch frames. The type of touch occured are listed in Table 2.4.

Table 2.4. Touch types occurrences

Labels	Head	Hand	Torso	Arm	Leg	Foot
frames	403	472	367	168	3346	1575

Evaluation Metrics

Precision and recall are used to assess the performance of the automated touch detection method. Our method aims to narrow down the potential touch time intervals and provide less frames for trained analyst to annotate compared to original video sequences. Thus, recall of this method is more important than precision. Another metric potential saving amount in number of frames is used to show how much work may be reduced for trained analyst when annotating only the potential touch frames after using automated touch detection compared to annotating the entire sequence. In other words, trained analyst could skip

annotating predicted non-touch frames ($FN + TN$), because they were less likely to contain touch frames.

$$PotentialSavingAmount = \frac{FN + TN}{TP + FP + FN + TN} \tag{2.17}$$

For touch type detection, we use a confusion matrix to evaluate the quality of our detection results.

Experiment Results

The touch/non-touch detection results are showed in Table 2.5. The proposed automatic touch detector successfully captured 99.19% of touch frames with a precision score of 48.13%. Potential saving amount in table 2.5 shows the trained analyst could skip 58.41% number of frames, which is a great reduction compared to annotating every frame. The proposed method outperforms previous work [89] by having less missed touches, higher precision in predicted touches and less frames needed for trained analyst to annotate.

The confusion matrix in Figure 2.28 illustrates the performance of touch type detection. We observed that "Head" class and "Foot" class have higher scores compare to other classes. Considering those two classes are located in the top and bottom part of an infant segmentation respectively, they are less likely to be confused with other body parts. Taking "Torso" class for an example, 77% of "Torso" class are predicted correctly with 12% are classified as neighbor class "Arm" and 9% are labeled as "Leg". Because infant body part torso are spatially close to legs and arms.

Table 2.5. Touch interaction detection results from our dataset

	Total Frames	Method	Recall	Precision	Potential Saving Amount
Testing sequences	31,374	Proposed[89]	72.03%	24.66%	41.07%
		Improved	99.19%	48.13%	58.41%

With 48.13% of precision for touch detection results, the proposed method still detected more false alarm than true touches. This was mainly due to the lack of precise hand contour detection for some frames in the video and difficulty in dealing with occlusions due to the camera viewing angle. In addition, without the third dimension information, it is difficult to

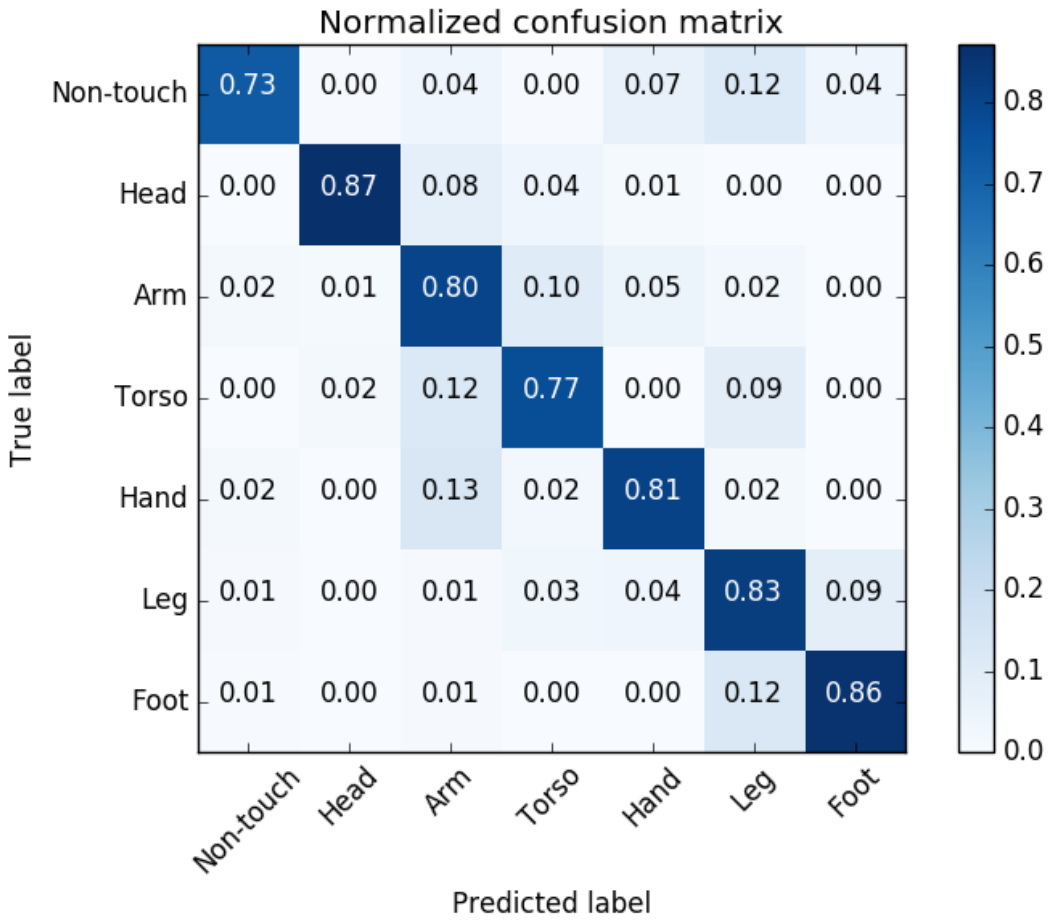


Figure 2.28. Confusion matrix of touch type labels

distinguish from a true touch to fake touch illustrate in Figure 2.29(a) and Figure 2.29(b). Furthermore, we feel these challenging potential touch frames require a second look from trained analyst. From our results, the potential saving amount is larger for videos sequences where the caregiver is well separated from the infant when they are not interacting (Figure 2.29(c)) than those caregiver and infant pairs that are in close-proximity (Figure 2.29(d)) for entire sequences.

Conclusion

We proposed an automatic touch event detection system that detects and tracks the caregiver's hands, detects the location of the infant and then defines a "touch" to occur whenever the caregiver's hand contours overlap with the infants contour. The touch type label is assigned to predicted touch frames based on the spatial relationship between caregiver's hands and infant body parts. The proposed method avoids using expensive precise touch annotations for training. Instead it takes advantage of CNN models that are trained on large public datasets to produce intermediate results needed to identify touches. The proposed method allows trained analyst skip annotating 58.41% of frames and still be able to capture more than 99% true touch frames.

(a)

(b)

(c)

(d)

Figure 2.29. Examples frames from testing set, (a) true touch (b) false touch (c) well-separated (d) close-proximity

3. TEXTURE ANALYSIS IN VIDEO COMPRESSION

3.1 Overview

Modern video codecs such as HEVC [37] and AV1 [50] use hybrid coding techniques consisting of motion compensation and 2D transform to remove spatial and temporal redundancy. However, efficient exploitation of statistical dependencies measured by a mean squared error (MSE) does not always produce the best psychovisual result. Some regions in the frame, e.g., texture, are "perceptually insignificant" where an observer does not notice any difference without observing the original video sequence, but are costly to encode. Texture based approaches have been shown to improve the coding efficiency for "perceptually insignificant" regions [51]–[54].

The previous attempt [51] yielded encouraging bit rate savings without decreasing visual quality. This was accomplished by perceptually differentiating pInSIG textures and other areas to be encoded in a hybrid coding framework. However, the corresponding texture masks were derived using traditional methods, at the coding block level. Similar ideas were explored in work [55], where they do not encode the texture regions, but instead these regions are reconstructed at the decoder based on a motion model. A block diagram that illustrates the video coding using texture analysis and reconstruction is shown in Fig. 3.1. A

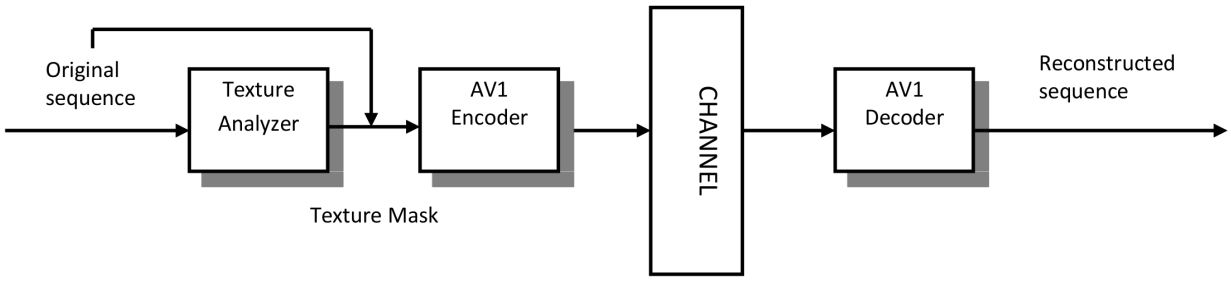


Figure 3.1. Overview of texture-based video coding

Convolutioanl Neural Networks (CNN) based texture analyzer was developed to identify the texture regions in a frame and generates block-based texture masks. The displacement of the entire texture region is modeled by a set of motion parameters. At the decoder, instead

of performing motion compensation prediction to reconstruct blocks in the texture region, the texture blocks are warped from the reference frames towards the current frame using the motion parameters.

While the proposed block-based approach in [55] can achieve a data rate saving of 1% to 13% compared to the baseline when implemented using AV1 with satisfactory visual quality, the block-based texture masks cannot always accurately represent the texture regions. The block-based texture masks can be seamlessly integrated into AV1 since the common coding units are blocks. However, it can sometimes cause noticeable visual artifacts when an identified texture block consist of small structural region. In addition, the smallest texture block size in [55] was 32×32 in order to avoid detecting small moving objects, but at the same time limits the size of identified texture regions and reduces potential data rate savings.

In this thesis, a switchable texture-based video pre-processing that leverages DNN-based semantic understanding for subsequent coding improvement is present. In short, we exploit DNNs to accurately segment "perceptually InSIGnifcant" (pInSIG) texture areas to produce a corresponding pInSIG mask in pixel level. In many instances, this mask drives the encoder to perform separately for pInSIG textures that are typically inferred without additional residuals, and "perceptually SIGnificant" (pSIG) areas elsewhere using traditional hybrid coding method. This approach is implemented on top of the AV1 codec [50], [110], [111] by enabling the GoP-level switchable mechanism, resulting noticeable bit rate savings for both standard test sequences and additional challenging sequences from YouTube UGC dataset [112], under similar perceptual quality. The method we propose is a pioneering work that integrates learning-based texture analysis and reconstruction approaches with modern video codec to enhance video compression performance.

On the other hand, building upon advancements created by DNNs and large-scale labeled datasets (*e.g.*, ImageNet [113], COCO [105], and ADE20K [114]), learning-based semantic scene segmentation algorithms [114]–[116] have been tremendously improved to generate accurate pixel-level texture masks. In this section, a switchable texture-based video pre-processing that leverages DNN-based semantic understanding for subsequent coding improvement is introduced.

3.2 Related Work

Pre-processing techniques are generally applied prior to the video coding block, with the objective of guiding the video encoder to remove psychovisual redundancy and to maintain or improve visual quality, while simultaneously lowering bit rate consumption. One category of pre-processing techniques is the execution of pre-filtering operations. Recently, a number of deep learning-based pre-filtering approaches have been adopted for targeted coding optimization. These include denoising [117], [118], motion deblurring [119], [120], contrast enhancement [121], edge detection [122], [123], *etc.* Another important topic area is closely related to the analysis of video content semantics, *e.g.*, object instance, saliency attention, texture distribution, *etc.*, and its application to intelligent video coding. For the sake of simplicity, we refer to this group of techniques as "pre-processing" for the remainder of this paper. In our discussion below, we also limit our focus to saliency-based and analysis/synthesis-based approaches.

3.2.1 Saliency-Based Video Pre-processing

Saliency Prediction

Saliency is the quality of being particularly noticeable or important. Thus, the *salient area* refers to regions of an image that predominantly attracts the attention of subjects. This concept corresponds closely to the highly discriminative and selective behaviour displayed in visual neuronal processing [124], [125]. Content feature extraction, activation, suppression, and aggregation also occur in the visual pathway [126].

Earlier attempts to predict saliency typically utilized handcrafted image features, such as color, intensity, and orientation contrast [127], motion contrast [128], camera motion [129], *etc.* Later on, DNN-based semantic-level features were extensively investigated for both image content [130]–[136] and video sequences [137]–[143]. Among these features, image saliency prediction only exploits spatial information, while video saliency prediction often relies on spatial and temporal attributes jointly. One typical example of video saliency is a moving object that incurs spatio-temporal dynamics over time, and is therefore more

likely to attract users' attention. For example, Bazzani *et al.* [137] modeled the spatial relations in videos using 3D convolutional features and the temporal consistency with a convolutional long short-term memory (LSTM) network. Bak *et al.* [138] applied a two-stream network that exploited different fusion mechanisms to effectively integrate spatial and temporal information. Sun *et al.* [139] proposed a step-gained FCN to combine the time-domain memory information and space-domain motion components. Jiang *et al.* [140] developed an object-to-motion CNN that was applied together with a LSTM network. All of these efforts to efficiently predict video saliency leveraged spatio-temporal attributes. More details regarding the spatio-temporal saliency models for video content can be found in [144].

Salient Object

One special example of image saliency involved the *object instance* in a visual scene, specifically, the moving object in videos. A simple yet effective solution to the problem of predicting image saliency in this case involved segmenting foreground objects and background components. The segmentation of foreground objects and background components has mainly relied on foreground extraction or background subtraction. For example, motion information has been frequently used to mask out foreground objects [145]–[149].

Recently, both CNN and foreground attentive neural network (FANN) models have been developed to perform foreground segmentation [150], [151]. In addition to conventional Gaussian mixture model-based background subtraction, recent explorations have also shown that CNN models could be effectively used for the same purpose [152], [153]. To address these separated foreground objects and background attributes, Zhang *et al.* [154] introduced a new background mode to more compactly represent background information with better R-D efficiency. To the best of our knowledge, such foreground object/background segmentation has been mostly applied in video surveillance applications, where the visual scene lends itself to easier separation.

Video Compression with UEQ Scales

Saliency or object, which refers to more visually attentive areas, is straightforward to apply UEQ setting in a video encoder, where light compression is used to encode the saliency area, while heavy compression is used elsewhere. Use of this technique often results in a lower level of total bit rate consumption without compromising QoE.

For example, Hadi *et al.* [155] extended the well-known Itti-Koch-Niebur (IKN) model to estimate saliency in the DCT domain, also considering camera motion. In addition, saliency-driven distortion was also introduced to accurately capture the salient characteristics, in order to improve R-D optimization in H.265/HEVC. Li *et al.* [156] suggested using graph-based visual saliency to adapt the quantizations in H.265/HEVC, to reduce total bits consumption. Similarly, Ku *et al.* [157] applied saliency-weighted Coding Tree Unit (CTU)-level bit allocation, where the CTU-aligned saliency weights were determined via low-level feature fusion.

The aforementioned methodologies rely on traditional handcrafted saliency prediction algorithms. As DNN-based saliency algorithms have demonstrated superior performance, we can safely assume that their application to video coding will lead to better compression efficiency. For example, Zhu *et al.* [158] adopted a spatio-temporal saliency model to accurately control the QP in an encoder whose spatial saliency was generated using a 10-layer CNN, and whose temporal saliency was calculated assuming the 2D motion model (resulting in an average of 0.24 BD-PSNR gains over H.265/HEVC reference model (version HM16.8)). Performance improvement due to fine-grained quantization adaptation was reported using an open-source x264 encoder in [159]. This was accomplished by jointly examining the input video frame and associated saliency maps. These saliency maps were generated by utilizing three CNN models suggested in [140], [144], [160]. Up to 25% bit rate reduction was reported when distortion was measured using the edge-weighted SSIM. Similarly, Sun *et al.* [161] implemented a saliency-driven CTU-level adaptive bit rate control, where the static saliency map of each frame was extracted using a DNN model and the dynamic saliency region was tracked using a moving object segmentation algorithm. Experiment results revealed that the PSNR of salient regions was improved by 1.85 dB on average.

Though saliency-based pre-processing is mainly driven by psychovisual studies, it heavily relies on saliency detection to perform UEQ-based adaptive quantization with a lower rate of bit consumption but visually identical reconstruction. On the other hand, visual selectivity behaviour is closely associated with video content distribution (*e.g.*, frequency response), leading to perceptually unequal preference. Thus, it is highly expected that such content semantics-induced discriminative features can be utilized to improve the system efficiency when integrated into the video encoder. To this end, we will discuss the analysis/synthesis-based approach for pre-processing in the next section.

3.2.2 Analysis/Synthesis Based Pre-processing

Since most videos are consumed by human vision, subjective perception of HVS is the *best* way to evaluate quality. However, it is quite difficult to devise a profoundly accurate mathematical HVS model in actual video encoder for rate and perceptual quality optimization, due to the complicated and unclear information processing that occurs in the human visual pathway. Instead, many pioneering psychovisual studies have suggested that neuronal response to compound stimuli is highly nonlinear [162]–[169] within the receptive field. This leads to well-known visual behaviors, such as frequency selectivity, masking, *etc.*, where such stimuli are closely related to the content texture characteristics. Intuitively, video scenes can be broken down into areas that are either "perceptually significant" (*e.g.*, measured in an MSE sense) or "perceptually insignificant". For "perceptually insignificant" regions, users will not perceive compression or processing impairments without a side-by-side comparison with the original sample. This is because the HVS gains semantic understanding by viewing content as a whole, instead of interpreting texture details pixel-by-pixel [170]. This notable effect of the HVS is also referred to as "masking," where visually insignificant information, *e.g.*, perceptually insignificant pixels, will be noticeably suppressed.

In practice, we can first analyze the texture characteristics of original video content in the pre-processing step, *e.g.*, *Texture Analyzer* in Fig. 3.2, in order to sort textures by their significance. Subsequently, we can use any standard compliant video encoder to encode the perceptually significant areas as the main bitstream payload, and apply a statistical model

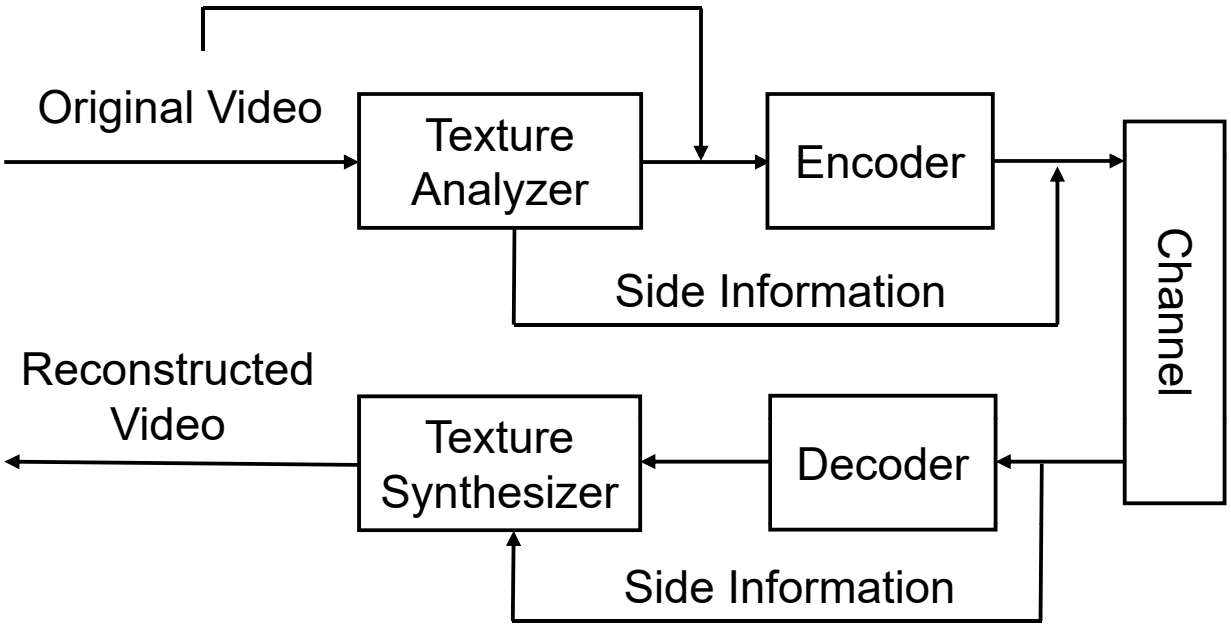


Figure 3.2. Texture Coding System. A general framework of analysis/synthesis based video coding.

to represent the perceptually insignificant textures with model parameters encapsulated as side information. Finally, we can use decoded areas and parsed textures to jointly synthesize the reconstructed sequences in *Texture Synthesizer*. This type of texture modeling makes good use of statistical and psychovisual representation jointly, generally requiring fewer bits, despite yielding visually identical sensation, compared to the traditional hybrid "prediction+residual" method[1]. Therefore, texture analysis and synthesis play a vital role for subsequent video coding. We will discuss related techniques below.

Texture Analysis

Early developments in texture analysis and representation can be categorized into *filter-based* or *statistical modeling-based* approaches. Gabor filter is one typical example of filter-based approaches, by which the input image is convoluted with nonlinear activation for the derivation of corresponding texture representation [172], [173]. At the same time, in order to identify static and dynamic textures for video content, Thakur *et al.* [174] utilized the 2D

[1]↑A comprehensive survey of texture analysis/synthesis based video coding technologies can be found in [171].

dual tree complex wavelet transform and steerable pyramid transform [175], respectively. To accurately capture the temporal variations in video, Bansal *et al.* [176] again suggested the use of optic flow for dynamic texture indication and later synthesis, where optical flow could be generated using temporal filtering. Leveraging statistical models such as the Markovian random field (MRF) [177], [178] is an alternative way to analyze and represent texture. For efficient texture description, such statistical modeling was then extended using handcrafted local features, *e.g.*, the scale invariant feature transform (SIFT) [88], speeded up robust features (SURF) [179], and local binary patterns (LBP) [180]

Recently, stacked DNNs have demonstrated their superior efficiency in many computer vision tasks. This efficiency is mainly due to the powerful capacity of DNN features to be used for video content representation. The most straightforward scheme directly extracted features from the FC6 or FC7 layer of AlexNet [181] for texture representation. Furthermore, Cimpoi *et al.* [182] demonstrated that Fisher vectorized [183] CNN features were a decent texture descriptor candidate. Recent work in [55], a binary CNN based classifier for texture/non-texture is trained by using 32×32 image patches that obtained from the Salzburg Texture Image Database (STex) [184] and Places365 [185]. The CNN architecture and the sample training data are shown in Fig. 3.3 and Fig. 3.4 respectively.

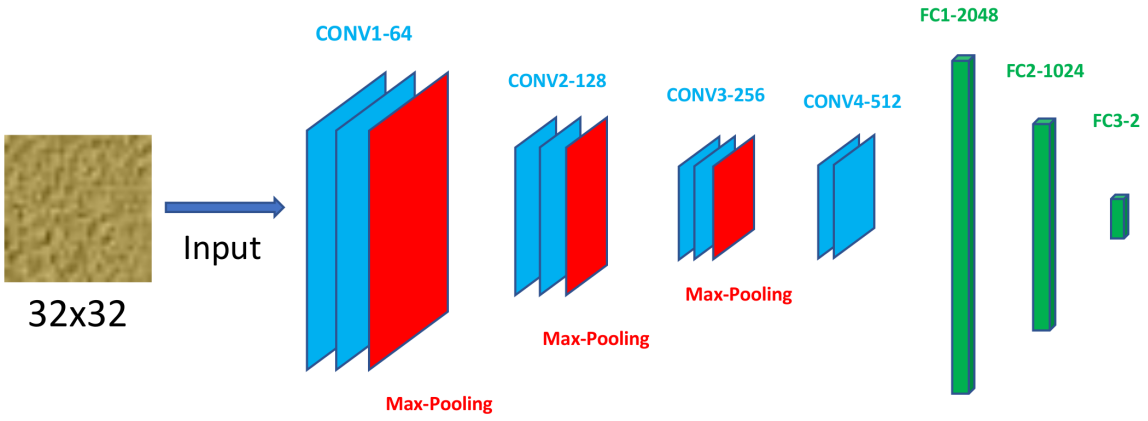


Figure 3.3. CNN architecture for block-based texture classification [55]

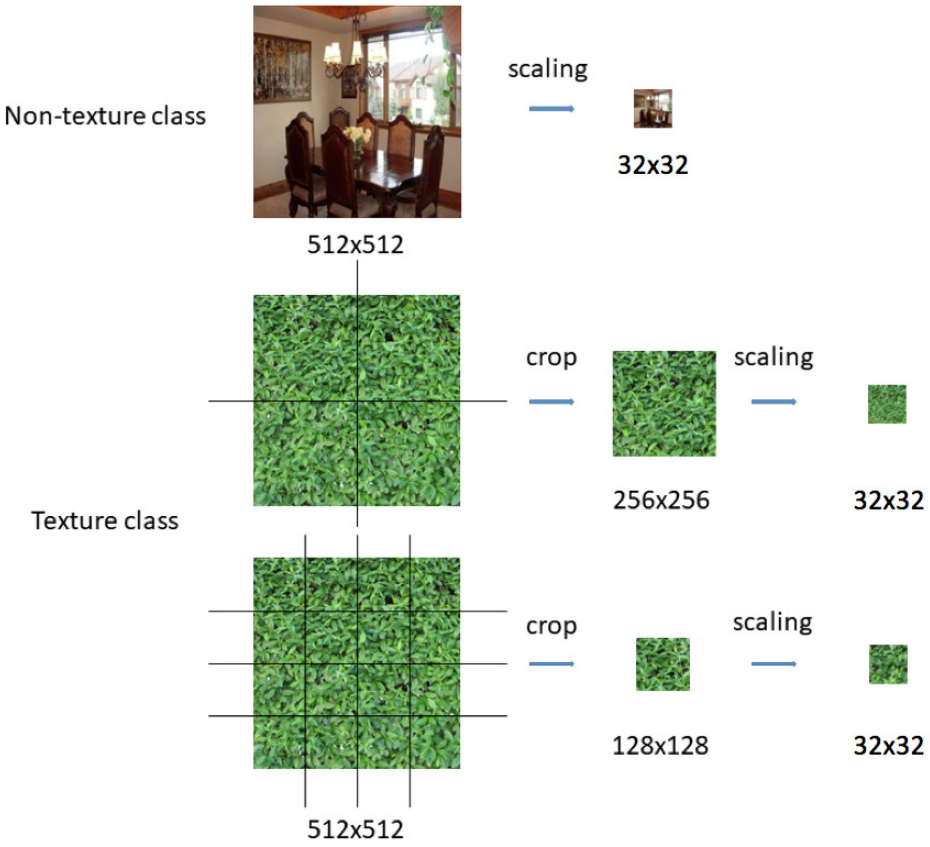


Figure 3.4. Training data preparation [55]

Texture Synthesis

Texture synthesis reverse-engineers the analysis in pre-processing to restore pixels accordingly. It generally includes both non-parametric and parametric methods. For non-parametric synthesis, texture patches are usually resampled from reference images [186]–[188]. In contrast, the parametric method utilizes statistical models to reconstruct the texture regions by jointly optimizing the observation outcomes and the model itself [175], [189], [190].

DNN-based solutions exhibit great potential for texture synthesis applications. One notable example demonstrating this potential used a pre-trained image classification-based

CNN model to generate texture patches [191]. Li [192] then demonstrated that a Markovian GAN-based texture synthesis could offer remarkable quality improvement.

To briefly summarize, earlier "texture analysis/synthesis" approaches often relied on handcrafted models, as well as corresponding parameters. While they have shown good performance to some extent for a set of test videos, it is usually very difficult to generalize them to large-scale video datasets without fine-tuning parameters further. On the other hand, related neuroscience studies propose a broader definition of texture which is more closely related to perceptual sensation, although existing mathematical or data-driven texture representations attempt to fully fulfill such perceptual motives. Furthermore, recent DNN-based schemes present a promising perspective. However, the complexity of these schemes has not yet been appropriately exploited. So, in Section 3.3, we will reveal a CNN-based pixel-level texture analysis approach to segment perceptually insignificant texture areas in a frame for compression and later synthesis. To model the textures both spatially and temporally, we introduce a new coding mode called the "switchable texture mode" that is determined at group of pictures (GoP) level according to the bit rate saving.

3.3 Proposed texture-based video pre-processing

The previous work [55] using block-based texture analyzer has shown data rate savings in texture regions. However, block-based texture mask cannot accurately represents texture regions and may cause coding artifacts. Therefore, in this section, a pixel-level texture mask generation is described to obtain more accurate texture masks. We incorporate semantic scene segmentation into video compression by generating pixel-level texture segmentation masks to represent "perceptually insignificant" regions in a frame and use motion models to reconstruct the texture regions at the decoder to improve the coding efficiency.

3.3.1 Deep learning based pixel level semantic segmentation

Recent advances in deep neural networks have led to a renewed interest in semantic scene segmentation [115], [116], [193]. Large-scale datasest like ImageNet [113], COCO [105] and ADE20K [193] have enabled improved performance for these tasks. For example, the Fully Convolutional Network (FCN) [115] is one of the most commonly used network architectures for semantic scene segmentation. The major issue with FCN [115] is the lack of global contextual information to categories global scene which could lead parsing error. The pyramid scene parsing network(PSPNet) [116] addresses this issue by adding a global pyramid pooling module to extract global information from the image.

In this work, we first rely on the powerful ResNet50 [194] with dilated convolutions [195], [196] to extract feature maps that effectively embed the content semantics. We then introduce the pyramid pooling module from PSPNet [116] to produce a pixel-level semantic segmentation map shown in Fig. 3.5. The pyramid pooling module uses four different sizes of CNN receptive field to represent global contextual information contained in four pyramid scales. The pyramid pooling module reduces the scene parsing errors by considering global contextual relationship in a scene. Cross-entropy loss is used at the end of each stream.

Our implementation starts with a pre-trained PSPNet model generated using the MIT SceneParse150 [197] as a scene parsing benchmark. We then retrain the model on a subset of a densely annotated dataset ADE20K [114]. In the end, the model offers a pixel segmentation accuracy of 80.23%.

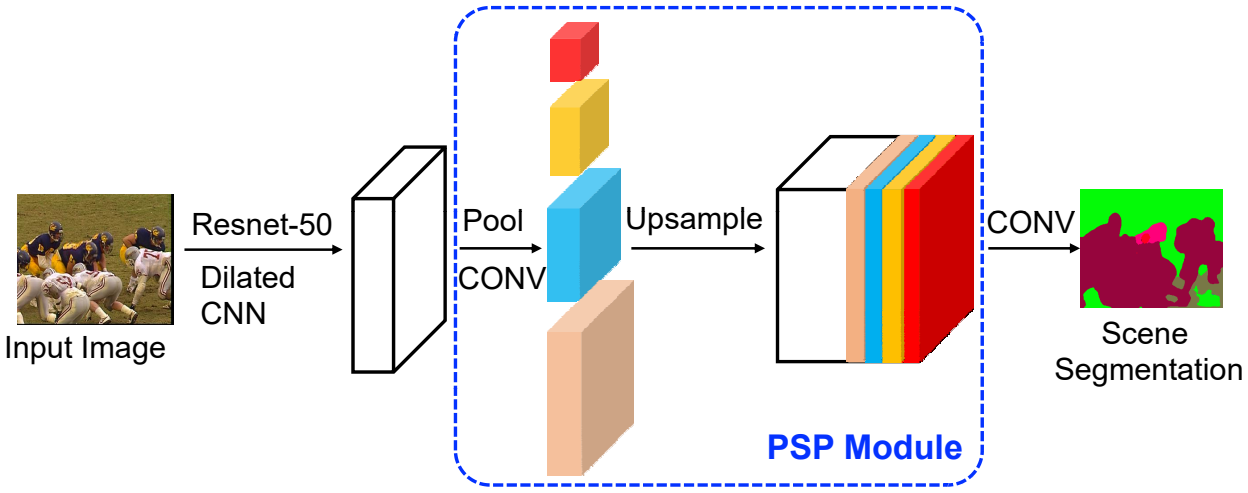


Figure 3.5. Texture Analyzer. Proposed semantic segmentation network using PSPNet [116] and ResNet-50 [194].

3.3.2 Post-processing of pInSIG mask generation

After obtaining semantic segmentation from the PSPNet, a series of post-processing is used to generate the perceptual insignificant mask for AV1 codec.

Grouping by class labels

It is worthwhile to note that the semantic pixel-level segmentation may result in the creation of a number of semantic classes. Nevertheless, this study suggests grouping similar texture classes commonly found in nature scenes together into four major categories, as shown in Table. 3.1, *e.g.*, "earth and grass", "water, sea and river", "mountain and hill", and "tree". The four texture classes are based on groupings of different semantic classes defined in ADE20K dataset [193] that have similar textures. The texture classes grouping criteria should be set accordingly for the training data that used for semantic segmentation. The chosen texture groups in ADE20K dataset fit the need of obtaining the perceptual insignificant region for AV1 codec. Furthermore, each texture category would have an individual segmentation mask to guide the compression performed by the succeeding video encoder.

Table 3.1. Texture classes grouping

Texture group	Class labels
Texture class 1	earth, grass
Texture class 2	water, sea, river
Texture class 3	mountain, hill
Texture class 4	tree

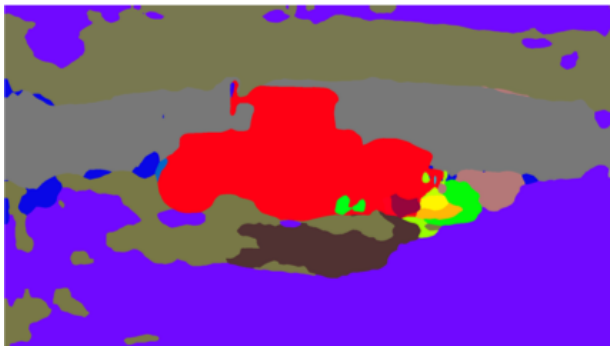
Figure 3.6 shows an example of pixel-level texture segmentation mask for texture class 2, which combines semantic segmentation of water and river. The combined mask is considered as the pixel level perceptual insignificant(pInSIG) mask.

(a) Semantic segmentation

(b) Texture mask for class 2

Figure 3.6. An example of pixel-level texture segmentation for video sequence *bridgefar*. Texture mask for class 2 contains semantic segmentations of water and river in this example.

(a) An example frame in tractor



(b) Semantic segmentation

Figure 3.7. An example frame in video tractor

Class label refinement

Generating pInSIG mask based on semantic class labels is effective, however it also has its limitation. In some scenarios, the generated semantic class labels do not well represent actual perceptual insignificant region in the scene. An example has show in Fig. 3.7, where the upper part of "ground" is classified as "ceiling", and "ceiling" is not belongs to the predefined texture group which leads to a decrease in perceptual insignificant region.

In this section, we introduce a class label refinement process that enlarge the possible perceptual insignificant mask by adjusting the class confidence scores for potential texture regions. A block diagram of the overall refinement process is shown in Fig. 3.8. A scene detection is performed first to split the target video into several splits. Within each video split, we assume the desired class labels are consistent in the same scene. In practice, we use the intra frames selected by AV1 as the scene change frames, which is consistent with scene detection used in switchable-texture mode.

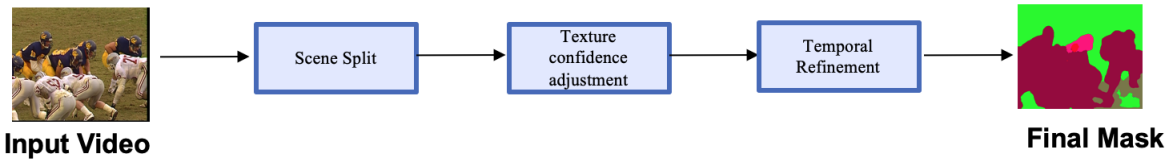


Figure 3.8. Proposed class label refinement process

Within the same scene, texture confidence adjustment is performed to increase the confidence score for potential "texture-like" regions , thus enlarges the pInSIG mask and improve the bit rate saving for compression. In original semantic segmentation, for every pixel p, the top confidence score is used to determine the class label for that pixel. And now, instead of just looking at the top choice, we analyze the top three confidence scores, note as c_1, c_2, c_3 and their corresponding label l_1, l_2, l_3, that are generated by the semantic segmentation. Let the group of texture class be set G. The the new confidences are updated as following:

$$c_1' = c_1 \times w_1 \tag{3.1}$$

$$c_2' = c_2 \times w_2 \tag{3.2}$$

$$c_3' = c_3 \times w_3 \tag{3.3}$$

With the condition that for any $l_\mathrm{i} \in G$, then $w_\mathrm{i} equal sto \theta_t$, others weights equal to θ_{nt}, where θ_t and θ_{nt} are empirically set to be 0.5 and 0.33 respectively.

$$c' = max(c_1', c_2', c_3') \tag{3.4}$$

The the label l' is adjust according to c'.

A temporal refinement is then performed to maintain the consistency of texture segmentation mask and minimize the artifacts throughout the entire video sequence. A median filter with filter size five is used to adjust the class label for each pixel. In other words, the final group label for a pixel is corrected by the majority voting of labels in five consecutive frames.

3.4 Experiments

3.4.1 Texture analysis

Compared to the BM method, the proposed PM method shows larger texture area in general. For the BM method, the fixed size blocks for CNN based texture analyzer need to be large enough to ensure classification accuracy. While for the PM method, there is no such limitation so we use 16×16 as the minimum size for texture blocks instead of 32×32 in the texture mode. Therefore, there are more pixels in a frame that are reconstructed using the texture mode in the PM method leading to larger data rate savings. We did not using smaller texture blocks because further block splitting will require extra bits to send the motion information for these blocks. The PM method with refinement enlarges the texture area especially for the sequence flower and tractor. Because in the sequence flower, some perceptual insignificant area is classified as "flower" in PM method where they further adjust to "ground" after class label refinement. Examples of enlarging the perceptual insignificant area are shown in Fig. 3.9 and Fig. 3.10

And for the sequence tractor, the top area is classified as "ceiling" which also been adjust after the refinement. Table 3.2 shows the texture region percentage, defined in equation 3.5 as average percentage of the regions that are reconstructed using texture mode within the frames where texture mode is enabled.

$$P_{tex} = (\sum_{\mathrm{j}=1}^{F_{tex}} (\frac{\sum_{\mathrm{i}=1}^{N_{\mathrm{j}}} B_{\mathrm{ij}}}{W \times H})) / F_{tex} \times 100\% \tag{3.5}$$

where F_{tex} is the number of frames that enables texture mode, N_{j} is the number of texture blocks in the j^{th} frame, B_{ij} is the block size of texture block i in frame j, W and H are frame width and height.

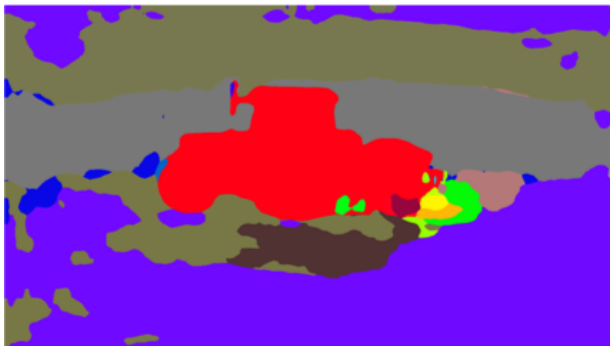
(a) An example frame in flower

(b) Original semantic segmentation

(c) New segmentation after class refinement process

Figure 3.9. An example of refinement result in video flower

(a) An example frame in tractor



(b) Original semantic segmentation

(c) New segmentation after class refinement process

Figure 3.10. An example of refinement result in video tractor

Table 3.2. Texture region percentage

Texture region encoded	BM (%)	PM (%)	PM with refinement (%)
coastguard	37	41	41
flower	58	24	32
football	10	22	22
waterfall	61	77	77
netflix_aerial	37	53	53
intotree	43	52	52
tractor	20	23	38

3.4.2 Coding performance

To evaluate the performance of the proposed method using pixel-level texture mask, data rate savings at four quantization levels (QP=16, 24, 32, 40) are calculated for low and high resolution videos from standard video test sequences. We use the original AV1 codec as the baseline for comparison. We compare the pixel-level texture segmentation results before/after refinement steps with our previous work [55] which uses block-based single class texture mask with our AV1 texture mode. All four methods use the same golden frame group structure that has fixed golden frame and a group interval of eight frames. Results for the test videos are shown in Table ??. BM in the tables refers to block-based texture segmentation method [55] and the PM refers to the proposed pixel-level texture segmentation method, and PMR refers to pixel texture segmentation with refinement. The data rate saving is calculated as

$$P_{bit} = (R - R_b)/R_b \times 100\% \tag{3.6}$$

where P_{bit} represents data rate saving, R represents the bitstream size using BM or PM method, R_b represents the bitstream size of the AV1 baseline method. A negative value indicates a reduction in the codec's bitstream data rate compared to the AV1 baselines.

In general, compared to the AV1 baseline, the coding performance of the both BM and PM shows larger data rate savings with low QP. However, as QP increases, the data rate saving decreases. As shown in the tables, *football*, *waterfall* and *netflix_aerial* have worse coding performance than the AV1 baseline at high QP. The reason is that at high QP, the

high compression ratio results in many zero residual blocks thus there is limited margin for data rate saving using texture based methods. In addition, the texture based method requires a few extra bits for the texture motion parameters, and some extra bits for using two reference frames in compound prediction for all the texture blocks.

Video Sequence	QP=16 (%)			QP=24 (%)			QP=32 (%)			QP=40 (%)		
	BM	PM	PMR	BM	PM	PMR	BM	PM	PMR	BM	PM	PMR
Coastguard	7.80	9.14	9.14	6.99	8.01	8.01	4.70	5.72	5.72	1.90	2.13	2.13
Flower	10.55	13.00	15.21	8.66	10.78	12.32	5.96	4.95	5.67	3.38	1.20	1.43
Waterfall	4.63	13.11	13.11	3.96	7.21	7.21	-0.33	1.30	1.30	-3.74	-3.48	-3.48
Netflix_aerial	8.59	9.15	9.15	2.15	5.59	5.59	-0.68	1.05	1.05	-4.59	-4.01	-4.01
Intotree	5.32	9.71	9.71	4.32	9.42	9.42	1.99	8.46	8.46	-2.83	4.92	4.92
Tractor	3.32	5.45	7.61	2.25	4.40	5.46	1.68	3.90	4.37	0.3	2.93	3.81

Table 3.3. Bit rate saving (%) comparison between block-level DNN (BM) [198] and pixel-level DNN (PM) [199] and pixel-level DNN with refinement(PMR) texture analysis against the AV1 baseline for selected standard test sequences using *tex-allfg* method.

In general the texture region percentage of PM method is larger than that of the BM method, thus the increase in data rate saving. The texture region percentage of PM for *flower* is smaller because the texture mask of BM contains sky and flowerbed area as it fails to identify them as two different classes of texture. Although texture mask of PM only contains flowerbed area, the sky area is very homogeneous which has small residual using AV1 baseline. Therefore, we still achieve more data rate saving using PM than BM. The PM method also reduces flickering artifacts in some of the reconstructed video when using the BM method. The pixel-level texture mask can more accurately represent the perceptually insignificant pixels. An example is illustrated in Figure 1.3 and discussed in Section 1.2. Furthermore, the refinement process provides larger potential texture region as mention in Section 3.4.1 for sequence *flower* and *tractor*, we observe a greater bit rate saving for those two videos.

In addition to test on the standard test sequences, we also selected sequences with texture regions from the YouTube UGC data set [112]. YouTube UGC dataset is a sampling from thousands of User Generated Content (UGC) videos uploaded to YouTube. We calculate the data rate savings at different quantization levels. Table. 3.4 shows the data rate saving using pixel-level texture mask with refinement compare to the AV1 baseline.

3.4.3 Subjective evaluation

Although significant bit rate savings have been achieved compared to the AV1 baseline, it is acknowledged that identical QP values do not necessarily imply the same video quality. We have performed a subjective visual quality study with 20 participants. Reconstructed videos produced by the switchable texture coding with pixel level mask and the baseline AV1 codec at QP = 16, 24, 32, and 40 are arranged randomly and assessed by the participants using a double stimulus continuous quality scale (DSCQS) method [200]. Subjects have been asked to choose among three options: the first video has better visual quality, the second video has better visual quality, or there is no difference between the two versions.

The result of this study is summarized in Figure 3.11. The "Same Quality" indicates the percentage of participants that cannot tell the difference between the reconstructed videos by

Table 3.4. Data rate saving (%) comparison between AV1 baseline and PM methods(tex-switch) on UGC dataset videos. A negative value indicates a reduction in the bitstream data rate compared to the AV1 baseline.

Video Sequence	Resolution	QP=16 PMR	QP=24 PMR	QP=32 PMR	QP=40 PMR
NewsClip_360P-1e1c	360P	10.77	9.27	5.23	1.54
NewsClip_360P-22ce	360P	17.37	15.79	16.37	17.98
TelevisionClip_360P-3b9a	360P	1.45	0.48	0.00	0.00
TelevisionClip_360P-74dd	360P	1.66	1.17	0.36	0.00
HowTo_480P-04f1	480P	3.81	2.57	0.93	0.36
HowTo_480P-4c99	480P	2.36	1.67	0.00	0.00
MusicVideo_480P-1eee	480P	3.31	3.29	2.53	-0.30
NewsClip_480P-15fa	480P	6.31	5.79	0.11	0.03
NewsClip_480P-7a0d	480P	11.54	10.03	1.53	0.00
TelevisionClip_480P-19d3	480P	3.13	2.86	1.66	0.00
HowTo_720P-0b01	720P	12.72	11.84	9.31	6.35
MusicVideo_720P-3698	720P	1.76	1.07	0.30	0.00
MusicVideo_720P-4ad2	720P	6.93	3.81	1.87	0.11
HowTo_1080P-4d7b	1080P	7.31	6.07	3.21	0.72
MusicVideo_1080P-55af	1080P	3.88	1.78	0.33	-0.68

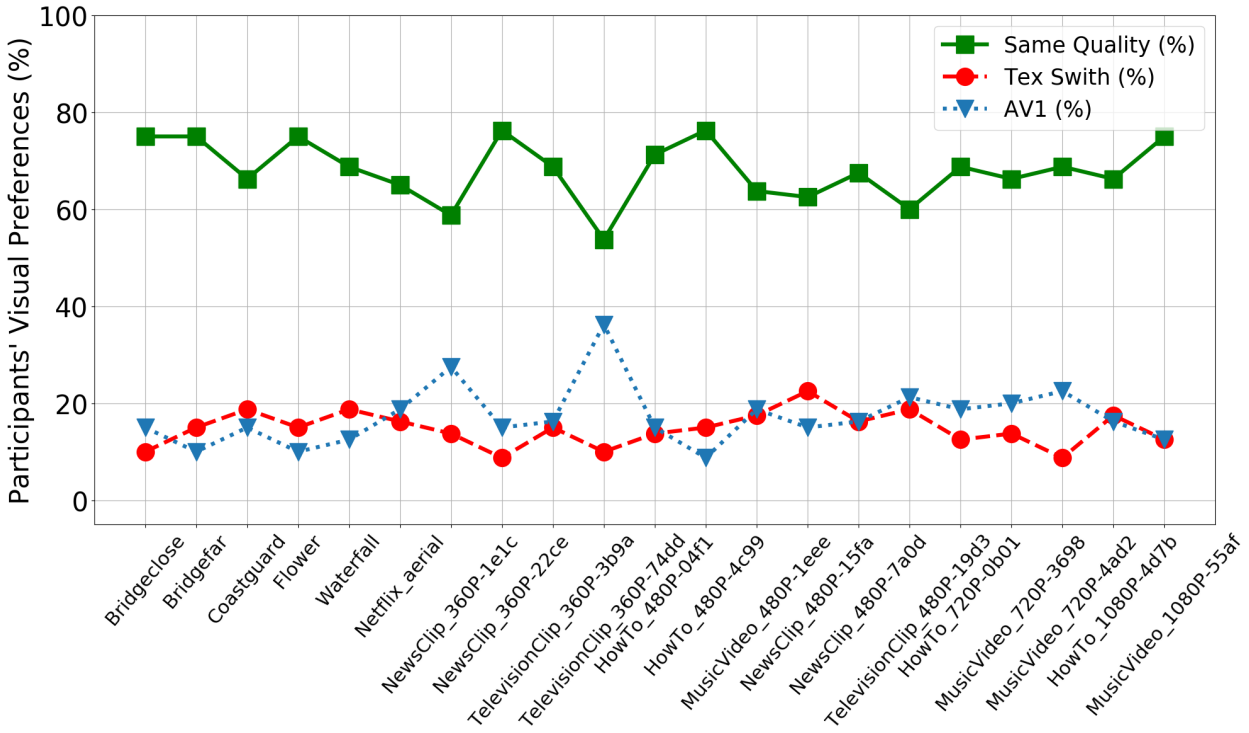


Figure 3.11. Subjective evaluation of visual preference. Results show average subjective preference (%) for QP = 16, 24, 32, 40 compared between AV1 baseline and proposed switchable texture mode.

the AV1 baseline codec and the proposed method *tex-switch* (69.03% on average). The term "tex-switch" indicates the percentage of participants that prefer the reconstructions by the proposed method *tex-switch* (14.32% on average); and the "AV1" indicates the percentage of participants who think the visual quality of the reconstructed videos using the AV1 baseline is better (16.65% on average).

We observe that the results are sequence dependent and both spatial and temporal artifacts can appear in the reconstructed videos. The main artifacts come from the inaccurate pixel-based texture mask. For example, in some frames of *TelevisionClip_360P-74dd* sequence, the texture masks include parts of the moving objects in the foreground, which are reconstructed using texture mode. Since the motion of the moving objects is different from the motion of the texture area, there are noticeable artifacts around those parts of the frame. To further improve the accuracy of region analysis using DNN-based pre-processing, we plan

to incorporate an in-loop perceptual visual quality metric for optimization during the texture analysis and reconstruction.

4. SUMMARY

4.1 Touch event detection

In this thesis, we propose an automatic touch event recognition method to determine the time when the caregiver touches the infant and label it as a "touch event" by analyzing the locations of the caregiver's hands and the infant's contour.

1. We described a touch event detection system that combines hand tracking and infant segmentation and analyzing the merging contours of the caregiver's hands and the infant's contour.

2. We propose a 2D hand tracking method using color and motion features with occlusion handling.

3. We propose a long-term 2D hand tracking method with re-initialization capability using a hand detection network and a human pose estimation network.

4. We propose a challenging hand tracking dataset with annotations for long sequences in which the subjects are mostly in profile view. The dataset also contains difficult scenarios such as hand interaction, hand occlusions and hand vanishing.

5. We propose a pixel-wise Grab-cut segmentation method with updated mask using morphological operations and a super-pixel based Grab-cut segmentation method using skeleton information as foreground mask.

6. We propose an automatic touch event detection and recognition method to determine the potential timing when the caregiver touches the infant, and classify the event into six touch types based on which body part of the infant has been touched.

We proposed an automatic touch event detection system that detects and tracks the caregiver's hands, detects the location of the infant and then defines a "touch" to occur whenever the caregiver's hand contours overlap with the infants contour. The touch type label is assigned to predicted touch frames based on the spatial relationship between caregiver's hands and infant body parts. The proposed method avoids using expensive precise

touch annotations for training. Instead it takes advantage of CNN models that are trained on large public datasets to produce intermediate results needed to identify touches. The proposed method allows trained analyst skip annotating 58.41% of frames and still be able to capture more than 99% true touch frames.

The touch detection system is designed specifically for the touch project from Purdue infant speech lab. However, the tracking method used in this project and the problem solving approaches could be broadly applied. We explain this from three aspects: 1) dataset; 2) general human interactions detection; 3) other types of action analysis.

Dataset: For the touch detection project, the experimental data is captured in a controlled environment, *i.e.*, a sound proof room at the Purdue Infant Speech Lab where infant speech related studies are typically conducted. Our overall approach works well for this particular type of data, and key components of the system also performs well on public dataset compared to other methods. For example, the proposed long term hand tracking method with re-initialization capability not only performs well with our own dataset, but is also robust to many general scenarios. In [201], experiments have been done to compare our hand tracking method with method proposed in [109] on a public hand tracking dataset [109] with typical room setting and background. Our method with auto re-initialization performed comparably to the semi-automatic method with manual hand re-initialization [109] on that public dataset.

General human interaction detection: This project aims to detect a specific type of action "touch" in human interaction. Analyzing the positions and trajectories of human could also be used in other action detection and recognition tasks. Taking the video surveillance as an example, detecting whether a scene contains fighting is a common task [202]–[204]. Both the tracking techniques and the human pose estimation techniques could be adopted to fight detection, as tracking human body joints is also a key component to identify fighting.

Other types of action analysis: Our proposed methods can also be adopted for other types of action analysis. For example, there is an increasing need to automate the monitoring of animal behavior [205]–[208]. Similar concepts of detecting and tracking skeleton and keypoints to analyze actions could be transferred to analyze behaviors of animals. In addition, skeleton detection may not always provide sufficiently accurate information for

animal's joints in each frame. Thus, combining skeleton keypoints detection with traditional tracking techniques used in this project, to track feature points from frame to frame, could provide continuous analysis in cases when skeleton detection fails.

The touch detection project guided by Prof. Zhu, was my first individual project which kicked off my Ph.D. research. We started by treating the touch detection as a frame level two classes classification problem. However, directly using entire frame as input did not lead to good results. Then we broke down the problem into two parts, finding caregiver's hand localization and infant's position. We built and improved several methods for coarse to fine analysis. The first touch detection method with contour merging uses feature based approach. It was able to detect touches, however, detection error would propagate whenever the hand tracker loses track. This situation was improved when we proposed a long term hand tracking method with re-initialization using human pose estimation. Furthermore, we extended the usages of human pose estimation to identifying six different touch types.

4.2 Texture analysis for Video compression

For the texture analysis for video compression project, our contributions are as below:

1. We combined semantic segmentation with a few post-processing steps to generate a pixel-level texture mask that is more accurate than our previously proposed block-based texture method.

2. The proposed texture analysis method is integrated into a switchable texture-based video coding method.

3. We show that for many standard test sets and user generated test sequences, the proposed method achieves significant data rate reductions with improved visual quality.

We proposed a DNN based texture pre-processing for texture analysis/synthesis coding tool in AV1 codec. Experimental results show that our proposed method can achieve noticeable bit rate reduction with satisfying visual quality for both standard test sets and user generated contents, which is verified by a subjective study. We envision that video coding driven by semantic understanding will continue to improve in terms of both quality and bit

rate, especially by leveraging advances of deep learning methods. However, there remain several open challenges that require further investigation.

Accuracy of region analysis is one of the major challenges for integrating semantic understanding into video coding. However, recent advances in scene understanding have significantly improved the performance of region analysis. Visual artifacts are still noticeable when a non-texture region is incorrectly included in the texture mask, particularly if the analysis/synthesis coding system is open loop. One potential solution is to incorporate some perceptual visual quality measures in-loop during the texture region reconstruction.

Benchmark video segmentation datasets are important for developing machine learning methods for video based semantic understanding. Existing segmentation datasets are either based on images with texture [209], or contain general video objects only [210], [211], or focus on visual quality but lack segmentation ground truth.

From Section 1.4.1 in Di's thesis [212], we observed that the memory and computation efficiency are still challenging for deploying module based video coding in practical system. As for the memory and computational efficiency for our texture analysis, it requires 197MB for model storage. The proposed method processes around 5 frames per second for a CIF resolution video on an NVIDIA GTX 1080 Ti GPU, which is four times slower than the block-level texture analysis [55]. There exists a trade-off between the accuracy of the texture representation and computational complexity. To address this challenge, future work needs to explore more compact and efficient networks structures, and to deploy on deep neural network accelerators for efficient processing.

REFERENCES

[1] E. Anisfeld, V. Casper, M. Nozyce, and N. Cunningham, "Does infant carrying promote attachment? An experimental study of the effects of increased physical contact on the development of attachment," *Child Development*, vol. 61, no. 5, pp. 1617–1627, Oct. 1990.

[2] R. Feldman, M. Singer, and O. Zagoory, "Touch attenuates infants' physiological reactivity to stress," *Developmental Science*, vol. 13, no. 2, pp. 271–278, Mar. 2010.

[3] S. G. Ferber, "The nature of touch in mothers experiencing maternity blues: The contribution of parity," *Early Human Development*, vol. 79, no. 1, pp. 65–75, Aug. 2004.

[4] S. G. Ferber, R. Feldman, and I. R. Makhoul, "The development of maternal touch across the first year of life," *Early Human Development*, vol. 84, no. 6, pp. 363–370, Jun. 2008.

[5] F. Franco, A. Fogel, D. S. Messinger, and C. A. Frazier, "Cultural differences in physical contact between hispanic and anglo mother–infant dyads living in the united states," *Early Development and Parenting*, vol. 5, no. 3, pp. 119–127, May 1996.

[6] E. Herrera, N. Reissland, and J. Shepherd, "Maternal touch and maternal child-directed speech: Effects of depressed mood in the postnatal period," *Journal of Affective Disorders*, vol. 81, no. 1, pp. 29–39, Jul. 2004.

[7] M. J. Hertenstein, "Touch: Its communicative functions in infancy," *Human Development*, vol. 45, no. 2, pp. 70–94, Mar. 2002.

[8] A. D. Jean and D. M. Stack, "Functions of maternal touch and infants' affect during face-to-face interactions: New directions for the still-face," *Infant Behavior and Development*, vol. 32, no. 1, pp. 123–128, Jan. 2009.

[9] A. D. Jean, D. M. Stack, and A. Fogel, "A longitudinal investigation of maternal touching across the first 6 months of life: Age and context effects," *Infant Behavior and Development*, vol. 32, no. 3, pp. 344–349, Jun. 2009.

[10] R. Moszkowski and D. M. Stack, "Infant touching behavior during mother-infant face-to-face interactions," *Infant and Child Development*, vol. 16, no. 3, pp. 307–319, Jun. 2007.

[11] D. W. Muir, "Adult communications with infants through touch: The forgotten sense," *Human Development*, vol. 45, no. 2, pp. 95–99, Mar. 2002.

[12] I. Nomikou and K. J. Rohlfing, "Language does something: Body action and language in maternal input to three-month-olds," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 2, pp. 113–128, Jun. 2011.

[13] D. M. Stack and S. L. Arnold, "Changes in mothers' touch and hand gestures influence infant behavior during face-to-face interchanges," *Infant Behavior and Development*, vol. 21, no. 3, pp. 451–468, Jun. 1998.

[14] A. D. Jean and D. M. Stack, "Full-term and very-low-birth-weight preterm infants' self-regulating behaviors during a still-face interaction: Influences of maternal touch," *Infant Behavior and Development*, vol. 35, no. 4, pp. 779–791, Dec. 2012.

[15] D. M. Stack and D. W. Muir, "Tactile stimulation as a component of social interchange: New interpretations for the still-face effect," *British Journal of Developmental Psychology*, vol. 8, no. 2, pp. 131–145, Jun. 1990.

[16] S. J. Weiss, P. Wilson, M. J. Hertenstein, and R. Campos, "The tactile context of a mother's caregiving: Implications for attachment of low birth weight infants ," *Infant Behavior and Development*, vol. 23, no. 1, pp. 91–111, 2000.

[17] R. Feldman, A. I. Eidelman, L. Sirota, and A. Weller, "Comparison of skin-to-skin (kangaroo) and traditional care: Parenting outcomes and preterm infant development," *Pediatrics*, vol. 110, no. 1, pp. 16–26, 2002.

[18] S. J. Weiss, P. Wilson, M. S. J. Seed, and S. M. Paul, "Early tactile experience of low birth weight children: Links to later mental health and social adaptation," *Infant and Child Development*, vol. 10, no. 3, pp. 93–115, 2001.

[19] R. Feldman, Z. Rosenthal, and A. I. Eidelman, "Maternal-preterm skin-to-skin contact enhances child physiologic organization and cognitive control across the first 10 years of life," *Biological psychiatry*, vol. 75, no. 1, pp. 56–64, 2014.

[20] J. Van de Weijer, "Language input for word discovery," 1999.

[21] E. K. Johnson, "Constructing a proto-lexicon: An integrative view of infant language development," *Annual Review of Linguistics*, vol. 2, pp. 391–412, 2016.

[22] A. Seidl, R. Tincoff, C. Baker, and A. Cristia, "Why the body comes first: Effects of experimenter touch on infants' word finding," *Developmental Science*, vol. 18, no. 1, pp. 155–164, 2015.

[23] R. Abu-Zhaya, A. Seidl, and A. Cristia, "Multimodal infant-directed communication: How caregivers combine tactile and linguistic cues," *Journal of Child Language*, To appear.

[24] H. Brugman, A. Russel, and X. Nijmegen, "Annotating multi-media/multi-modal resources with elan.," 2004.

[25] J. D. Goldman and U. K. Padayachi, "Some methodological problems in estimating incidence and prevalence in child sexual abuse research," *Journal of Sex Research*, vol. 37, no. 4, pp. 305–314, 2000.

[26] D. P. Southall, M. C. Plunkett, M. W. Banks, A. F. Falkov, and M. P. Samuels, "Covert video recordings of life-threatening child abuse: Lessons for child protection," *Pediatrics*, vol. 100, no. 5, pp. 735–760, 1997.

[27] Y. Li, J. Sun, C. Tang, and H. Shum, "Lazy snapping," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 303–308, Aug. 2004.

[28] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. S. Kittle, D. R. Golish, E. Vera, and S. D. Feller, "Multiscale gigapixel photography," *Nature*, vol. 486, no. 7403, pp. 386–389, 2012.

[29] M. Cheng, Z. Ma, S. Asif, Y. Xu, H. Liu, W. Bao, and J. Sun, "A dual camera system for high spatiotemporal resolution video acquisition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 01, pp. 1–1, 2020.

[30] F. Dufaux, P. Le Callet, R. Mantiuk, and M. Mrak, *High dynamic range video: from acquisition, to display and applications*. Academic Press, 2016.

[31] M. Winken, D. Marpe, H. Schwarz, and T. Wiegand, "Bit-depth scalable video coding," in *2007 IEEE International Conference on Image Processing*, IEEE, vol. 1, 2007, pp. I–5.

[32] P. Tudor, "Mpeg-2 video compression," *Electronics & communication engineering journal*, vol. 7, no. 6, pp. 257–264, 1995.

[33] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital video: an introduction to MPEG-2*. Springer Science & Business Media, 1996.

[34] T. Sikora, "The mpeg-4 video standard verification model," *IEEE Transactions on circuits and systems for video technology*, vol. 7, no. 1, pp. 19–31, 1997.

[35] W. Li, "Overview of fine granularity scalability in mpeg-4 video standard," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 301–317, 2001.

[36] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[37] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[38] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (hevc)," in *Integrated circuit and systems, algorithms and architectures*, vol. 39, Springer, 2014, pp. 49–90.

[39] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions," in *Applications of Digital Image Processing XXVII*, International Society for Optics and Photonics, vol. 5558, 2004, pp. 454–474.

[40] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the h. 264/mpeg-4 avc standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, 2011.

[41] L. Yu, S. Chen, and J. Wang, "Overview of AVS-video coding standards," *Signal processing: Image communication*, vol. 24, no. 4, pp. 247–262, 2009.

[42] S. Ma, S. Wang, and W. Gao, "Overview of ieee 1857 video coding standard," in *2013 IEEE International Conference on Image Processing*, IEEE, 2013, pp. 1500–1504.

[43] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao, "Recent development of avs video coding standard: AVS3," in *2019 Picture Coding Symposium (PCS)*, IEEE, 2019, pp. 1–5.

[44] Y. Chen, D. Murherjee, J. Han, A. Grange, Y. Xu, Z. Liu, S. Parker, C. Chen, H. Su, U. Joshi, *et al.*, "An overview of core coding tools in the av1 video codec," in *2018 Picture Coding Symposium (PCS)*, IEEE, 2018, pp. 41–45.

[45] J. Han, B. Li, D. Mukherjee, C.-H. Chiang, C. Chen, H. Su, S. Parker, U. Joshi, Y. Chen, Y. Wang, *et al.*, "A technical overview of av1," *arXiv preprint arXiv:2008.06091*, 2020.

[46] *AOM - alliance for open media*, http://www.aomedia.org/.

[47] A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, *Per-Title Encode Optimization*, The Netflix Tech Blog, https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2, (14 December 2015).

[48] T. Shoham, D. Gill, S. Carmel, N. Terterov, and P. Tiktov, "Content-adaptive frame level rate control for video encoding using a perceptual video quality measure," in *Applications of Digital Image Processing XLII*, A. G. Tescher and T. Ebrahimi, Eds., vol. 11137, Sep. 2019, p. 26.

[49] Y.-C. Lin, H. Denman, and A. Kokaram, "Multipass encoding for reducing pulsing artifacts in cloud based video transcoding," in *2015 IEEE International Conference on Image Processing*, Sep. 2015, pp. 907–911.

[50] Y. Chen, D. Murherjee, J. Han, A. Grange, Y. Xu, Z. Liu, S. Parker, C. Chen, H. Su, U. Joshi, *et al.*, "An overview of core coding tools in the AV1 video codec," in *2018 Picture Coding Symposium*, IEEE, 2018, pp. 41–45.

[51] M. Bosch, F. Zhu, and E. J. Delp, "Segmentation-Based Video Compression Using Texture and Motion Models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1366–1377, Nov. 2011.

[52] J. Balle, A. Stojanovic, and J. Ohm, "Models for static and dynamic texture synthesis in image and video compression," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1353–1365, Nov. 2011, ISSN: 1932-4553. DOI: 10.1109/JSTSP.2011.2166246.

[53] F. Zhang and D. R. Bull, "A parametric framework for video compression using region-based texture models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, Nov. 2011.

[54] F. Racapé, O. Déforges, M. Babel, and D. Thoreau, "Spatiotemporal texture synthesis and region-based motion compensation for video compression," *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 993–1005, Oct. 2013. DOI: 10.1016/j.image.2013.05.004. [Online]. Available: https://hal.inria.fr/hal-00921485.

[55] D. Chen, C. Fu, and F. Zhu, "AV1 video coding using texture analysis with convolutional neural networks," *ArXiv e-prints*, Apr. 2018. arXiv: 1804.09291.

[56] "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," *CRCV-TR-12-01*, 2012, University of Central Florida, Orlando, FL.

[57] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2929–2936, 2009.

[58] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3169–3176, 2011.

[59] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[60] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[61] C. Chen, M. Zhang, K. Qiu, and Z. Pan, "Real-time robust hand tracking based on camshift and motion velocity," *Proceedings of the IEEE International Conference on Digital Home*, pp. 20–24, Nov. 2014, Guangzhou, China.

[62] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 91–99, Dec. 2015, Montreal, Canada.

[63] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," *Proceedings of the European Conference on Computer Vision*, pp. 483–499, Sep. 2016, Amsterdam, The Netherlands.

[64] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677–695, Jul. 1997.

[65] N. H. Do and K. Yanai, "Hand detection and tracking in videos for fine-grained action recognition," *Proceedings of the Asian Conference on Computer Vision Workshops*, pp. 19–34, Nov. 2014, Singapore, Singapore.

[66] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," *Computational Imaging and Vision*, vol. 9, pp. 227–243, 1997.

[67] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," *Proceedings of the British Machine Vision Conference*, pp. 101.1–101.11, Sep. 2011, Nethergate, UK.

[68] B. Stenger, P. R. Mendonça, and R. Cipolla, "Model-based 3d tracking of an articulated hand," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 310–315, Dec. 2001, Kauai, HI.

[69] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM Transactions on Graphics*, vol. 28, no. 3, 2009.

[70] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*, pp. 214–219, Oct. 1998, Princeton, NJ.

[71] A. Mittal, A. Zisserman, and P. H. Torr, "Hand detection using multiple proposals," *Proceedings of the British Machine Vision Conference*, pp. 75.1–75.11, Sep. 2011, Nethergate, UK.

[72] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[73] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1949–1957, Dec. 2015, Santiago, Chile.

[74] T. H. N. Le, K. G. Quach, C. Zhu, C. N. Duong, K. Luu, and M. Savvides, "Robust hand detection and classification in vehicles and in the wild," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1203–1210, Jul. 2017, Honolulu, Hawaii.

[75] J. R. Zhang and J. R. Kender, "Recognizing and tracking clasping and occluded hands," *Proceedings of the IEEE International Conference on Image Processing*, pp. 2817–2821, 2013.

[76] V. Spruyt, A. Ledda, and W. Philips, "Real-time, long-term hand tracking with unsupervised initialization," *Proceedings of the IEEE International Conference on Image Processing*, pp. 3730–3734, Sep. 2013, Melbourne, Australia.

[77] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, Jan. 2002.

[78] Q. Zhu, C.-T. Wu, K.-T. Cheng, and Y.-L. Wu, "An adaptive skin model and its application to objectionable image filtering," pp. 56–63, 2004.

[79] S. L. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color pixel classification: Analysis and comparison," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 1, pp. 148–154, 2005.

[80] B. K. Horn and B. G. Schunck, "Determining optical flow," *Journal of Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[81] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the Fourth Alvey Vision Conference*, vol. 15, p. 50, Aug. 1988, Manchester, UK.

[82] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Technical Report*, 1999, Intel Corporation, Santa Clara, CA.

[83] T. Yang, Q. Pan, J. Li, and S. Z. Li, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 970–975, Jun. 2005, San Diego, CA.

[84] M. Cai, K. M. Kitani, and Y. Sato, "Understanding hand-object manipulation with grasp types and object attributes," *Proceedings of the Robotics: Science and Systems Conference*, pp. 1–10, Jun. 2016, Ann Arbor, Michigan.

[85] I. M. Bullock, T. Feix, and A. M. Dollar, "The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 251–255, 2015.

[86] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, Jun. 2014, Columbus, Ohio.

[87] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1799–1807, Dec. 2014, Montreal,Canada.

[88] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[89] Q. Chen, H. Li, R. Abu-Zhaya, A. Seidl, F. Zhu, and E. J. Delp, "Touch event recognition for human interaction," *Proceedings of IS&T International Symposium on Electronic Imaging*, vol. 2016, no. 11, pp. 1–6, Feb. 2016, San Francisco, CA.

[90] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[91] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[92] J. D. R. Girshick, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Jun. 2014, Columbus, OH.

[93] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1097–1105, Dec. 2012, Lake Tahoe, NV.

[94] R. Girshick, "Fast R-CNN," *Proceedings of the International Conference on Computer Vision*, pp. 1440–1448, Dec. 2015, Santiago, Chile.

[95] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Proceedings of the European Conference on Computer Vision*, pp. 818–833, Sep. 2014, Zürich, Switzerland.

[96] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[97] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," pp. 3674–3681, 2013.

[98] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Strong appearance and expressive spatial models for human pose estimation," pp. 3487–3494, 2013.

[99] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 67–92, 1973.

[100] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," pp. 1014–1021, 2009.

[101] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2014, Columbus, Ohio.

[102] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, Aug. 2004.

[103] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[104] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields," *arXiv preprint arXiv:1812.08008*, 2018.

[105] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *Proceedings of the IEEE European Conference on Computer Vision*, pp. 740–755, Sep. 2014, Zürich, Switzerland.

[106] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017.

[107] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[108] S. D. Beugher, G. Brône, and T. Goedemé, "Semi-automatic hand detection: A case study on real life mobile eye-tracker data," *Proceedings of the International Conference on Computer Vision Theory and Applications*, vol. 2, pp. 121–129, Mar. 2015, Berlin, Germany.

[109] S. D. Beugher, G. Brône, and T. Goedemé, "Semi-automatic hand annotation making human-human interaction analysis fast and accurate," *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 552–559, Feb. 2016, Rome, Italy.

[110] U. Joshi, D. Mukherjee, J. Han, Y. Chen, S. Parker, H. Su, A. Chiang, Y. Xu, Z. Liu, Y. Wang, *et al.*, "Novel inter and intra prediction tools under consideration for the emerging av1 video codec," *Applications of Digital Image Processing XL*, vol. 10396, 103960F, 2017.

[111] Z. Liu, D. Mukherjee, W. Lin, P. Wilkins, J. Han, and Y. Xu, "Adaptive multi-reference prediction using a symmetric framework," *Electronic Imaging*, vol. 2017, no. 2, pp. 65–72, 2017.

[112] Y. Wang, S. Inguva, and B. Adsumilli, "YouTube UGC dataset for video compression research," *IEEE International Workshop on Multimedia Signal Processing*, Sep. 2019, Kuala Lumpur, Malaysia.

[113] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[114] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.

[115] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Jun. 2015, Boston, MA.

[116] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Jul. 2017, Honolulu, HI.

[117] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[118] C. Tian, Y. Xu, L. Fei, and K. Yan, "Deep learning for image denoising: A survey," in *International Conference on Genetic and Evolutionary Computing*, Springer, 2018, pp. 563–572.

[119] A. Chakrabarti, "A neural approach to blind motion deblurring," in *European conference on computer vision*, Springer, 2016, pp. 221–235.

[120] J. Koh, J. Lee, and S. Yoon, "Single-image deblurring with neural networks: A comparative survey," *Computer Vision and Image Understanding*, p. 103 134, 2020.

[121] Y. Zhu, X. Fu, and A. Liu, "Learning dual transformation networks for image contrast enhancement," *IEEE Signal Processing Letters*, 2020.

[122] W. Guan, T. Wang, J. Qi, L. Zhang, and H. Lu, "Edge-aware convolution neural network based salient object detection," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 114–118, 2018.

[123] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *International Conference on Machine Learning*, 2015, pp. 1669–1678.

[124] L. Zhaoping, "A new framework for understanding vision from the perspective of the primary visual cortex," *Current opinion in neurobiology*, vol. 58, pp. 1–10, 2019.

[125] X. Chen, M. Zirnsak, G. M. Vega, E. Govil, S. G. Lomber, and T. Moore, "The contribution of parietal cortex to visual salience," *bioRxiv*, 2019, doi: http://doi.org/10.1101/619643.

[126] O. Schwartz and E. Simoncelli, "Natural signal statistics and sensory gain control.," *Nature neuroscience*, vol. 4, no. 8, p. 819, 2001.

[127] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[128] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE transactions on image processing*, vol. 13, no. 10, pp. 1304–1318, 2004.

[129] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan, "Static saliency vs. dynamic saliency: A comparative study," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 987–996.

[130] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2798–2805.

[131] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 362–370.

[132] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[133] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 678–686.

[134] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203–3212.

[135] B. Yan, H. Wang, X. Wang, and Y. Zhang, "An accurate saliency prediction method based on generative adversarial networks," in *2017 IEEE International Conference on Image Processing*, IEEE, 2017, pp. 2339–2343.

[136] Y. Xu, S. Gao, J. Wu, N. Li, and J. Yu, "Personalized saliency and its prediction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 12, pp. 2975–2989, 2018.

[137] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," *arXiv preprint arXiv:1603.08199*, 2016.

[138] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1688–1698, 2017.

[139] M. Sun, Z. Zhou, Q. Hu, Z. Wang, and J. Jiang, "SG-FCN: A motion and memory-based deep learning model for video saliency detection," *IEEE transactions on cybernetics*, vol. 49, no. 8, pp. 2900–2911, 2018.

[140] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "Deepvs: A deep learning based video saliency prediction approach," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 602–617.

[141] Z. Wang, J. Ren, D. Zhang, M. Sun, and J. Jiang, "A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos," *Neurocomputing*, vol. 287, pp. 68–83, 2018.

[142] R. Cong, J. Lei, H. Fu, F. Porikli, Q. Huang, and C. Hou, "Video saliency detection via sparsity-based reconstruction and propagation," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 4819–4831, 2019.

[143] K. Min and J. J. Corso, "Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2394–2403.

[144] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2019.

[145] A. Vetro, T. Haga, K. Sumi, and H. Sun, "Object-based coding for long-term archive of surveillance video," *Proceedings of International Conference on Multimedia and Expo*, vol. 2, p. 417, Jul. 2003, Baltimore, MD.

[146] T. Nishi and H. Fujiyoshi, "Object-based video coding using pixel state analysis," *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 306–309, Aug. 2004, Cambridge, UK.

[147] L. Zhu and Q. Zhang, "Motion-based foreground extraction in compressed video," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, IEEE, vol. 2, 2010, pp. 711–714.

[148] Z. Zhang, T. Jing, J. Han, Y. Xu, and X. Li, "Flow-process foreground region of interest detection method for video codecs," *IEEE Access*, vol. 5, pp. 16 263–16 276, 2017.

[149] Y. Guo, Z. Xuan, and L. Song, "Foreground target extraction method based on neighbourhood pixel intensity correction," *Australian Journal of Mechanical Engineering*, pp. 1–10, 2019.

[150] A. Shahbaz, V.-T. Hoang, and K.-H. Jo, "Convolutional neural network based foreground segmentation for video surveillance systems," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, vol. 1, 2019, pp. 86–89.

[151] S. Zhou, J. Wang, D. Meng, Y. Liang, Y. Gong, and N. Zheng, "Discriminative feature learning with foreground attention for person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4671–4684, 2019.

[152] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for background subtraction," *arXiv preprint arXiv:1702.01731*, 2017.

[153] X. Liang, S. Liao, X. Wang, W. Liu, Y. Chen, and S. Z. Li, "Deep background subtraction with guided learning," in *2018 IEEE International Conference on Multimedia and Expo*, IEEE, 2018, pp. 1–6.

[154] S. Zhang, K. Wei, H. Jia, X. Xie, and W. Gao, "An efficient foreground-based surveillance video coding scheme in low bit-rate compression," in *2012 Visual Communications and Image Processing*, IEEE, 2012, pp. 1–6.

[155] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2013.

[156] Y. Li, W. Liao, J. Huang, D. He, and Z. Chen, "Saliency based perceptual HEVC," in *2014 IEEE International Conference on Multimedia and Expo Workshops*, IEEE, 2014, pp. 1–5.

[157] C. Ku, G. Xiang, F. Qi, W. Yan, Y. Li, and X. Xie, "Bit allocation based on visual saliency in HEVC," in *2019 IEEE Visual Communications and Image Processing*, IEEE, 2019, pp. 1–4.

[158] S. Zhu and Z. Xu, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing*, vol. 275, pp. 511–522, 2018.

[159] V. Lyudvichenko, M. Erofeev, A. Ploshkin, and D. Vatolin, "Improving video compression with deep visual-attention models," in *Proceedings of the 2019 International Conference on Intelligent Medicine and Image Processing*, 2019, pp. 88–94.

[160] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an lstm-based saliency attentive model," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5142–5154, 2018.

[161] X. Sun, X. Yang, S. Wang, and M. Liu, "Content-aware rate control scheme for HEVC based on static and dynamic saliency detection," *Neurocomputing*, 2020.

[162] M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. A. Olshausen, J. L. Gallant, and N. C. Rust, "Do we know what the early visual system does?" *Journal of Neuroscience*, vol. 25, no. 46, pp. 10 577–10 597, 2005.

[163] J. Kremkow, J. Jin, S. J. Komban, Y. Wang, R. Lashgari, X. Li, M. Jansen, Q. Zaidi, and J.-M. Alonso, "Neuronal nonlinearity explains greater visual spatial resolution for darks than lights," *Proceedings of the National Academy of Sciences*, vol. 111, no. 8, pp. 3170–3175, 2014.

[164] J. Ukita, T. Yoshida, and K. Ohki, "Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network," *Scientific reports*, vol. 9, no. 1, pp. 1–17, 2019.

[165] P. Neri, "Nonlinear characterization of a simple process in human vision," *Journal of Vision*, vol. 9, no. 12, pp. 1–1, 2009.

[166] D. J. Heeger, "Normalization of cell responses in cat striate cortex," *Visual Neuroscience*, vol. 9, no. 2, pp. 181–197, 1992.

[167] N. J. Priebe and D. Ferster, "Mechanisms of neuronal computation in mammalian visual cortex," *Neuron*, vol. 75, no. 2, pp. 194–208, 2012.

[168] M. Carandini and D. J. Heeger, "Normalization as a canonical neural computation," *Nature Reviews Neuroscience*, vol. 13, no. 1, p. 51, 2012.

[169] M. H. Turner and F. Rieke, "Synaptic rectification controls nonlinear spatial integration of natural visual inputs," *Neuron*, vol. 90, no. 6, pp. 1257–1271, 2016.

[170] D. Doshkov and P. Ndjiki-Nya, "Chapter 6 - how to use texture analysis and synthesis methods for video compression," in *Academic Press Library in signal Processing*, ser. Academic Press Library in Signal Processing, S. Theodoridis and R. Chellappa, Eds., vol. 5, Oxford, UK: Elsevier, 2014, pp. 197–225.

[171] P. Ndjiki-Nya, D. Doshkov, H. Kaprykowsky, F. Zhang, D. Bull, and T. Wiegand, "Perception-oriented video coding based on image analysis and completion: A review," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 579–594, 2012.

[172] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Proceedings of the International Conference on Systems, Man, and Cybernetics Conference proceedings*, pp. 14–19, 1990, Los Angeles, CA.

[173] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, 1990.

[174] U. S. Thakur and O. Chubach, "Texture analysis and synthesis using steerable pyramid decomposition for video coding," *Proceedings of International Conference on Systems, Signals and Image Processing*, pp. 204–207, Sep. 2015, London, UK.

[175] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.

[176] S. Bansal, S. Chaudhury, and B. Lall, "Dynamic texture synthesis for video compression," *Proceeding of National Conference on Communications*, pp. 1–5, Feb. 2013, New Delhi, India.

[177] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 25–39, 1983.

[178] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 959–963, 1985.

[179] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Proceedings of the European Conference on Computer Vision*, pp. 404–417, 2006, Graz, Austria.

[180] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[181] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[182] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3828–3836, 2015, Boston, Massachusetts.

[183] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *Proceedings of the European Conference on Computer Vision*, pp. 143–156, 2010, Crete, Greece.

[184] R. Kwitt and P. Meerwald, *Stex: Salzburg texture image database*, http://www.wavelab.at/sources/STex/.

[185] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint*, arXiv:1610.02055, 2016.

[186] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1033–1038, 1999, Kerkyra, Greece.

[187] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 479–488, 2000, New Orleans, LA.

[188] M. Ashikhmin, "Synthesizing natural textures," *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 217–226, 2001, New York, NY.

[189] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 39–55, 1987.

[190] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," *Proceedings of the 22nd annual Conference on Computer Graphics and Interactive Techniques*, pp. 229–238, 1995, Los Angeles, CA.

[191] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, pp. 262–270, 2015.

[192] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," *Proceedings of the European Conference on Computer Vision*, pp. 702–716, 2016, Amsterdam, The Netherlands.

[193] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[194] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Jun. 2016, Las Vegas, NV.

[195] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[196] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[197] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20k dataset," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, p. 4, Jul. 2017, Honolulu, HI.

[198] C. Fu, D. Chen, Z. Liu, E. J. Delp, and F. Zhu, "Texture segmentation based video compression using convolutional neural networks," *Proceedings of Electronic Imaging*, Feb. 2018, Burlingame, CA, USA.

[199] D. Chen, Q. Chen, and F. Zhu, "Pixel-level texture segmentation based AV1 video compression," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1622–1626, May 2019, Brighton, UK.

[200] I.-R. R. BT.500-14, "Methodologies for the subjective assessment of the quality of television images," Geneva, Tech. Rep., 2019.

[201] Q. Chen and F. Zhu, "Long term hand tracking with proposal selection," *IEEE International Conference on Multimedia & Expo Workshops*, pp. 1–6, 2018.

[202] E. Esen, M. A. Arabaci, and M. Soysal, "Fight detection in surveillance videos," pp. 131–135, 2013.

[203] E. Y. Fu, H. V. Leong, G. Ngai, and S. Chan, "Automatic fight detection based on motion analysis," pp. 57–60, 2015.

[204] Y. Chen, L. Zhang, B. Lin, Y. Xu, and X. Ren, "Fighting detection based on optical flow context histogram," *Proceedings of the Second International Conference on Innovations in Bio-inspired Computing and Applications*, pp. 95–98, Dec. 2011, Shenzhen, China.

[205] P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," *Computers and Electronics in Agriculture*, vol. 76, no. 2, pp. 169–174, 2011.

[206] H. Liu, A. R. Reibman, and J. P. Boerman, "Video analytic system for detecting cow structure," *Computers and Electronics in Agriculture*, vol. 178, p. 105 761, 2020.

[207] A. F. Fernandes, J. R. Dorea, and G. J. Rosa, "Image analysis and computer vision applications in animal sciences: An overview," *Frontiers in Veterinary Science*, vol. 7, p. 800, 2020.

[208] S. Ohayon, O. Avni, A. L. Taylor, P. Perona, and S. R. Egnor, "Automated multi-day tracking of marked mice for the analysis of social behaviour," *Journal of neuroscience methods*, vol. 219, no. 1, pp. 10–19, 2013.

[209] M. Haindl and S. Mike, "Texture segmentation benchmark," *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1–4, Dec. 2008, Tampa, FL.

[210] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "YouTube-VOS: A large-scale video object segmentation benchmark," *arXiv preprint*, arXiv:1809.03327, 2018.

[211] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.

[212] D. Chen, "Advancing video compression with error resilience and content analysis," *Doctoral dissertation, Purdue Univeristy*, 2020.

VITA

Qingshuang Chen received her BS in Electrical Engineering from the Purdue University (2014) and began her PhD studies in the Video and Image Processing Laboratory (VIPER) lab at Purdue. She worked under the supervision of Professor Fengqing Maggie Zhu. She was an intern at BlippAR and Google in Mountain View, CA in the summer of 2017 and the summer of 2018 respectively. She also interned at Black Sesame Technologies in the fall of 2020. Her current research interests include learning based image/video compression and video analysis using deep neural networks. She is a student member of the IEEE and the IEEE Signal Processing Society.