Mobile AR/VR with Edge-based Deep Learning

Jiasi Chen
Department of Computer Science & Engineering
University of California, Riverside

CNSM
Oct. 23, 2019
Outline

• What is AR/VR?
• Edge computing can provide...
  1. Real-time object detection for mobile AR
  2. Bandwidth-efficient VR streaming with deep learning
• Future directions
What is AR/VR?
Multimedia is...

Content creation ➔ Compression ➔ Storage ➔ Internet ➔ End users

- Audio
- On-demand video
- Live video
- Virtual and augmented reality
What is AR/VR?

- Virtual reality
- Augmented virtuality
- Augmented reality
- Mixed reality
- Reality
Who’s Using Virtual Reality?

Smartphone-based hardware:

Google Cardboard

Google Daydream

High-end hardware:

Playstation VR

HTC Vive
Why VR now?

Portability

(1) Have to go somewhere
(2) Watch it at home
(3) Carry it with you

Movies:
CAVE (1992)

VR:
Virtuality gaming (1990s)
Oculus Rift (2016)

Similar portability trend for VR, driven by hardware advances from the smartphone revolution.
Who’s Using Augmented Reality?

Smartphone-based:
- Pokemon Go
- Google Translate (text processing)
- Snapchat filters (face detection)

High-end hardware:
- Google Glasses
- Microsoft Hololens
Is it all just fun and games?

• AR/VR has applications in many areas:
  - AR: process input from the real world (related to computer vision, robotics)
  - VR: output the virtual world to your display (related to computer graphics)
How AR/VR Works

VR:
1. Virtual world generation
2. Device tracking
3. Render
4. Display

AR:
1. Device tracking
2. Real object detection
3. Virtual object placement
4. Render
5. Display
What systems functionality is currently available in AR/VR?
Systems Support for VR

**Game engines**
- Unity
- Unreal

1. **Virtual world generation**

**Device SDKs**
- Cardboard
- Oculus Rift
- HTC Vive
- Microsoft MR

2. **Device tracking**

3. **Render**
- Mobile GPU
- Qualcomm VR/AR chips

4. **Display**
Systems Support for AR

1. Device tracking
2. Real object detection
3. Virtual object placement
4. Render
5. Display

Computer vision / machine learning libraries
- Vuforia
- OpenCV
- Tensorflow

- Google ARCore
- Apple ARKit
- Microsoft Hololens
- Magic Leap
- Smartphones
What AR/VR functionality is needed by researchers?
Research Space in AR

1. Device tracking

Typically done using deep learning (research, not industry)
- **Slow**: 600 ms per frame on a smartphone
- **Energy drain**: 1% battery per minute on a smartphone

MARLIN (SenSys’19), Liu et al. (MobiCom’19), DeepDecision (INFOCOM’18), DeepMon (MobiSys’17)

2. Real object detection

Typically done using SLAM (combine camera + IMU sensors)
- **Slow**: 30 ms per frame on a smartphone
- **Energy drain**: > 1.5 W on a smartphone

ShareAR (HotNets’19), MARVEL (SenSys’18), OverLay (MobiSys’15)
Research Space in AR

Example of slow object detection:

Comparison of different apps’ energy drain:

*Take-home message:* Machine learning is useful in AR
- As part of the AR processing pipeline (object detection)
- At the expense of energy
Research Space in VR

On a content/edge server

1a. Virtual world generation

1b. Transmission over the network

Internet

On the mobile device

2. Device tracking

3. Render

4. Display

High bandwidth: Up to 25 Mbps on YouTube at max resolution

Rubiks (MobiSys’18), FLARE (MobiCom’18), Characterization (SIGCOMM workshop’17), FlashBack (MobiSys’16)

Can machine learning help with VR traffic optimization?

Take-home message: Machine learning is useful in VR
• To help with user predictions, traffic management
Outline

• Overview of AR/VR

• Edge computing can provide...
  1. Real-time object detection for mobile AR
  2. Bandwidth-efficient VR streaming with deep learning

• Future directions
How AR Works

1. Device tracking
2. Real object detection
3. Virtual object placement
4. Render
5. Display

- **Object detection** is a computational bottleneck for AR
- Current AR is only able to detect flat planes or specific object instances
- Can we do more powerful processing on a server?
Reducing lag for augmented reality

- Augmented and virtual reality requires a lot of computational power
  - Run expensive computer vision and machine learning algorithms

Run on the device?  Too slow!

Run on the cloud?  Too far → too slow!

Run on the edge?  ✔️

Challenges with current approaches

- **Current approaches** for machine learning on mobile devices
  - Local-only processing
    - Apple Photos, Google Translate
    - GPU speedup
  - Remote-only processing
    - Apple Siri, Amazon Alexa
  - **Local processing**
    - Slow! (~600 ms/frame)
  - **Remote processing**
    - Doesn’t work when network is bad

- **Our observations**
  - Different AR apps have different accuracy and latency requirements
  - Network latency is often higher than CPU/GPU processing time on the edge server
  - Video streams and deep learning models can scale gracefully
Problem Statement

• **Problem:** How should the mobile device be configured to meet the lag requirements of the AR app and the user?

• **Solution:** Periodically profile, optimize, and update the configuration
Online decision framework

Degrees of freedom:
- Video resolution
- Neural net model size
- Offloading decision
- Video characteristics
  - Frame rate
  - Resolution
  - Bit rate
- Deep learning characteristics
  - Model size
  - Model latency / energy
  - Model accuracy

Constraints:
- Current network conditions
- Application requirements

Optimize decision

Metrics:
- Network condition
  - Bandwidth
  - Latency
- App requirements
  - Latency
  - Accuracy
  - Energy
- Time
- Energy consumption
- Detection accuracy
System design

Input live video

User’s battery constraint
Current network conditions
App latency requirement
App accuracy requirement

Front-end device

Online decision framework

Performance characterization

Edge server

Big deep learning

Output display

Tiny deep learning

Big deep learning
AR Object Detection Quality Metrics

• Accuracy
  • Classification and location both important for AR
    • Intersection over union (IoU) metric
  • Ground truth: Big deep learning running on highest resolution

• Timing
  • Latency: time from when we sent the frame to getting the result
  • Frame rate: 1 / time between consecutive frames
1. Offline performance characterization: How do latency and energy change with video resolution?

Energy and latency increase with pixels$^2$ for local processing.
1. Offline performance characterization: How does accuracy change with bit rate and resolution?

- Encoded videos at different bitrates and resolutions

Accuracy increases more with resolution than bitrate, especially for big deep learning.
1. Performance characterization: How does accuracy change with latency?

- Measured accuracy as deep learning processing latency increased

Accuracy decreases as latency increases.
## 2. Online decision framework: Optimization problem

From offline performance characterization:

\[ f + \alpha \left( \sum_{i=0}^{N} a_i(p, r, l_i) \cdot y_i \right) \]

Maximize

<table>
<thead>
<tr>
<th>Frame rate</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ f ]</td>
<td>[ \alpha \left( \sum_{i=0}^{N} a_i(p, r, l_i) \cdot y_i \right) ]</td>
</tr>
</tbody>
</table>

Subject to

- Local processing time
  \[ l_i = \begin{cases} 
  l_i^{CNN}(p) + \frac{r}{f_B} + L & \text{if } i = 0 \\
  l_i^{CNN}(p) & \text{if } i > 0 
\end{cases} \]
- Network transmission time

\[ \sum_{i=0}^{N} l_i^{CNN}(p) y_i \leq 1/f \]

\[ \sum_{i=0}^{N} b_i(p, r, f) \cdot y_i \leq B \]

\[ a_i(p, r, f) \geq A \cdot y_i, \forall i: \]

\[ f \geq F; \]

\[ r \cdot y_0 \leq R \]

\[ \sum_{i=0}^{N} y_i = 1 \]

### Variables

\[ p, r, f \geq 0; y_i \in \{0,1\}; \]

\[ a_i(p, r, l_i) : \text{accuracy function of model } i \]

\[ l_i^{CNN}(p) : \text{latency function of model } i \]

\[ b_i(p, r, f) : \text{battery function of model } i \]

- Finish processing a frame before next frame arrives.
- Don’t use more than \( B \) battery.
- Meet application accuracy requirement.
- Meet application frame rate requirement.
- Don’t use more than \( R \) bandwidth.

\[ p: \text{video resolution} \]

\[ r: \text{video bitrate} \]

\[ f: \text{frame rate} \]

\[ y_i: \text{which deep learning model to run (local, remote)} \]
After:
Key Take-Aways

Real-time video analysis using local deep learning is slow (~600 ms/frame on current smartphones)

Relationship between degrees of freedom and metrics is complex, and requires profiling

Choose the right device configuration (resolution, frame rate, deep learning model) to meet QoE requirements
Outline

• Overview of AR/VR

• Edge computing can provide...
  1. Real-time object detection for mobile AR
  2. Bandwidth-efficient VR streaming using deep learning

• Future directions
How VR Works

On a content/edge server

1a. Virtual world generation

1b. Transmission over the network

Internet

On the mobile device

2. Device tracking

3. Render

4. Display

- **Can we only send** what is needed?
- **How do we know what to send?**
360-degree Video Example

- https://www.youtube.com/watch?v=sT0hVLEe5mU
Only a portion of the scene is viewed
Motivation

• 360° videos are becoming popular
  • Predicted to become a $108B industry by 2021\(^1\)
  • More engaging and interesting for the user

• Off-the-shelf hardware and software for content creators
  • 360° camera hardware
  • Automatic stitching software

• Many companies/websites serving 360° videos

Challenges

• 360° videos take more bandwidth
  • Higher resolution: 360° videos cover all spatial directions
  • Portions out of the field-of-view are wasted

• How can we reduce the bandwidth requirements?
  1. Chop up the scene into tiles
  2. Predict the field-of-view beforehand
  3. Send the appropriate tiles to the client in advance

• How can we predict the future field-of-view of the user?
  • Machine learning / time series analysis
How much bandwidth do 360° videos need?

• Collected dataset of ~4600 YouTube 360° and regular videos
  • Duration
  • Resolution
  • Bit rate
  • Motion vector

• Measured variability of bit rates over time of 360° and regular videos

• Compared the motion vectors of 360° and regular videos

• Calculated effective resolution of 360° videos based on field-of-view

Duration

360° Videos are short:
- new medium
- complex to produce
Resolution

DASH: multiple resolutions of each video stored on server

360° videos have more resolutions

360° videos tend to have higher resolutions

Fraction of videos encoded at the given resolution
Bit rate

• What is the bit rate of the maximum resolution?

High bit rates for 360° video
System Design

Server

User prediction

Streaming optimization

Tile delivery

User’s head movements

Gyroscope, accelerometer

Client

VR Player

Other + current users’ historical data

Prediction of where the user will look

VR video metadata

Which tiles to fetch

-Buffer occupancy
-Bandwidth estimation

Downloaded tiles
User Prediction

• Combined 3 publicly available datasets of users watching 360 videos
• Used LSTM machine learning model for time series prediction
• Data representation + cleaning matters!

Sample dataset [1]:

<table>
<thead>
<tr>
<th>YouTube ID</th>
<th>Number of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diving-2OzlksZBTiA</td>
<td>58</td>
</tr>
<tr>
<td>Paris-sjxIPiAaB4k</td>
<td>58</td>
</tr>
<tr>
<td>Rhino-7IWp875pCxCQ</td>
<td>21</td>
</tr>
<tr>
<td>Rollercoaster-8lsE-P8nGSM</td>
<td>59</td>
</tr>
<tr>
<td>Timelapse-Ctw8R8thm8m8</td>
<td>58</td>
</tr>
<tr>
<td>Venise-s-AJRFQuAtE</td>
<td>58</td>
</tr>
<tr>
<td>Elephant-2bp3ClCAIg</td>
<td>38</td>
</tr>
</tbody>
</table>

Frame: 451
User Prediction Results

• Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
• Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
User Prediction Results

- Average loss: average loss of the prediction across all frames across all users in the test set (in degrees)
- Future value indicates how many frames ahead we are predicting
Does video content matter?

• Why are the losses so much different between two videos?
  • Can the content of the video help us predict more accurately?

• We plot the heat map of the head position of the users for each video
  • Videos where users don’t look around much → lower prediction error
Heat maps

Paris
Heat maps

Rollercoaster
Heat maps

Rhino
Heat maps

Venise
Heat maps

Timelapse
Key Take-Aways

360-degree VR video are large (up to 25 Mbps)

Machine learning or time series prediction can help predict user behavior and avoid wasted bandwidth

Domain representation and data pre-processing matter! ... Is machine learning really the optimal choice?
Future Directions
New application: Multi-User AR

How to create a synchronized world view for multiple users?

1. Device tracking
2. Real object detection
3. Virtual object placement
4. Render
5. Display
What does AR network traffic look like?

- AR traffic mainly involves sending device tracking information

  -> Unpredictable because of user interactions

  -> Large bursts (>20Mb) corresponding to tracking data

What should AR network architectures look like?

• Current AR platforms (Google, Apple, Microsoft) use cloud or P2P network architectures
• Focus is on device tracking computations

→ communication vs computation vs privacy tradeoffs

Can edge computing help device tracking-based AR systems?
What are AR quality-of-experience metrics?

• How to evaluate whether an AR/VR system is performing well?
  • Needed to evaluate the performance of traffic management schemes

• For video, we have MOS, PSNR, SSIM, stalls, bit rate

• What are equivalent quality-of-experience for AR/VR?
  • Motion-to-photons latency
  • Bit rate?
  • Just noticeable difference?
  • Immersion?
  • ...?
Summary

• VR != AR != video streaming

• Machine learning is helpful in certain aspects of AR/VR
  • As part of the AR processing pipeline (object detection)
  • To solve problems in VR (user prediction)

• Edge computing is helpful in certain aspects of AR/VR
  • Reduce the computational load on the AR devices
  • Trade off between computation, communication, privacy

• Many interesting research problems remain
  • Managing multi-user AR traffic
  • Defining user quality-of-experience metrics
  • ...?