

# **Optimizing 360 Video Delivery Over Cellular Networks**

Hélio Assakura - 8941064

Gabriel Baptista (Valdívia) - 8941300

# Artigo

[https://www.cs.indiana.edu/~fengqian/paper/360video\\_allthingscellular16.pdf](https://www.cs.indiana.edu/~fengqian/paper/360video_allthingscellular16.pdf)

# Autores

**Feng Qian**

Indiana University

Bloomington, IN

fengqian@indiana.edu

**Bo Han**

AT&T Labs – Research

Bedminster, NJ

bohan@research.att.com

**Lusheng Ji**

AT&T Labs – Research

Bedminster, NJ

lji@research.att.com

**Vijay Gopalakrishnan**

AT&T Labs – Research

Bedminster, NJ

gvijay@research.att.com

# Resumo

- Vídeo 360
- Display completo x Porção visível
- Estudo de uso de banda de plataformas comerciais
  - YouTube
  - Facebook
- Proposta de um método de stream para ambiente móvel

# Resumo

- Viabilidade do método
  - Coleta de dados visuais
    - Movimento da cabeça pode ser previsto ( > 90 %)
  - Redução do consumo de banda
    - Facilita o acesso pela nuvem
  - Erros de previsão e integração

# Vídeos 360

- Acessibilidade de dispositivos de realidade virtual



<https://vr.google.com/cardboard/get-cardboard/>

\$15.00 - 12/06/2017







<https://www.amazon.com/Oculus-Rift-Virtual-Reality-Headset-Pc/dp/B00VF0IXEY?th=1>

\$499.99 - 12/06/2017

# Vídeos 360

- Acessibilidade de dispositivos de realidade virtual
- Gravando um vídeo 360



Yi Halo



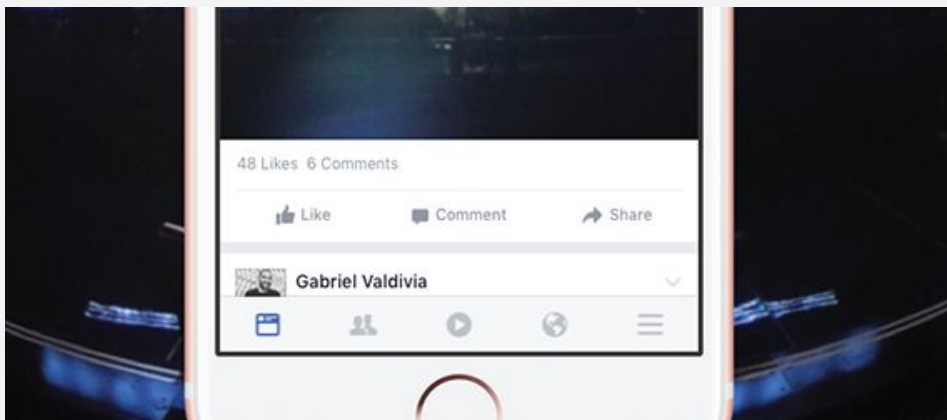
Facebook Surround360

<https://github.com/facebook/Surround360>

# Vídeos 360°

- Acessibilidade de dispositivos de realidade virtual
- Gravando um vídeo 360
  - Ambiente “compactado” em uma esfera 3D



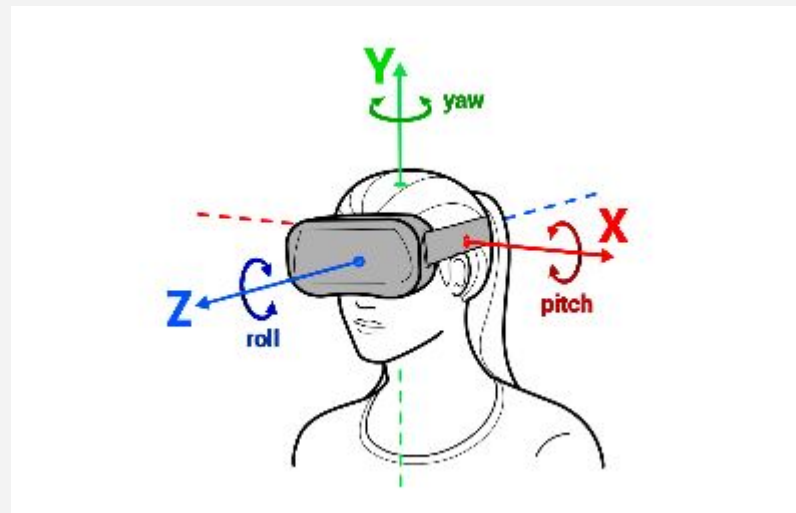


<https://facebook360.fb.com/360-photos/>



# Visão do observador

- Muda nos 3 eixos
- Campo de visão
  - $110^\circ$  horizontal e  $90^\circ$  vertical
  - Depende do aparelho
- Video player calcula e mostra a cena



# **Estudo de plataformas de vídeo comerciais**

# Youtube

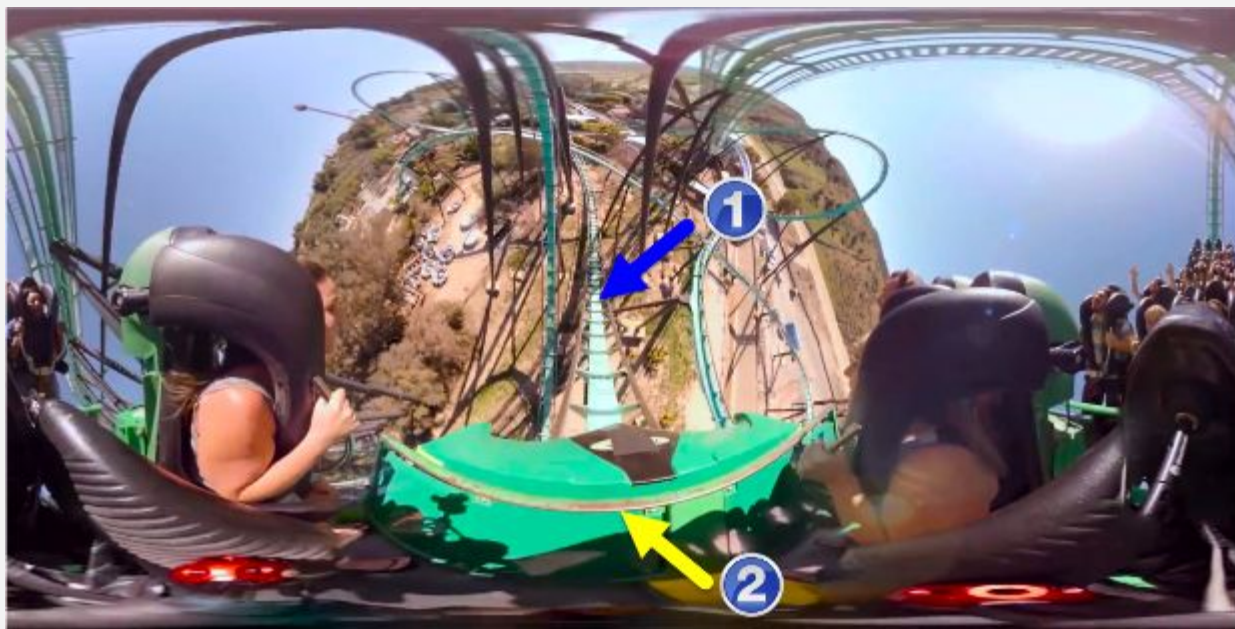
- App oficial
- Especificações
  - Samsung Galaxy S5
  - Android 4.4.2
- Captura do tráfego usando *mitmproxy*
  - <https://mitmproxy.org/>
- Teste com vídeos famosos

# Facebook

- Windows 10 + Google Chrome
  - *mitmproxy* x Facebook app
  - Chrome debug tool

# Resultados

- Codificação H.264 em recipiente MP4
- Reprodução em players convencionais
  - Projeção 3D x Área de visão determinada



# Resultados

- Codificação H.264 em recipiente MP4
- Reprodução em players convencionais
  - Projeção 3D x Área de visão determinada
- Ambos usam download progressivo
- Várias opções de bitrate
  - 144 até 2160 (frame inteiro)
    - Qualidade deve ser melhor (pelo menos 1080p)

# Resultados

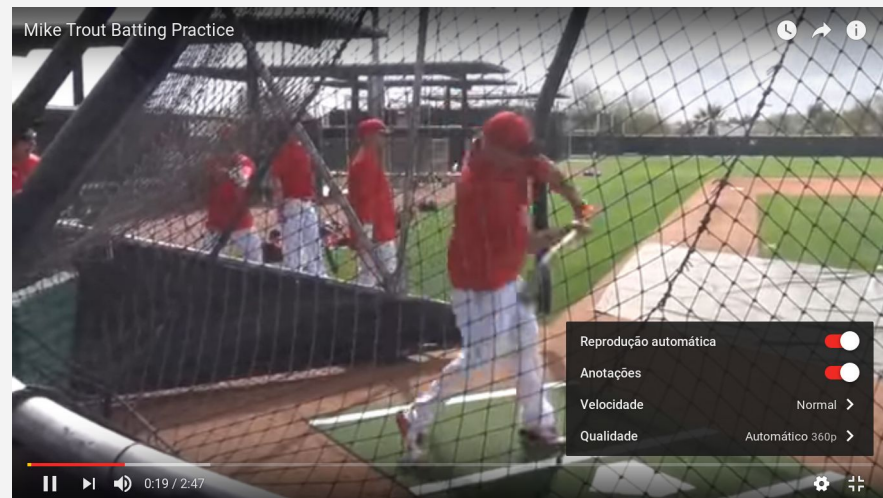
- Tamanho grande
  - Problemas com banda
- Download do frame inteiro
  - 80% do uso da banda desperdiçada
  - H.264 single stream

Video Scene	Length	1080s	1440s	2160s
Roller coaster	1'57''	66MB	105MB	226MB
Animals	2'49''	52MB	129MB	246MB
Aerobatic Flight	8'12''	172MB	250MB	778MB
Google IO 2016	2h8'34''	1.7GB	4.9GB	9.1GB



# Problemas e Desafios

- Grande consumidor de banda
  - Qualidade de percepção da visão, requer maior tamanho
  - Carrega-se a parte visível e a invisível



# Possível solução

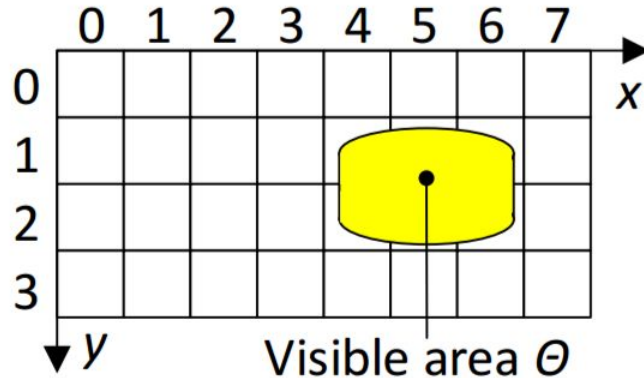
- Carregar somente a área visível
  - Mecanismo download pedaço da área do vídeo
  - Cliente precisa prever o movimento da cabeça
  - Deve incluir pequenas modificações(cliente e servidor)

# Segmentação Espacial

- Precisa ser espacialmente segmentado(online)
  - Cliente marca a área desejada com parâmetros de solicitação HTTP
  - Servidor gera essa área menor *Target Area* e transmite
- Desvantagens
  - Aumento do lado servidor computacional
  - Área marcada não é um retângulo

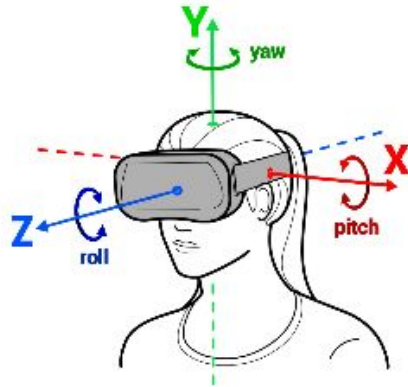
# Segmentação Espacial

- Segmentar de maneira offline
  - Pedacos do vídeo pré-segmentado em pedacos menores  $\longrightarrow$  *tiles*
  - *Tile* tem a mesma duração que um pedaco, mas cobrindo subárea
  - Dividir uniformemente um pedaco em  $m \times n$  *tiles*



# Previsão do Movimento da Cabeça

- Idealmente, se o movimento da cabeça é conhecido, pode-se gerar uma sequência de *tiles* que minimizaria o consumo de banda
- Na realidade, tentar prever movimento nos 3 eixos

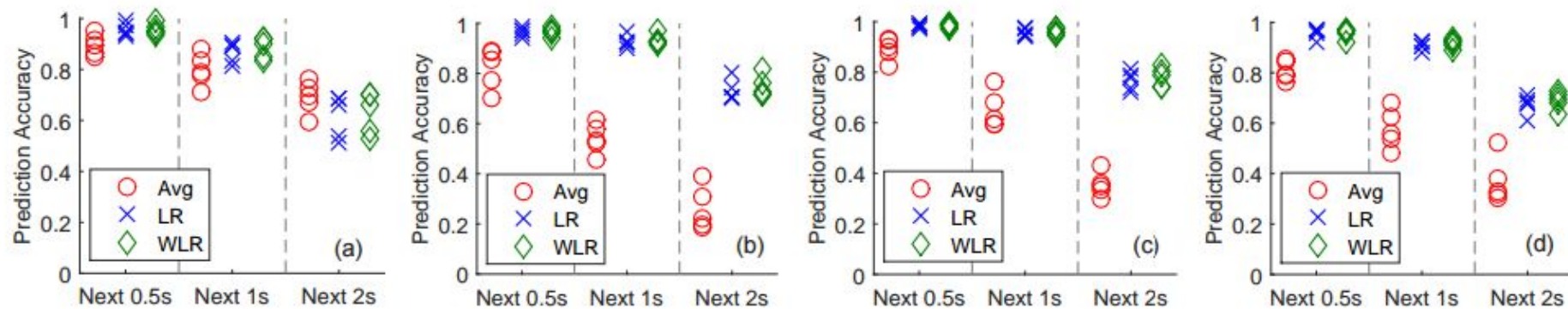


# É possível prever?

- Testes
  - Samsung Galaxy S5
  - 5 usuários
  - Vídeos curtos (1min 40s à 3min 26s) de diferentes gêneros
  - Uso app rastreador de cabeça enviando movimentos nos 3 eixos para um notebook próximo
- Durante os testes os usuários puderam ver em todas direções com o movimento livre de suas cabeças

# Abordagens para os testes


- Average(AVG)
- Linear Regression(LR)
- Weighted Linear Regression(WLR)
  - Peso da amostra em  $t_0$  -  $x$  é setado para  $1 - x$  (  $x \in [0;1]$  )



**Figure 4: Head prediction results for four 360 videos: (a) roller coaster, (b) NASA Mars, (c) sailing, and (d) sports. A moving window of 1s is used to predict next  $\delta=0.5s$ , 1s, and 2s using average (Avg), linear regression (LR), and weighted LR (WLR).**



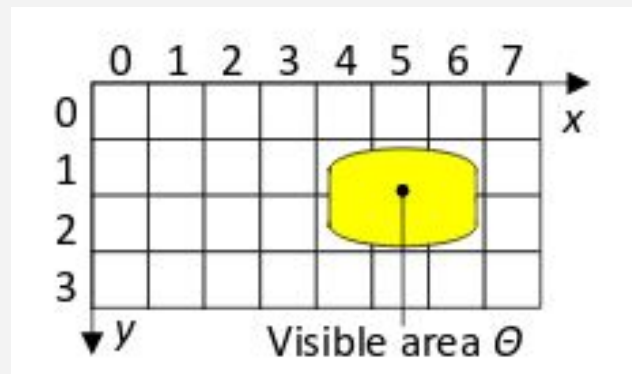
# Analizando Estatísticas Agregadas de Multidões

- Vídeos-360 comerciais bastante visualizações
- Em certas cenas é mais provável que as pessoas olhem para certos locais e/ou direções
  - Exemplo: vídeo de escalada de montanha  topo: olhar em volta
- Utilizar essas estatísticas para prever o comportamento dos usuários
  - Servidor salva frequência dos downloads de cada *tile*
  - Utiliza-se essas frequências para gerar um limiar entre o consumo de banda e a experiência do usuário

# **Simulação**

# Simulação

- Coleta de dados
  - x, y e z da movimentação de 5 pessoas
  - Pedacos de vídeo de 1s
  - FoV de  $110^\circ$  e  $90^\circ$  (H e V)
  - 4x8, 6x12 e 10x20 *tiles*
- Display dos *tiles*
  - Cálculo da área visível  $\Omega$
  - Projeção de  $\Omega$  no visor
  - Escolha dos *tiles* para carregar



# Economia de banda

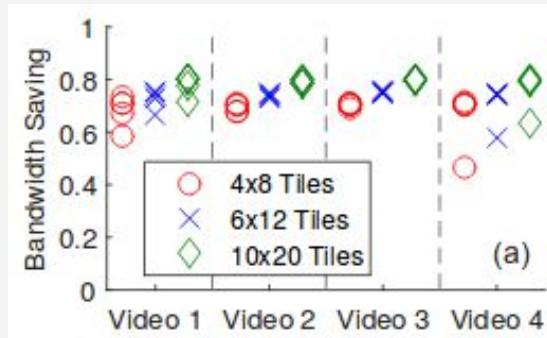
- Estimativa:  $1 - N_F / N$ 
  - $N$  é o número total de *tiles* e  $N_F$  o número de *tiles* carregados
  - Todos os *tiles* tem o mesmo tamanho

# Situações

- A. O player tem conhecimento de todas as leituras de posições
- B. O player carrega *tiles* adicionais, expandindo o campo de visão em  $10^\circ$  nas 4 direções
- c. Mesma configuração de **B** mas com pedaços de vídeo com 4s de duração

# Resultados

- Economia de até 80% em **A**
- Queda baixa de eficiência em **B**
- Quanto mais *tiles*, menos banda consumida
- Não considera erros de previsão



# **Trabalhos relacionados**

- Poucos estudos em vídeos 360
- Temas relacionados
  - Geração de vídeos 360 com apenas 1 câmera
  - Transmissão multiview
- Facebook e transmissões 360
  - Cube Map e pyramid encoding
  - Previsões de movimentos e pyramid encoding
  - <https://github.com/facebook/transform360>
- Vantagens
  - Framework não depende de esquemas de projeção
  - Propõe soluções para erros, levantamentos estatísticos e integração com HTTP e DASH.



<https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/>