SocialSync: Synchronization for a Smartphone Camera Network

Richard Latimer

Rice University

September 7, 2014

Intro

Why Smartphones?

- 1 billion smartphones sold in 2014
- Growth in computer vision applications
- Advanced development platform

Best camera is the one with you!

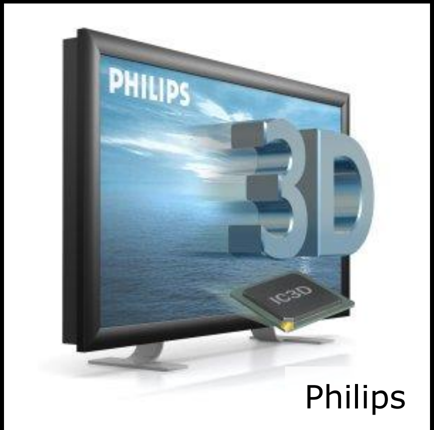
What if Smartphones Could Collaborate?

Lytro

Refocusing

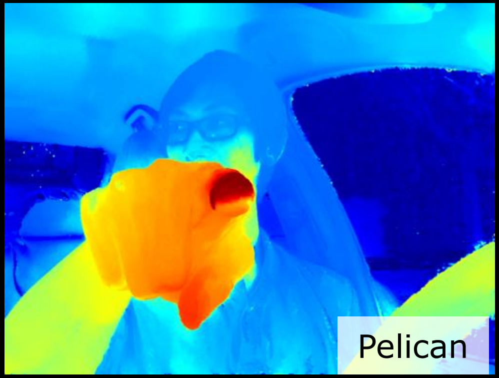
Kilner et al.

Free Viewpoint Television



Philips

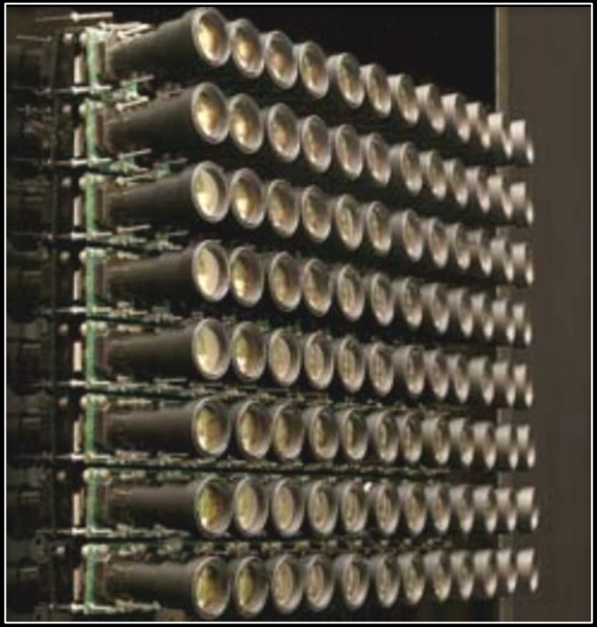
3D



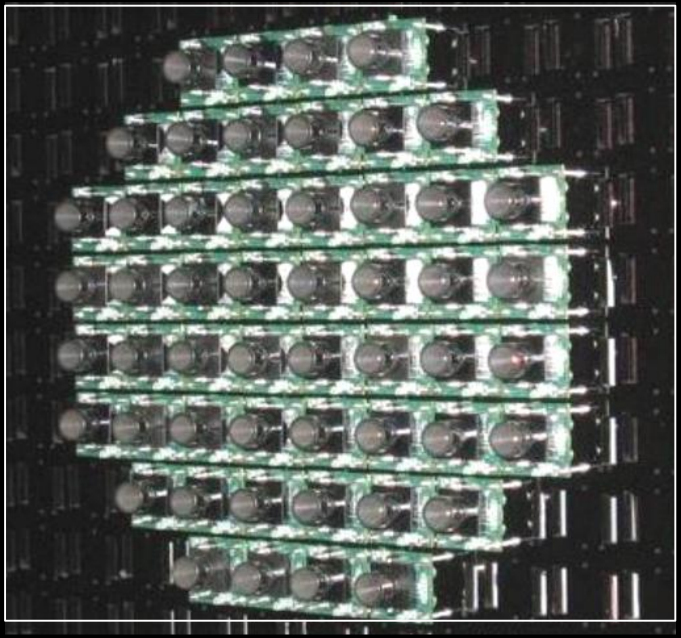
Pelican

Depth

Prior Work: Camera Arrays



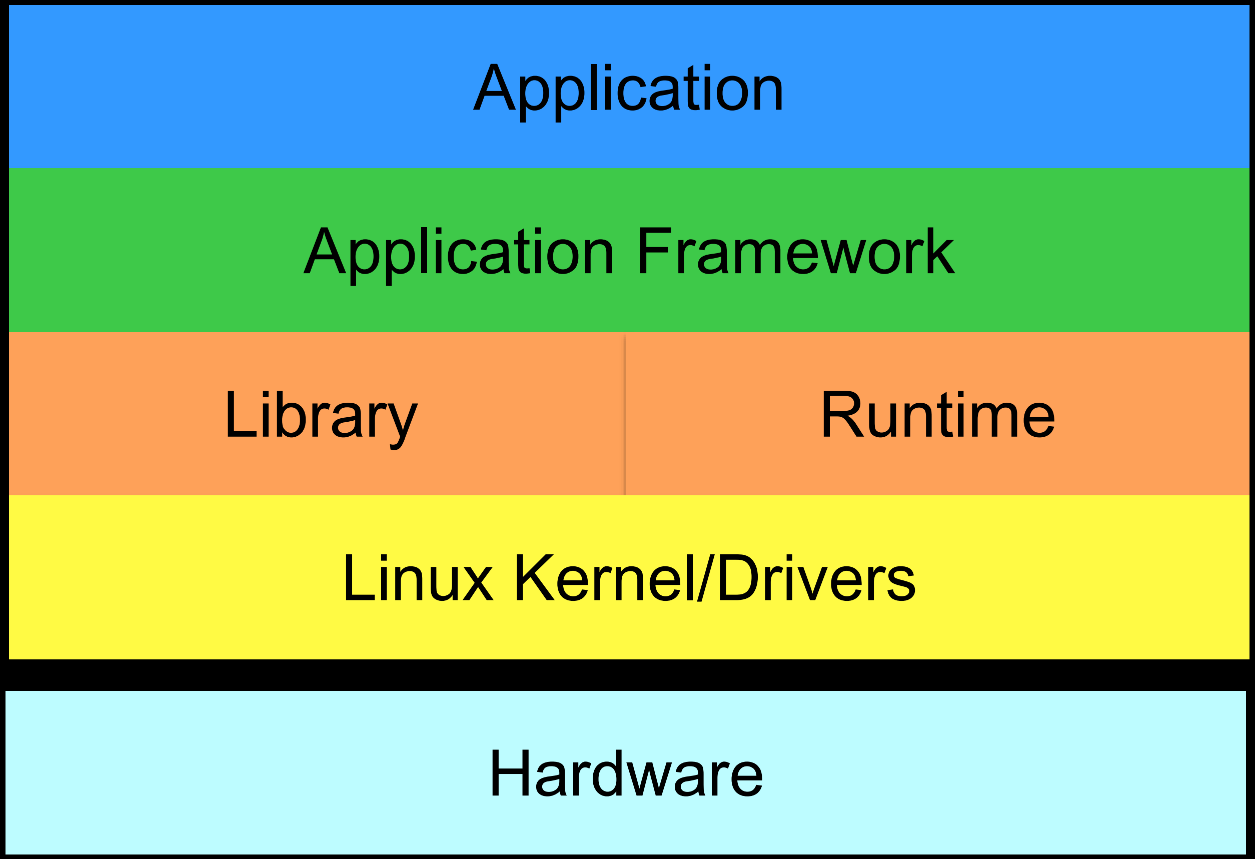
High Performance Imaging Using Large Camera Arrays [Wilburn et al.]



High-Speed Videography Using a Dense Camera Array [Wilburn et al.]

Traditional camera arrays are assumed to be embedded and tightly synchronized.

A Smartphone Camera Array Involves the OS Stack



Application

Application Framework

Library

Runtime

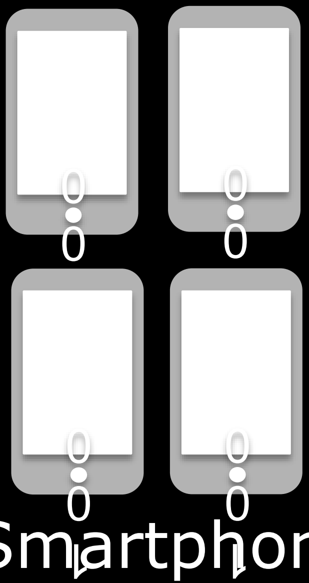
Linux Kernel/Drivers

Hardware

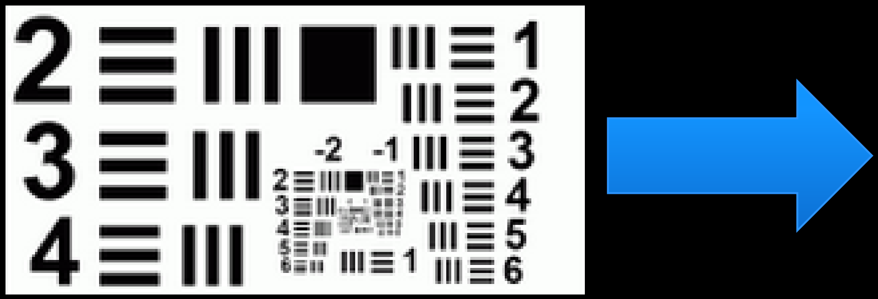
Social Capture: Problem Statement

If you use multiple smartphone cameras to capture video simultaneously, there is an inherit synchronization challenge that manifests (due to OS stack) when the scene is moving.

Scene Setup: How Bad is Misalignment?



Smartphone Camera Array

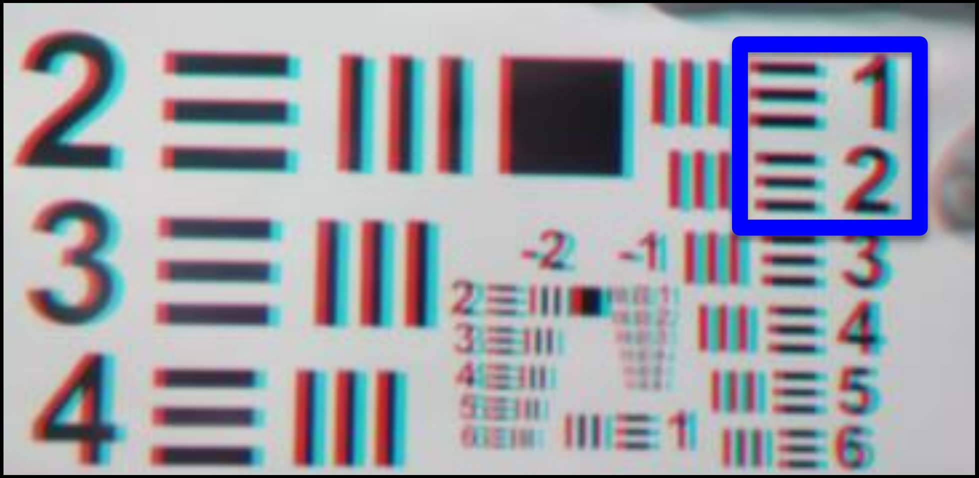


Resolution Target

Scene Setup

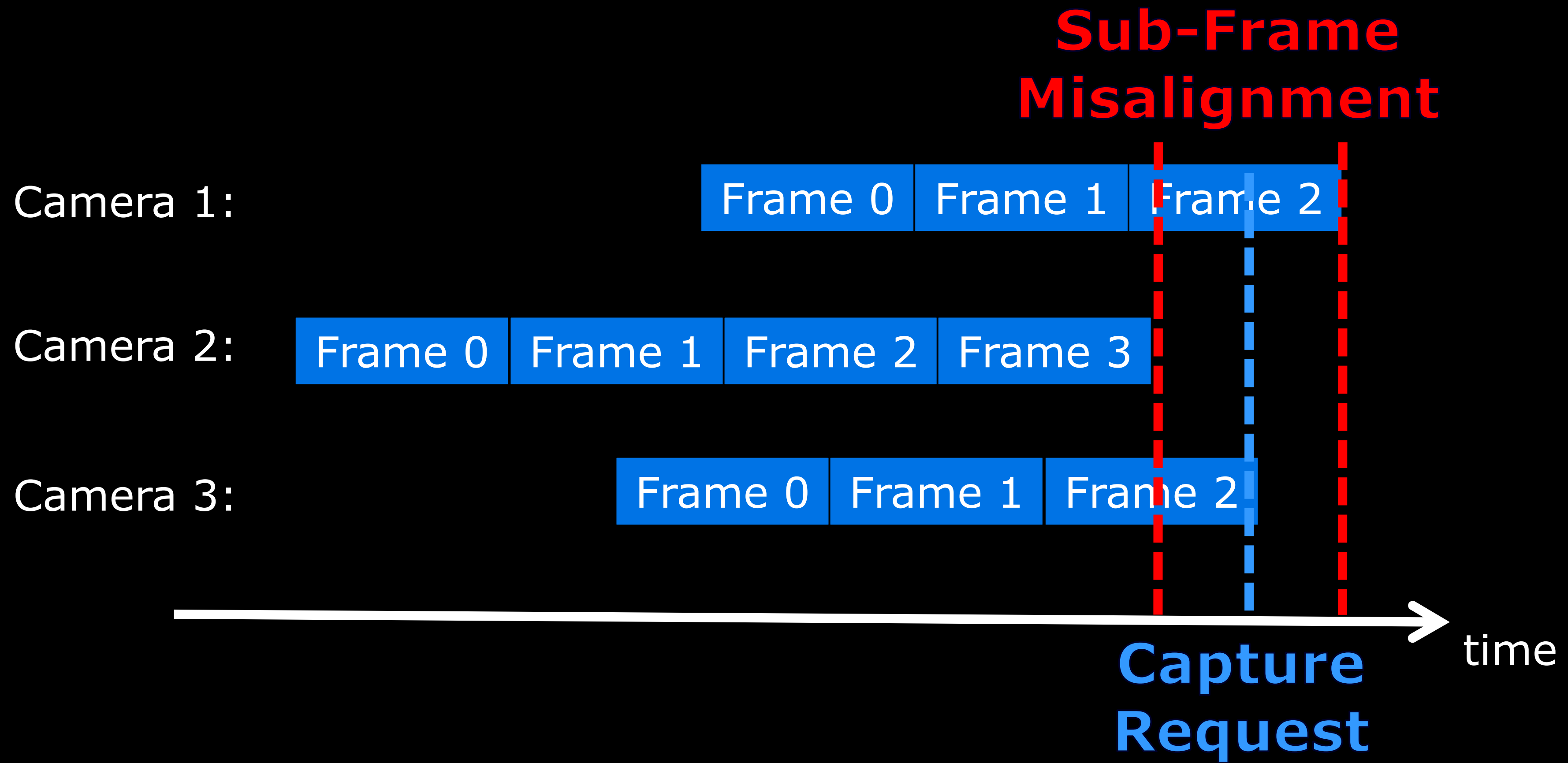
Outcome: How Bad is Misalignment?

Even when photos are captured simultaneously, misalignment of frames can cause problems when target is moving.



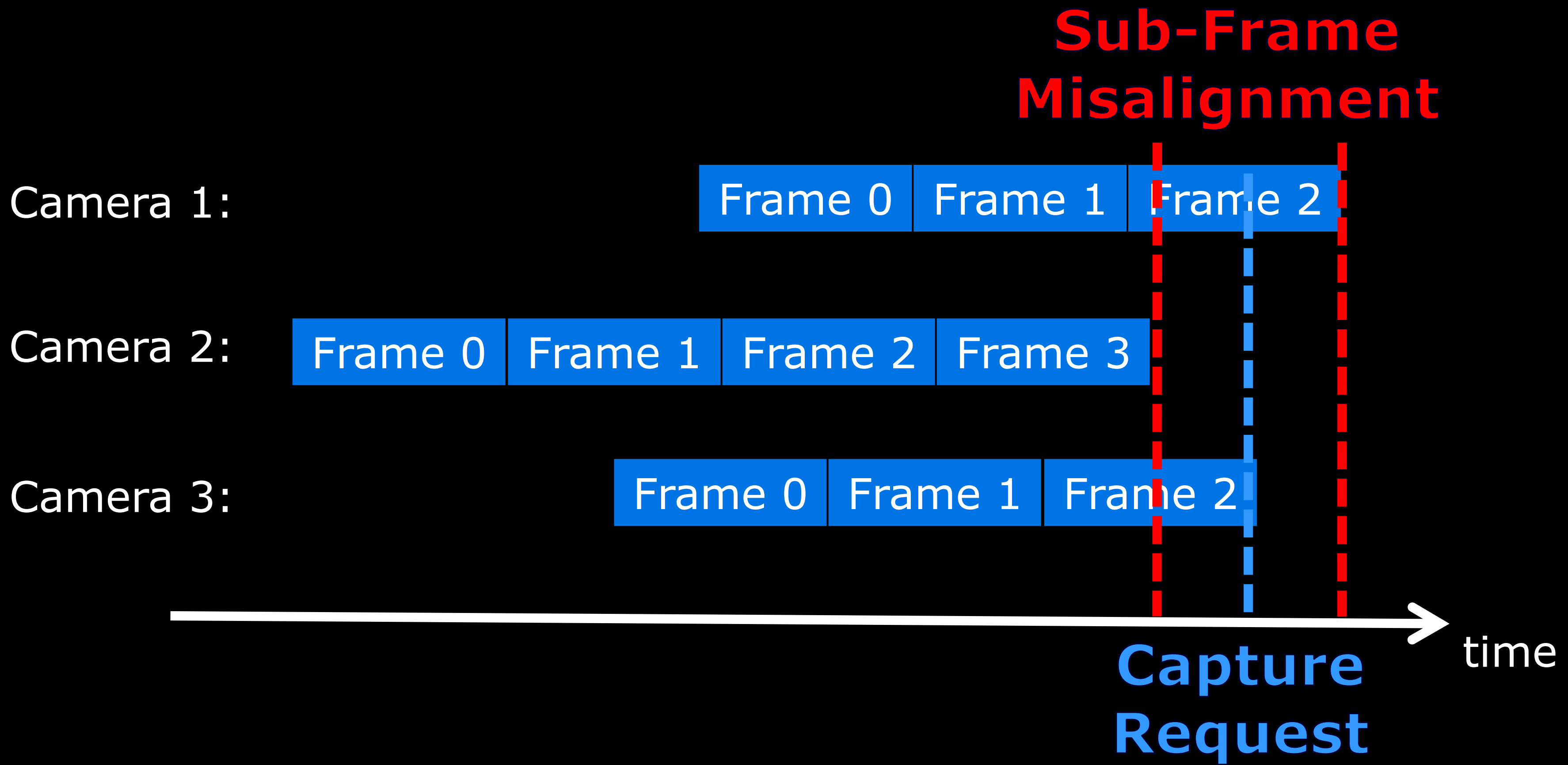
Moving Resolution Target Captured by 4 Phones Simultaneously

Source of Problem



Goal of SocialSync

Objective: Align frame sequences from smartphone cameras within tolerance ranges less than t milliseconds, where t < frame duration.

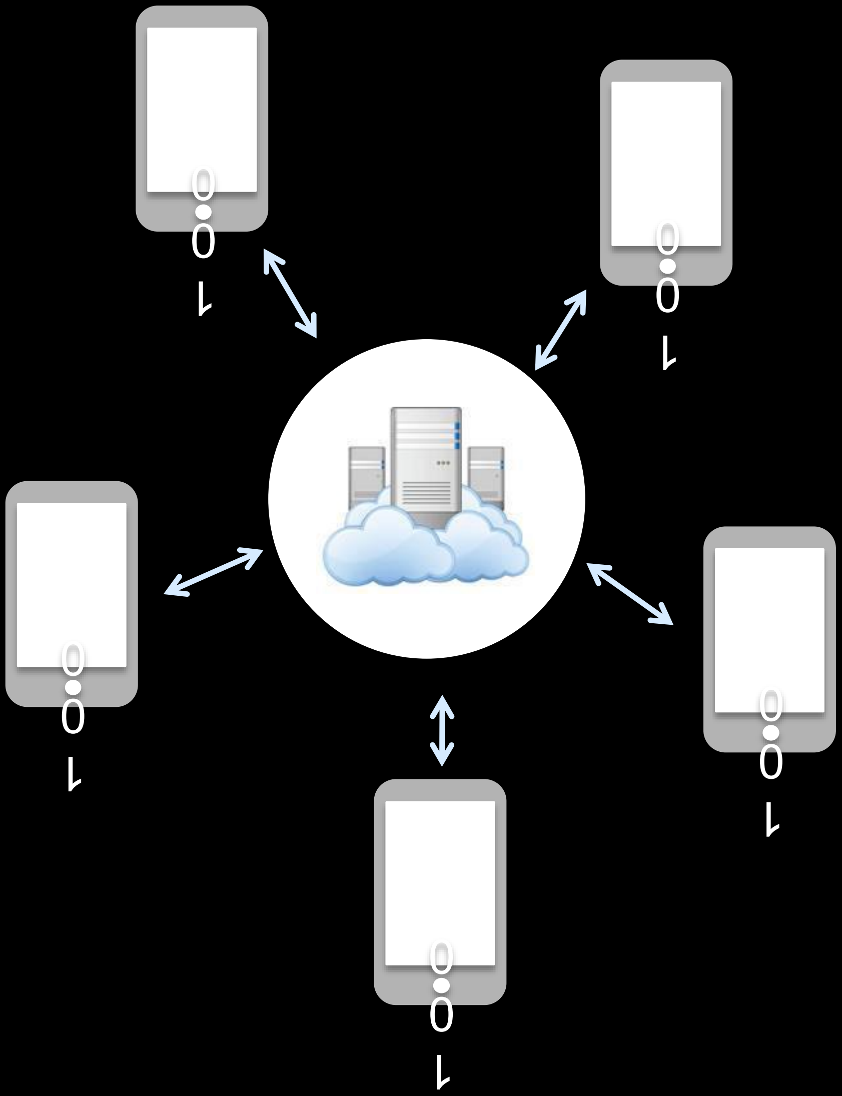


1. Characterization
2. SocialSync Protocol
3. Evaluation

1. Characterization
   - Measurement Test Apparatus
   - Results
2. SocialSync Protocol
3. Evaluation

What We Use: Android 4.4

Network Setup

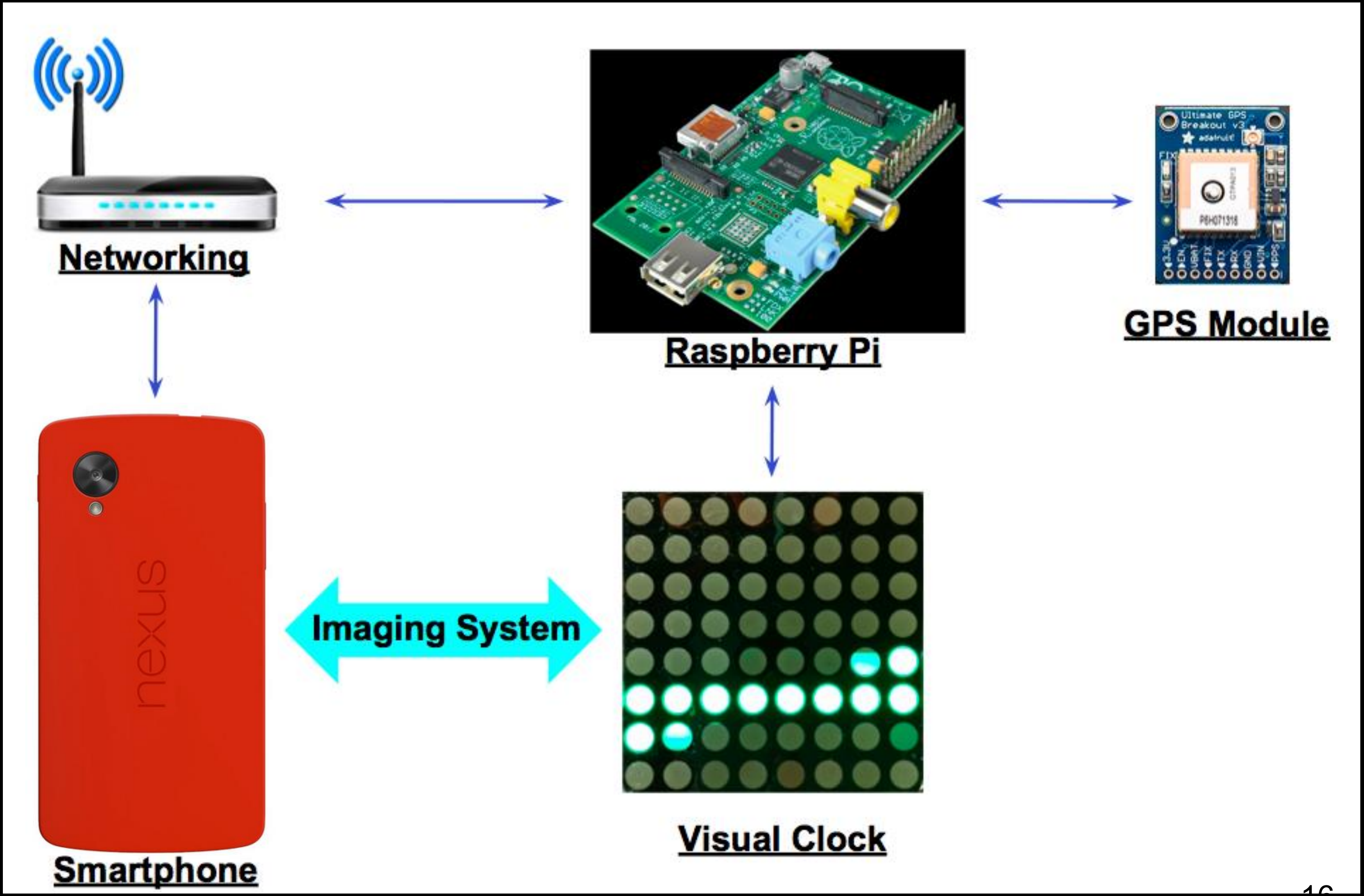


Hub and Spoke Network

•Server provides global reference clock

•Server broadcasts image capture request

System for Measuring Capture Timestamps



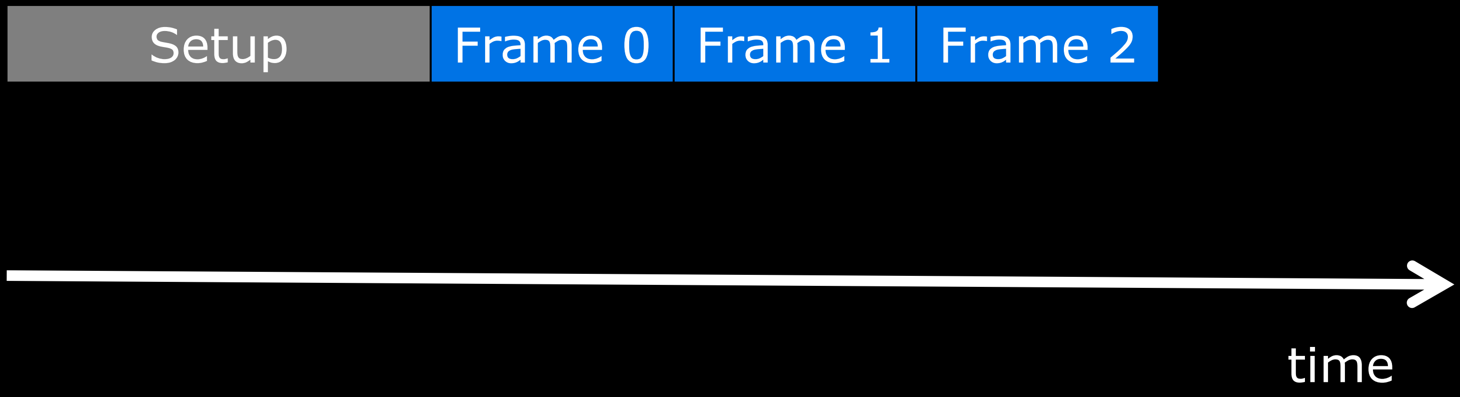
1. Characterization
   - Measurement Test Apparatus
   - Results
2. SocialSync Protocol
3. Evaluation

Random Variables Associated with Capturing a Frame

Setup Time: Delay to setup camera before capture of first frame

Delivery Time: After capture, there is delay to when frame reaches application and a timestamp can be recorded.

Camera Setup



Setup

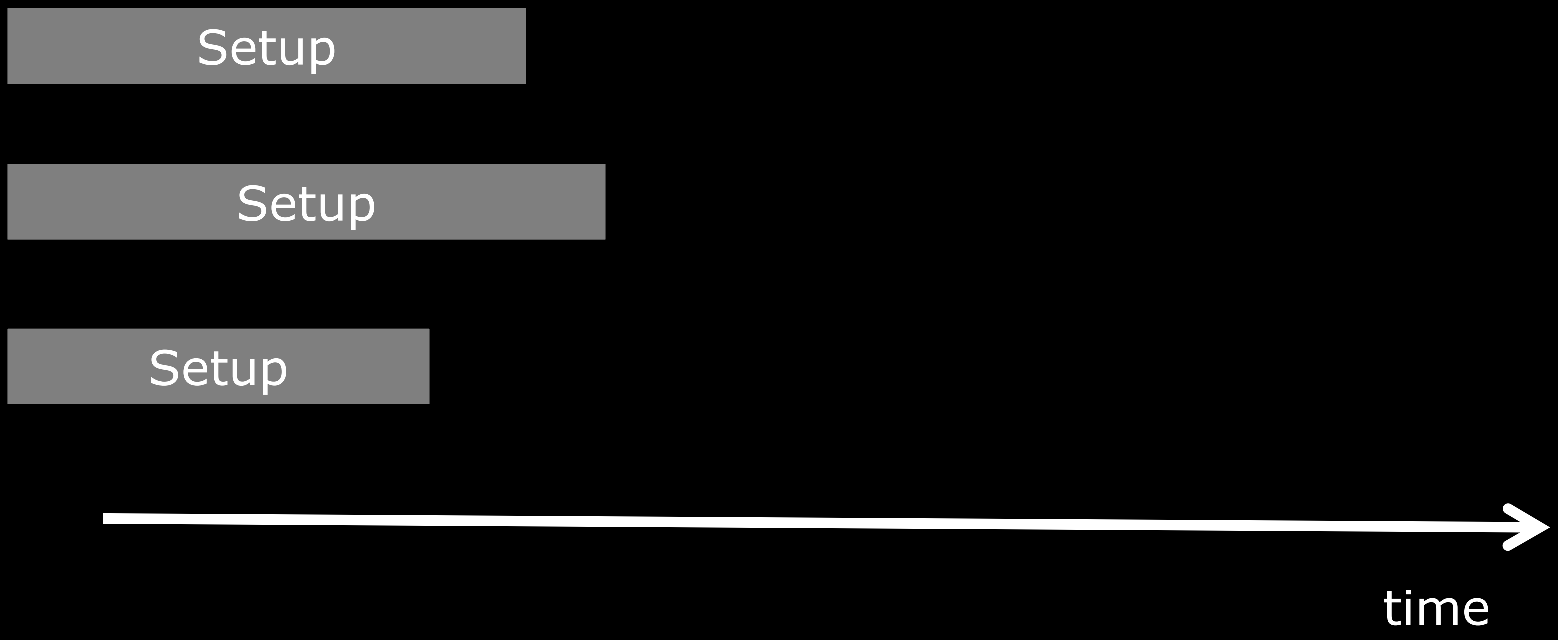
Frame 0

Frame 1

Frame 2

time

Camera 1: Is Setup Time Consistent?



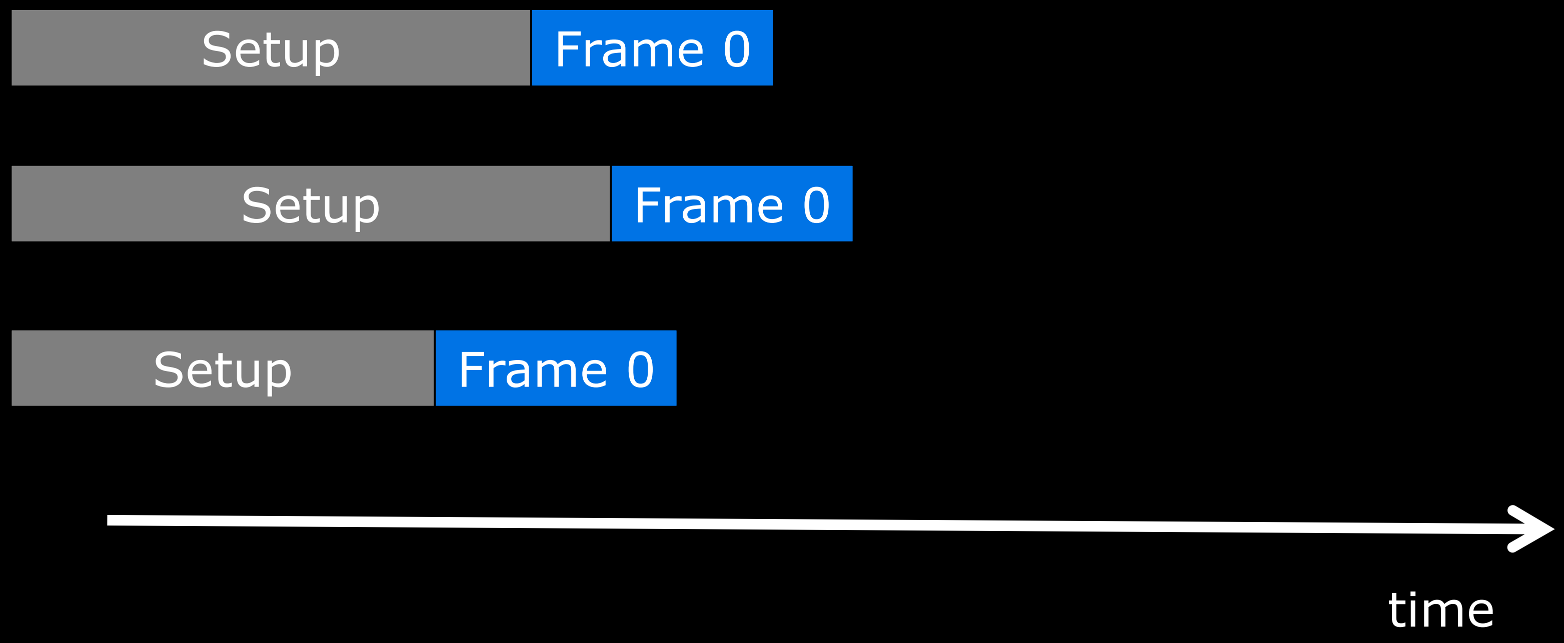
Setup

Setup

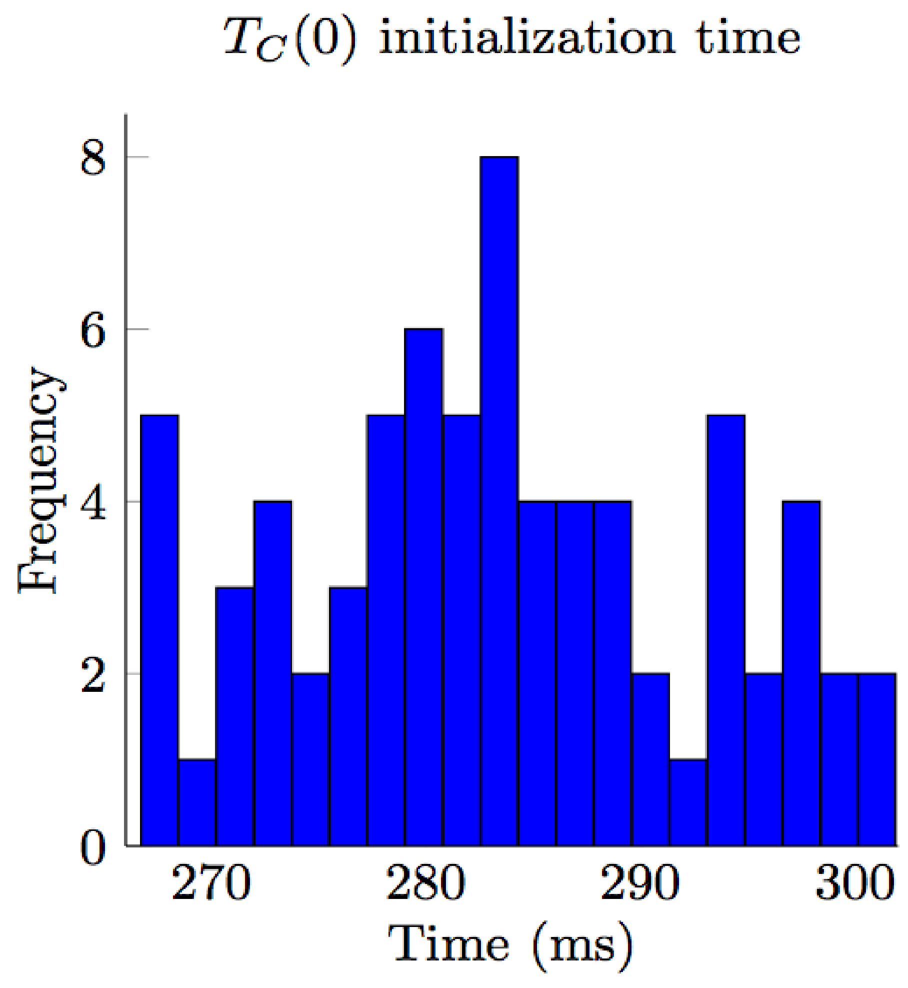
Setup

time

Camera 1: Is Setup Time Consistent?



Variability in Camera Setup Makes Aligning Frames Challenging



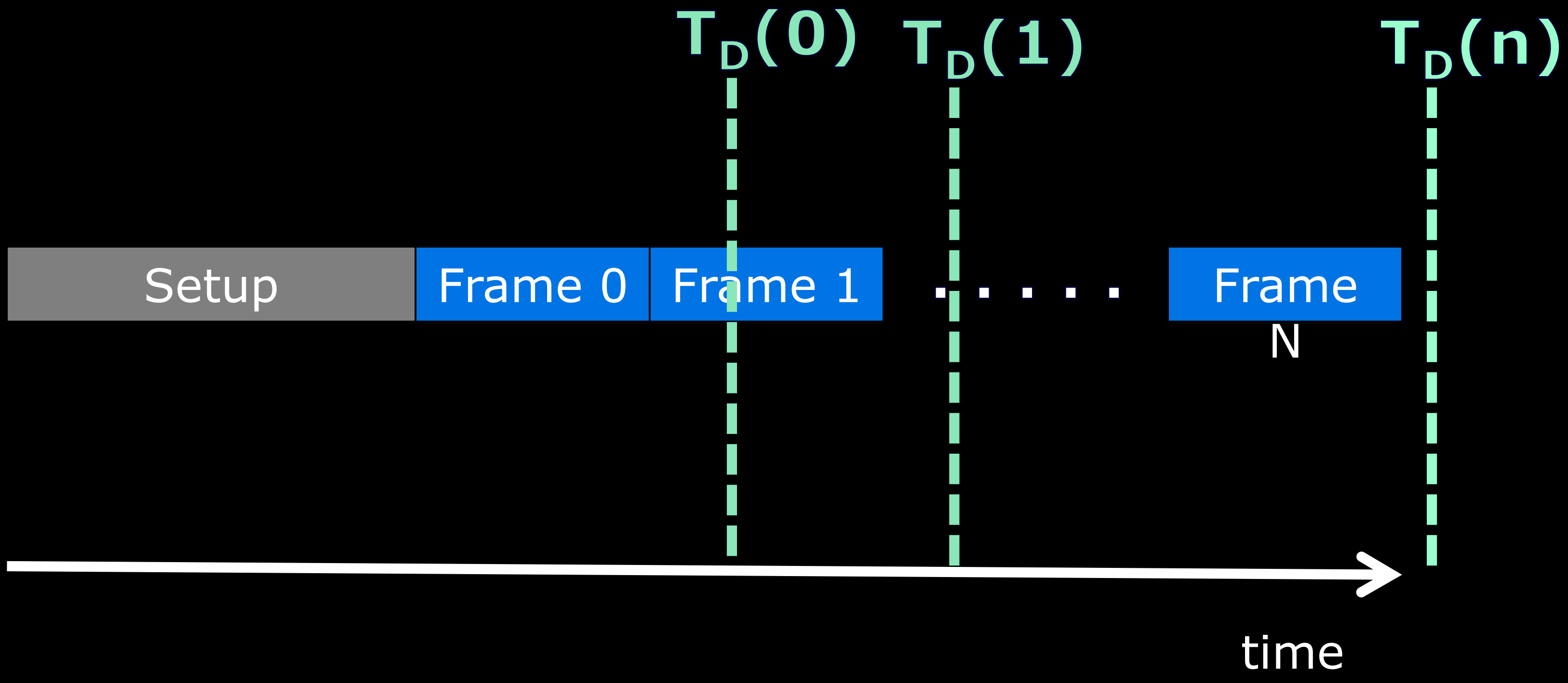
Each time camera started, the start of the image sequence will be delayed by a variable amount of time.

•Setup can vary at least 35 ms

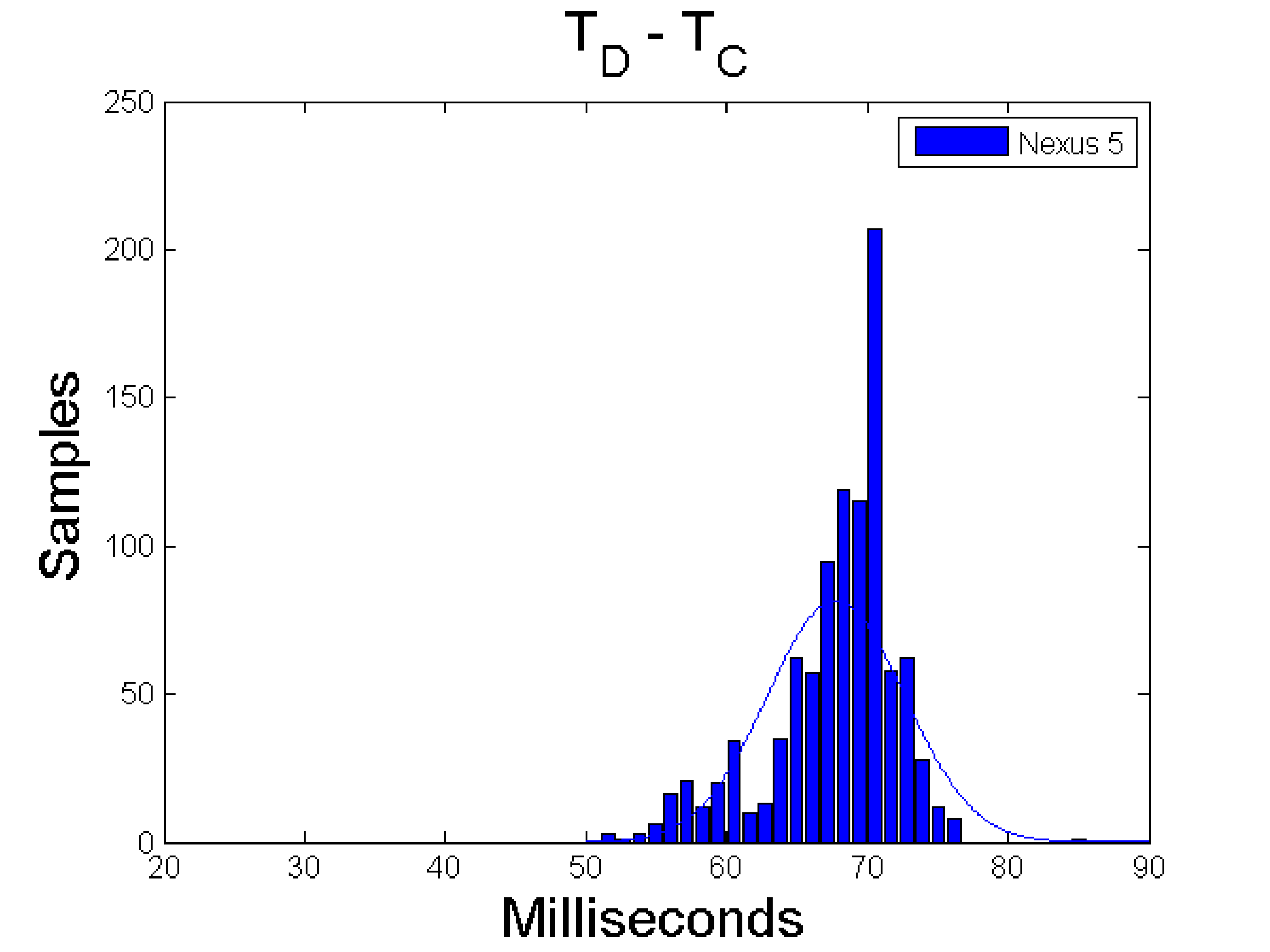
•StDev = 9.4 ms

•Not deterministic

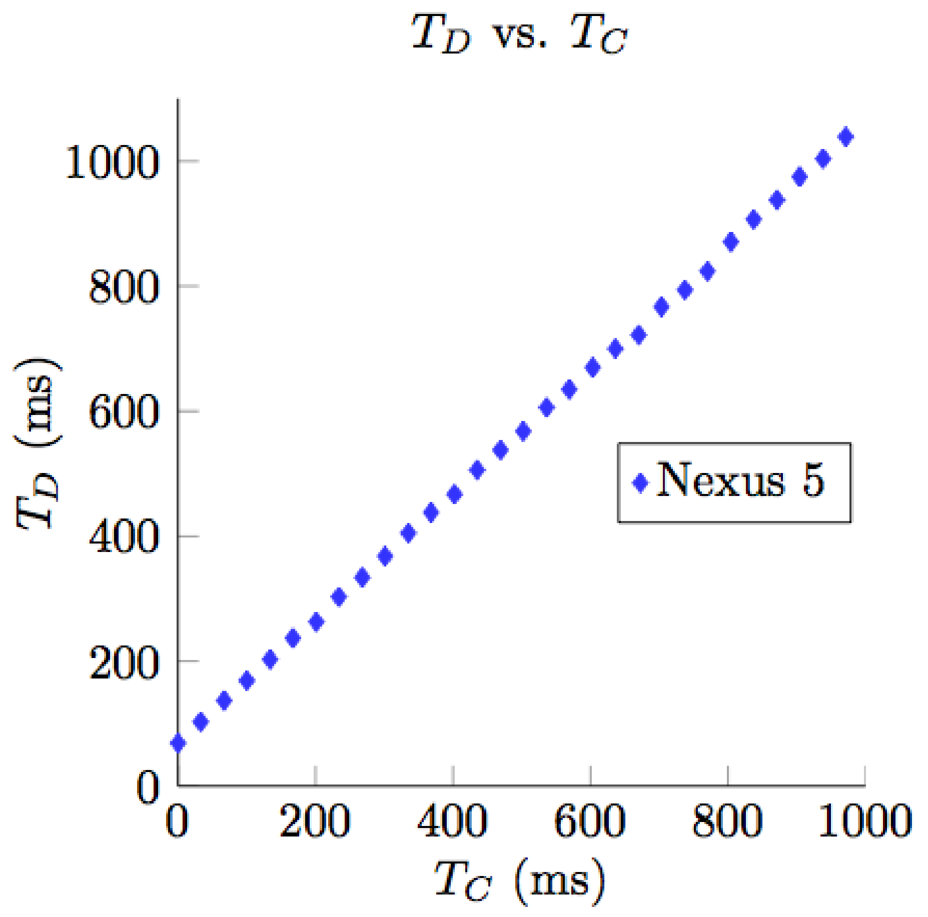
Frame Delay



Measurement of Frame Delay



Delivery Time of Frame to Application



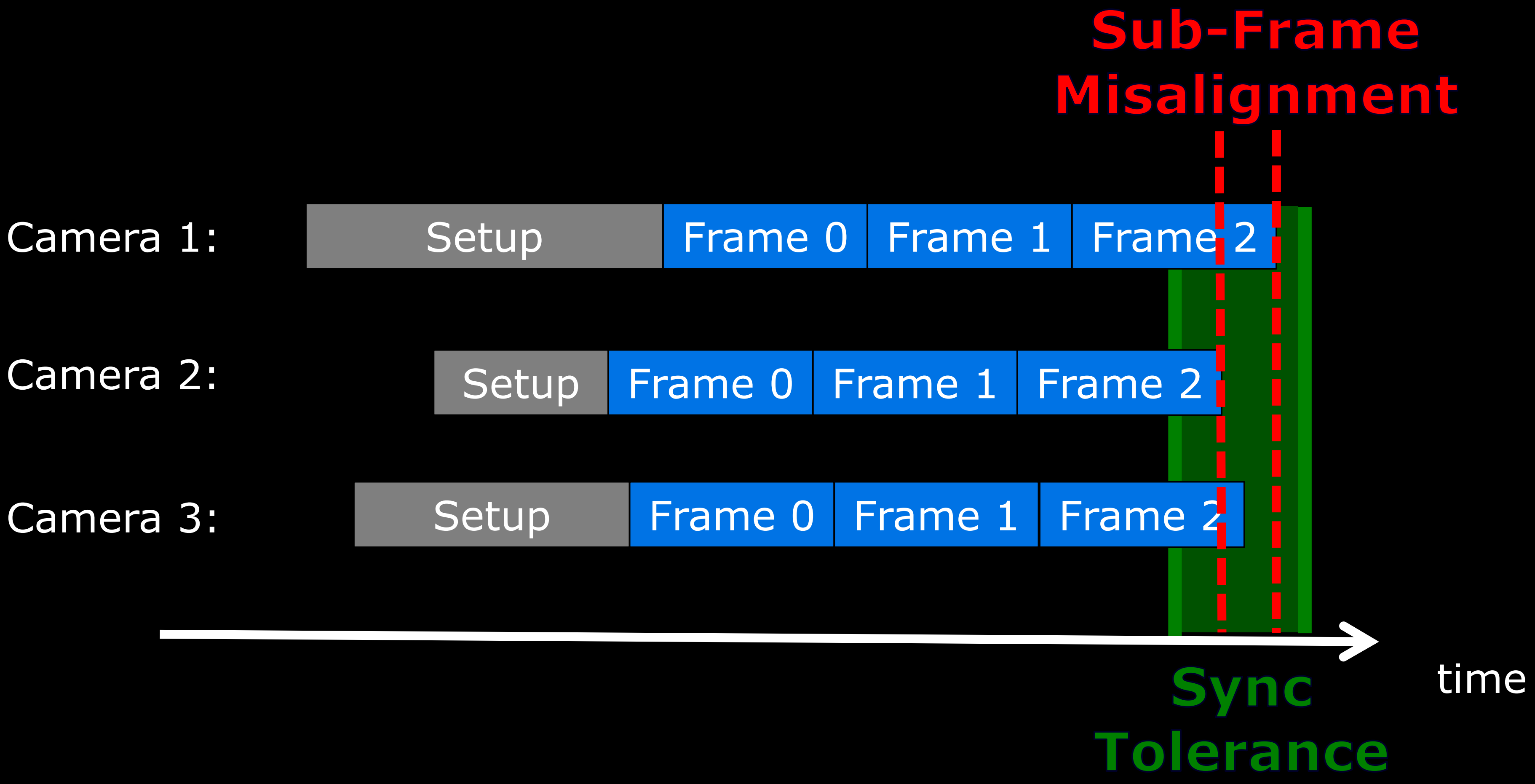
If we keep measuring capture times of frames, they are spaced evenly

Delivery time of frame to app is a proxy for estimating capture time with delay:

- μ = 67 ms
- σ = 4.5 ms

1. Characterization
2. SocialSync Protocol
   - Estimation
   - Alignment
3. Evaluation

Goal of Social Sync

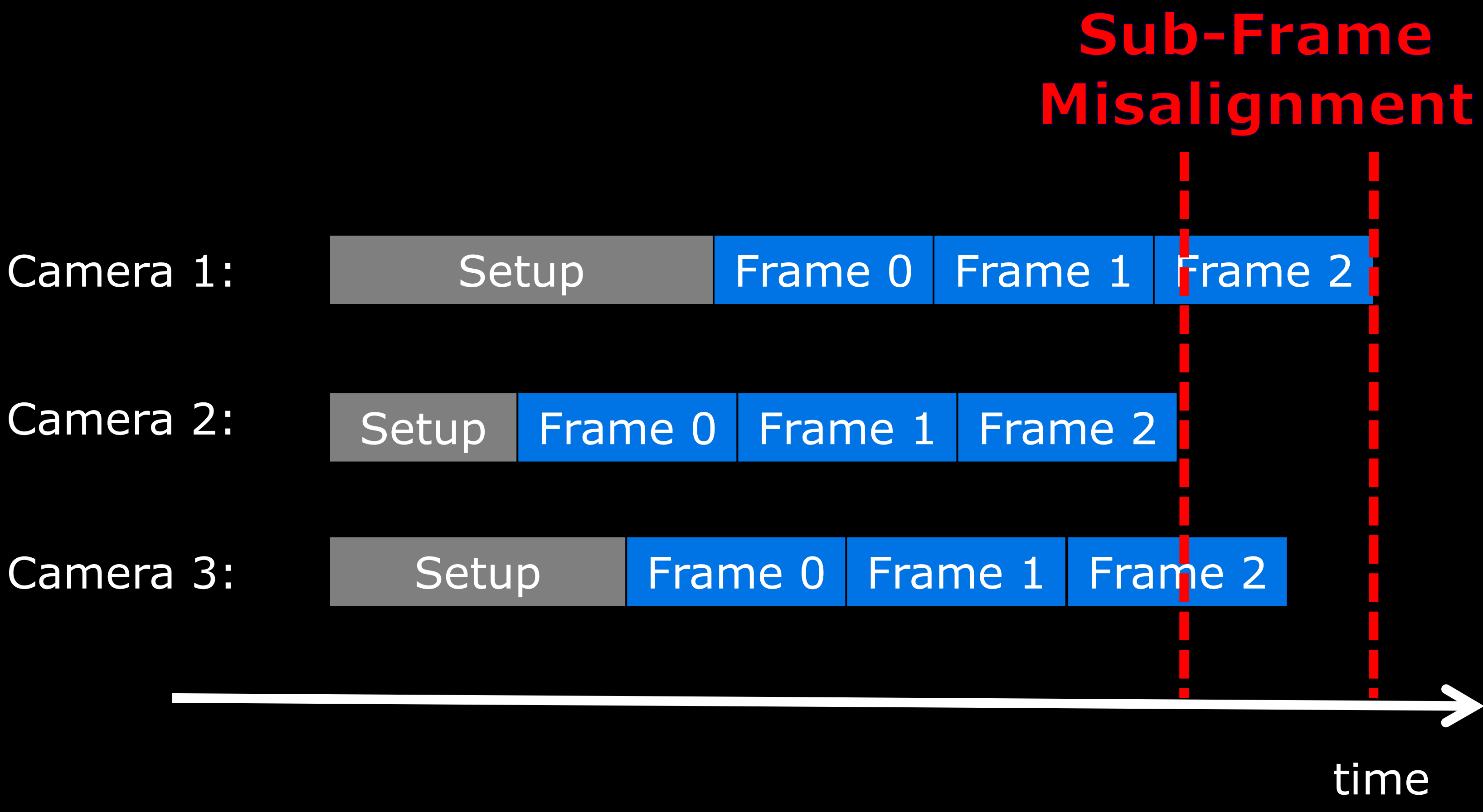


Social Sync Protocol

Estimation Step: Estimating Capture Timestamps to Measure Misalignment of Frames

Alignment Step: Using Repeated Attempts at Launching Image Sequence to Reduce Sub-Frame Misalignment

If Misalignment is Due to Camera Setup



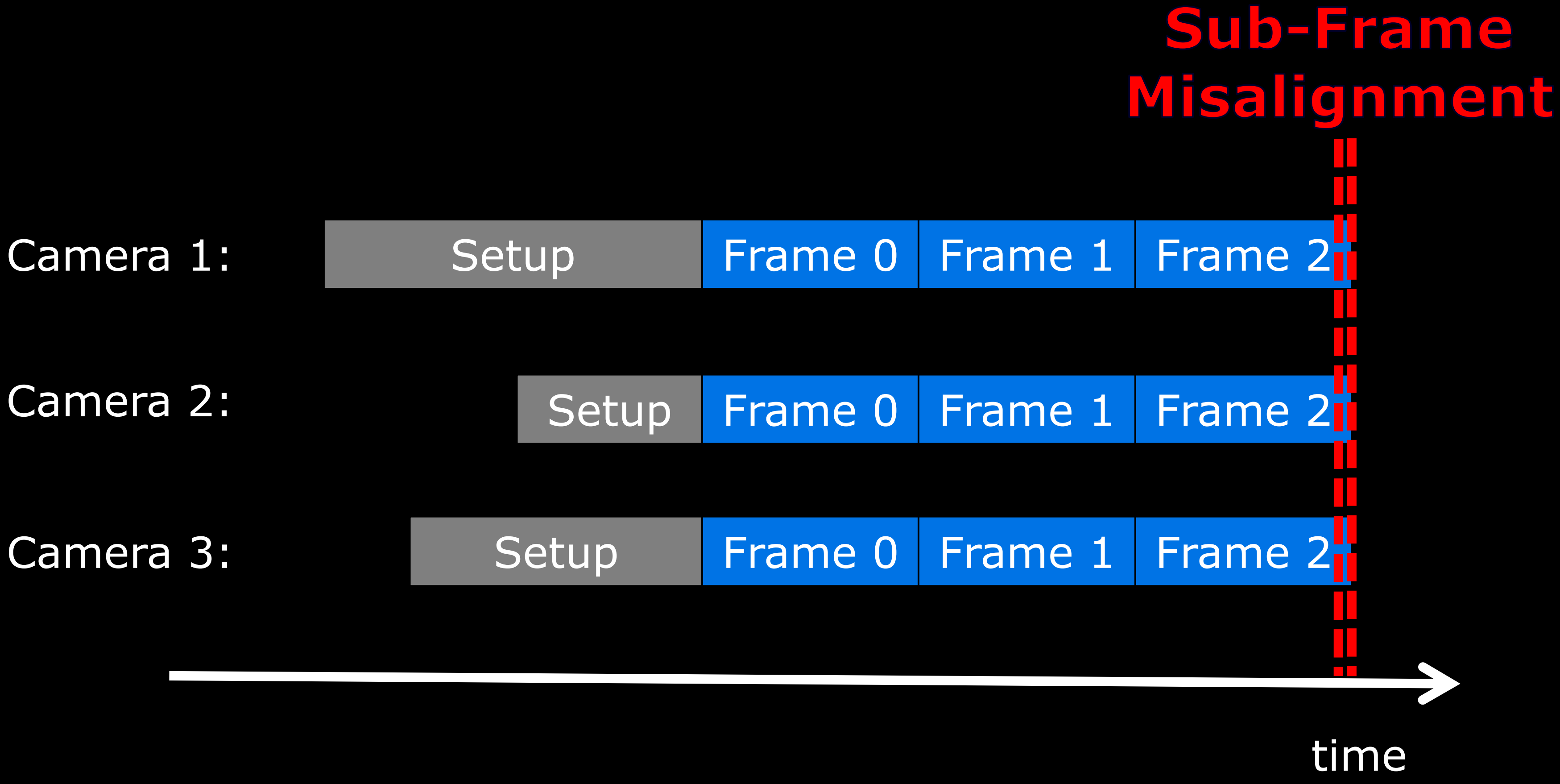
Sub-Frame Misalignment

Camera 1: Setup | Frame 0 | Frame 1 | Frame 2

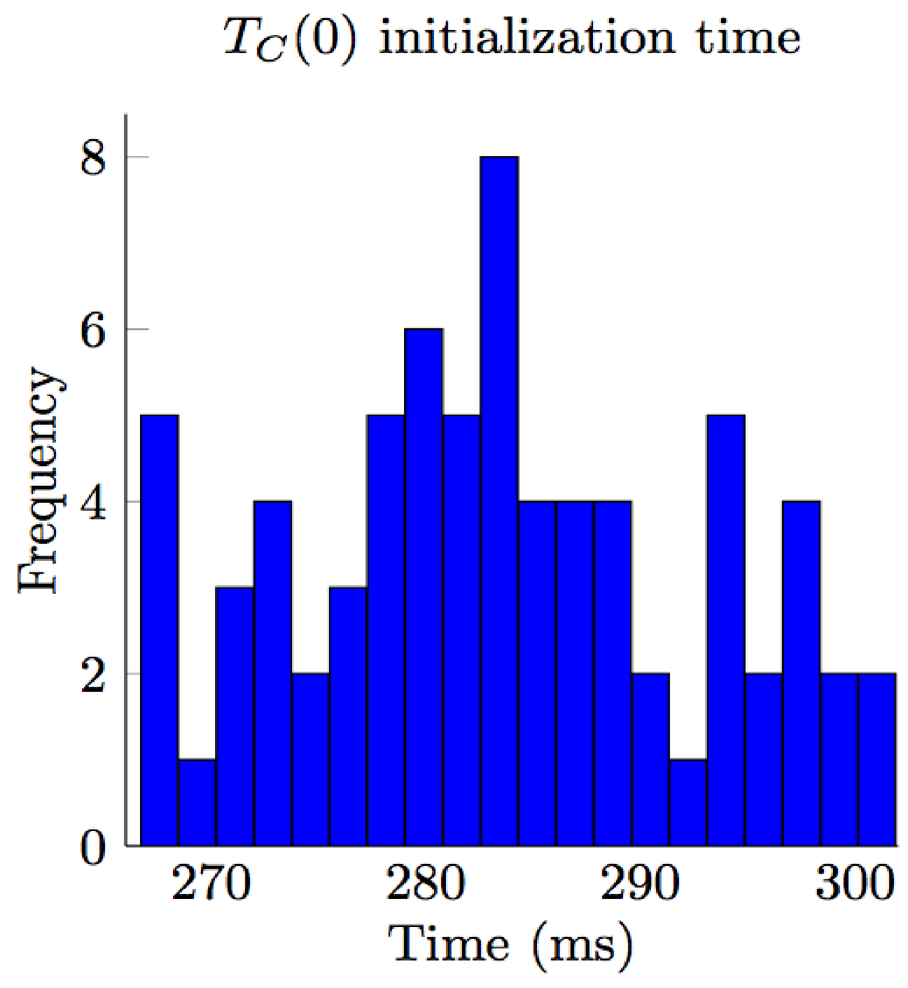
Camera 2: Setup | Frame 0 | Frame 1 | Frame 2

Camera 3: Setup | Frame 0 | Frame 1 | Frame 2

Could We Estimate Capture Time By Calibrating for Setup Time?



Recall Wide Variability in Setup

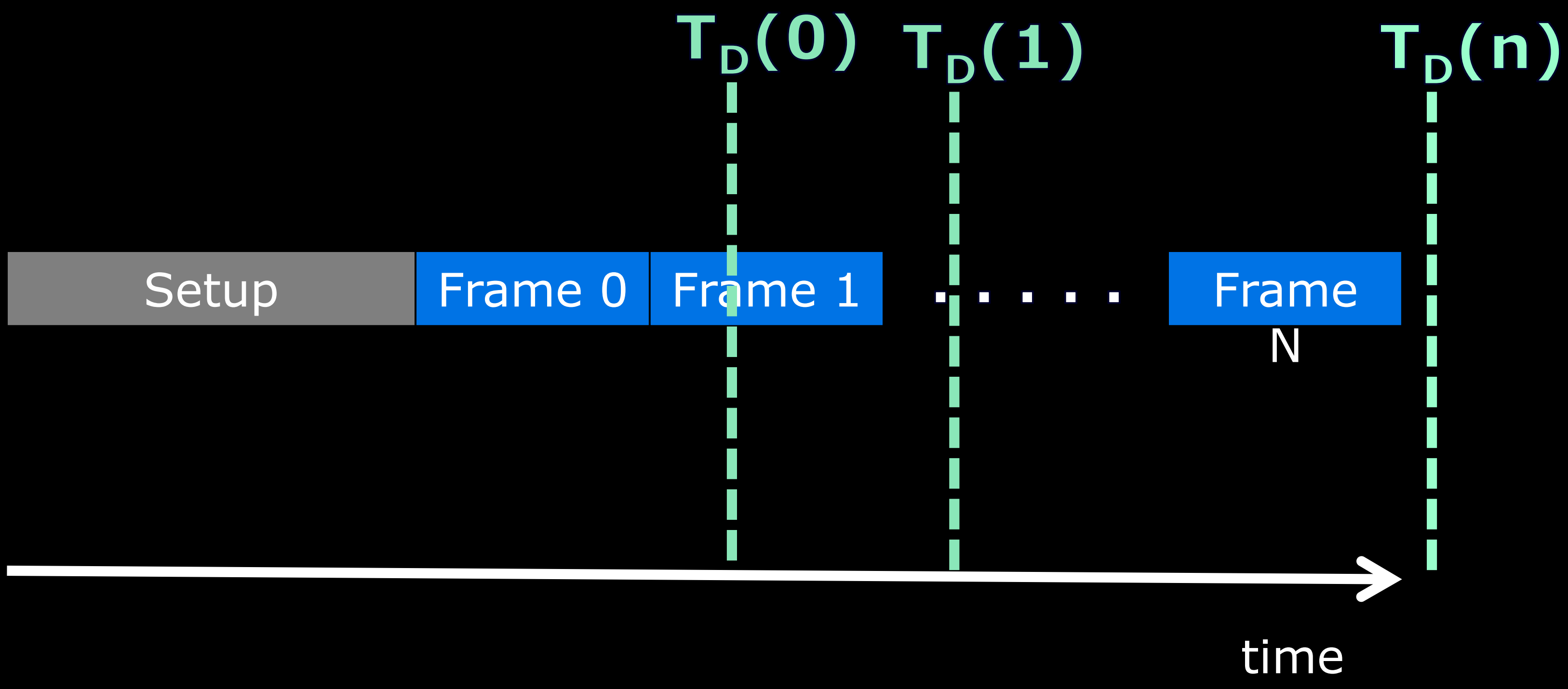


If setup time had little variance, could align capture timestamps by offsetting

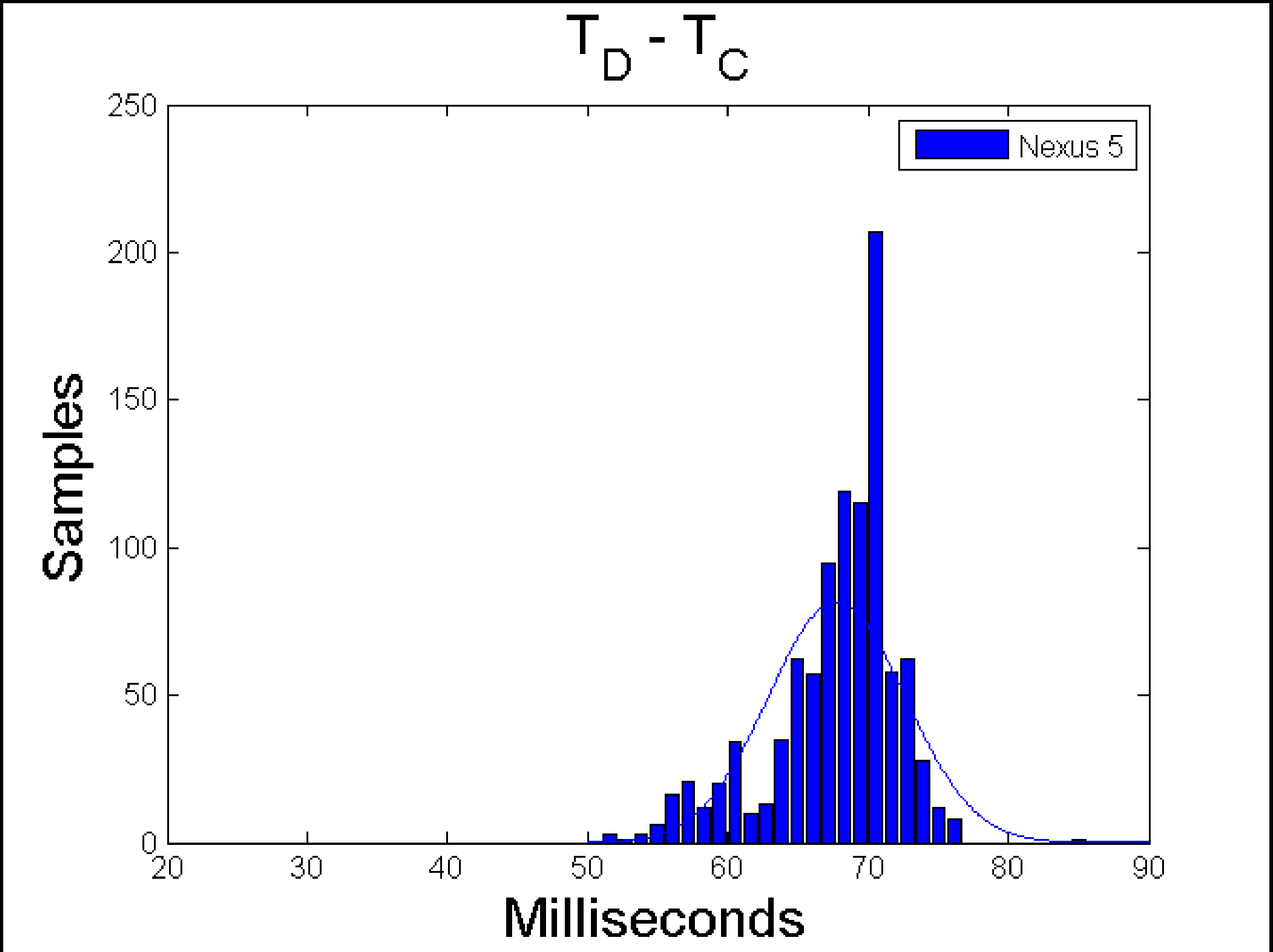
•Setup can vary at least 35 ms, when worst case frame misalignment was 33 ms (inverse of FPS)

•Can’t calibrate for setup time to reduce misalignment!

Does Frame Delay Provide a Better Estimate of the Capture Timestamp?



Measurement of Frame Delay



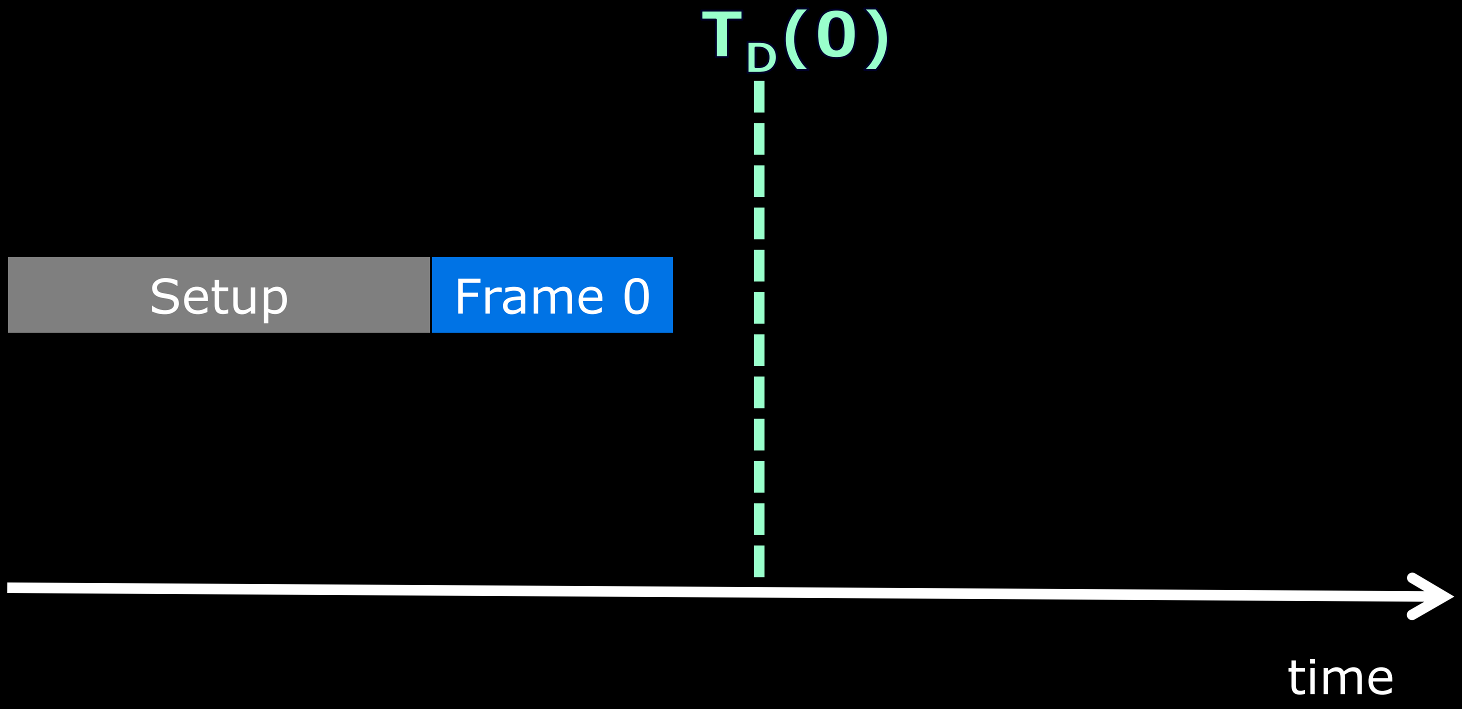
How does estimate of mean improve with number of samples?

Stats 101: Standard error of a sample mean is $\sigma/\sqrt{n}$.

Based on Central Limit Theorem, 95% confidence sample mean is within range:

$$(\bar{x} - 1.96\sigma/\sqrt{n}, \bar{x} + 1.96\sigma/\sqrt{n})$$

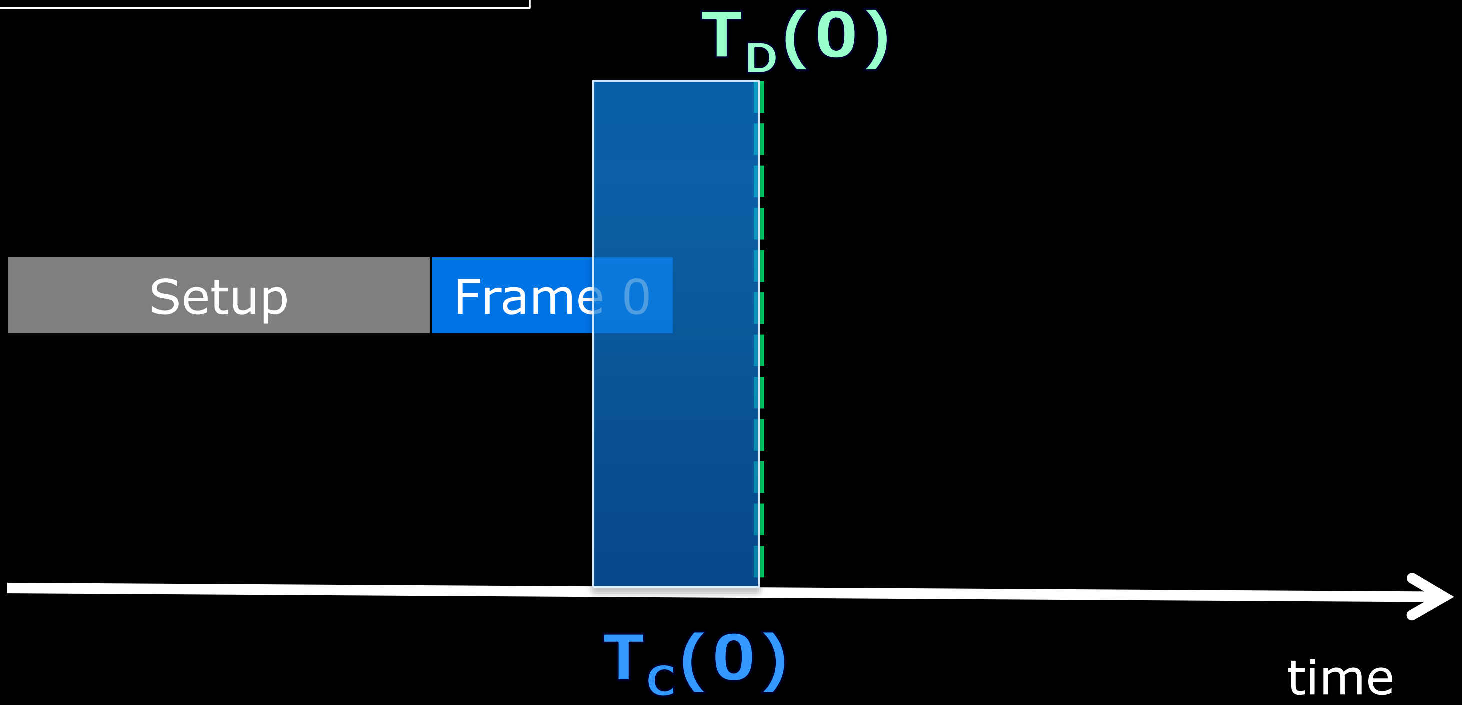
Mean Frame Delay Recap



Mean Frame Delay Recap

Mean Delay Est.:

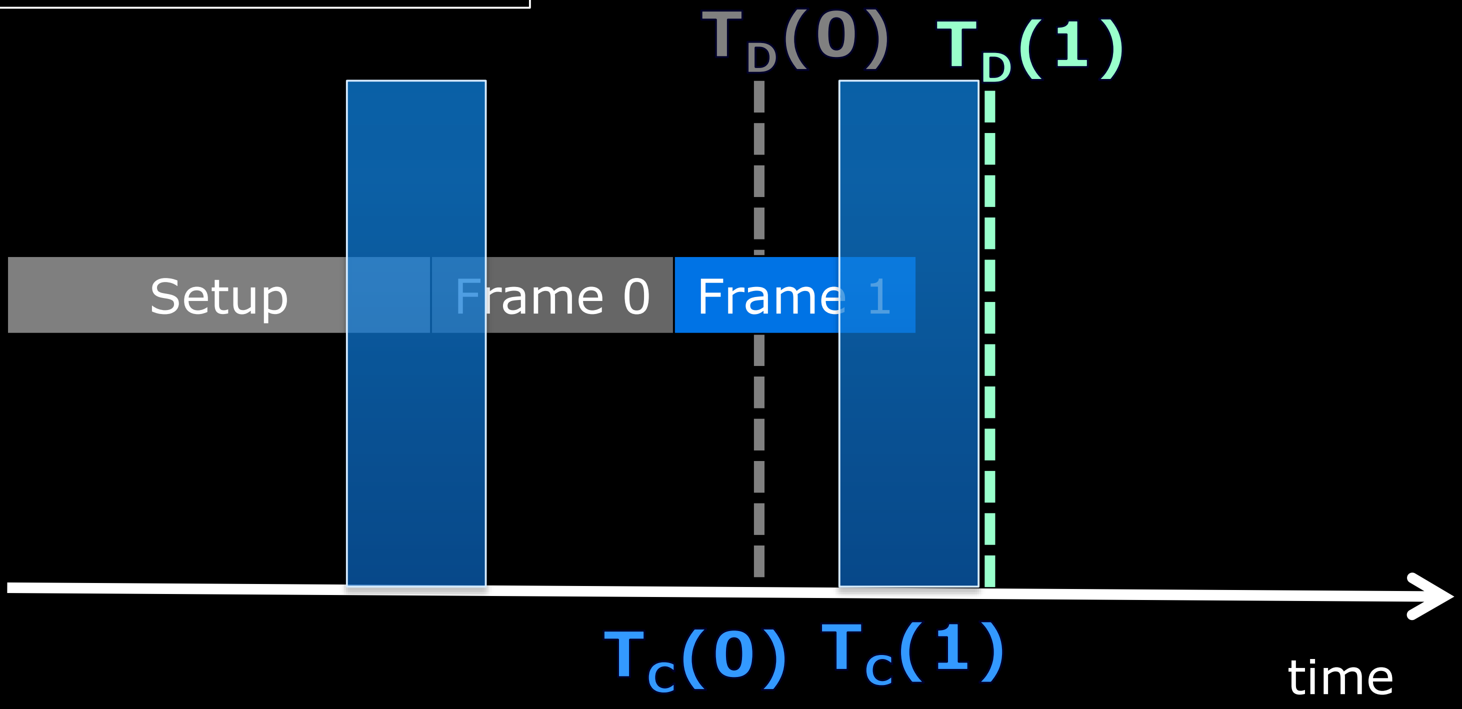
$$\bar{x} \pm 1.96\sigma/\sqrt{n}$$



Mean Frame Delay Recap

Mean Delay Est.:

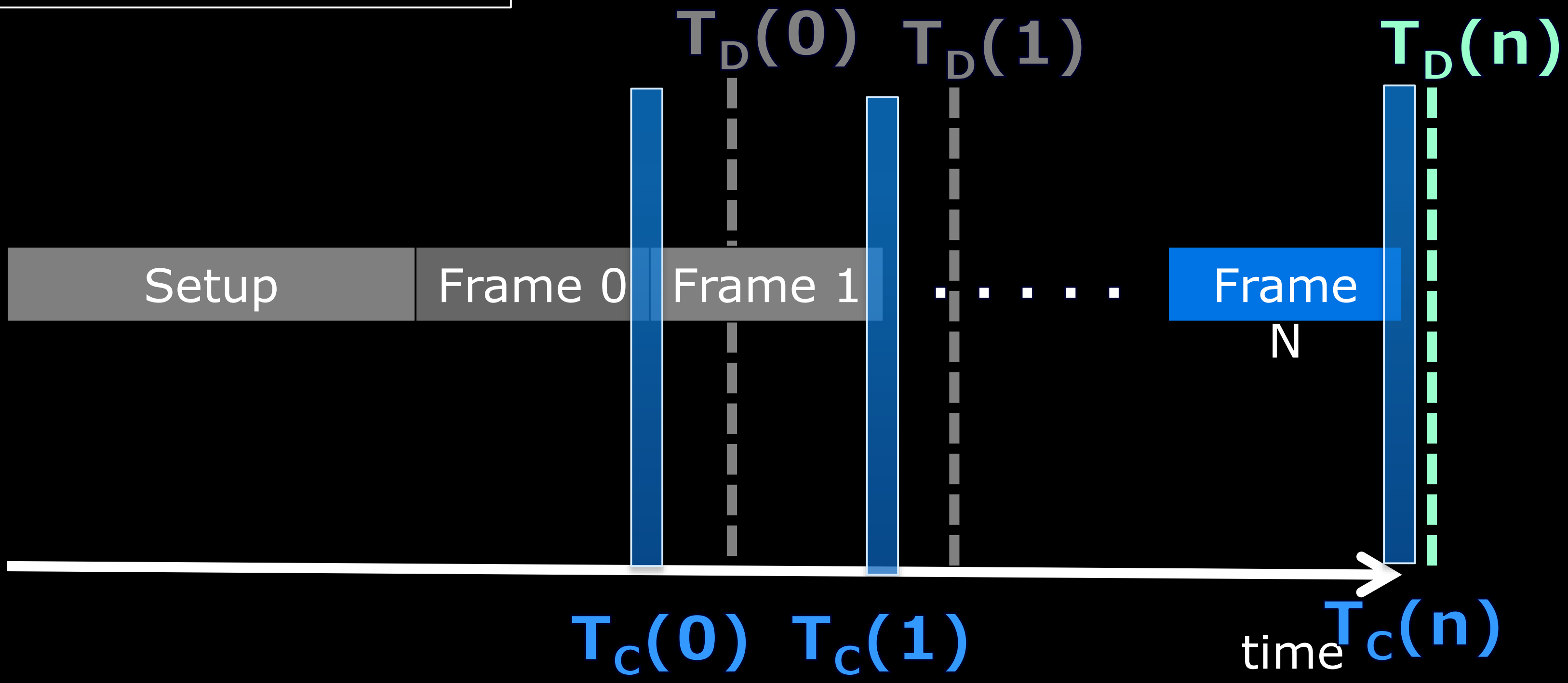
$$\bar{x} \pm 1.96\sigma/\sqrt{n}$$



Mean Frame Delay Recap

Mean Delay Est.:

$$\bar{x} \pm 1.96\sigma/\sqrt{n}$$



Samples Required for Estimate

Since mean delay is $\bar{x} \pm 1.96\sigma/\sqrt{n}$, we can determine the capture image sequence position for $\sigma = 4.5$ milliseconds using:

- 22 samples for 2 ms accuracy
- 85 samples for 1 ms accuracy

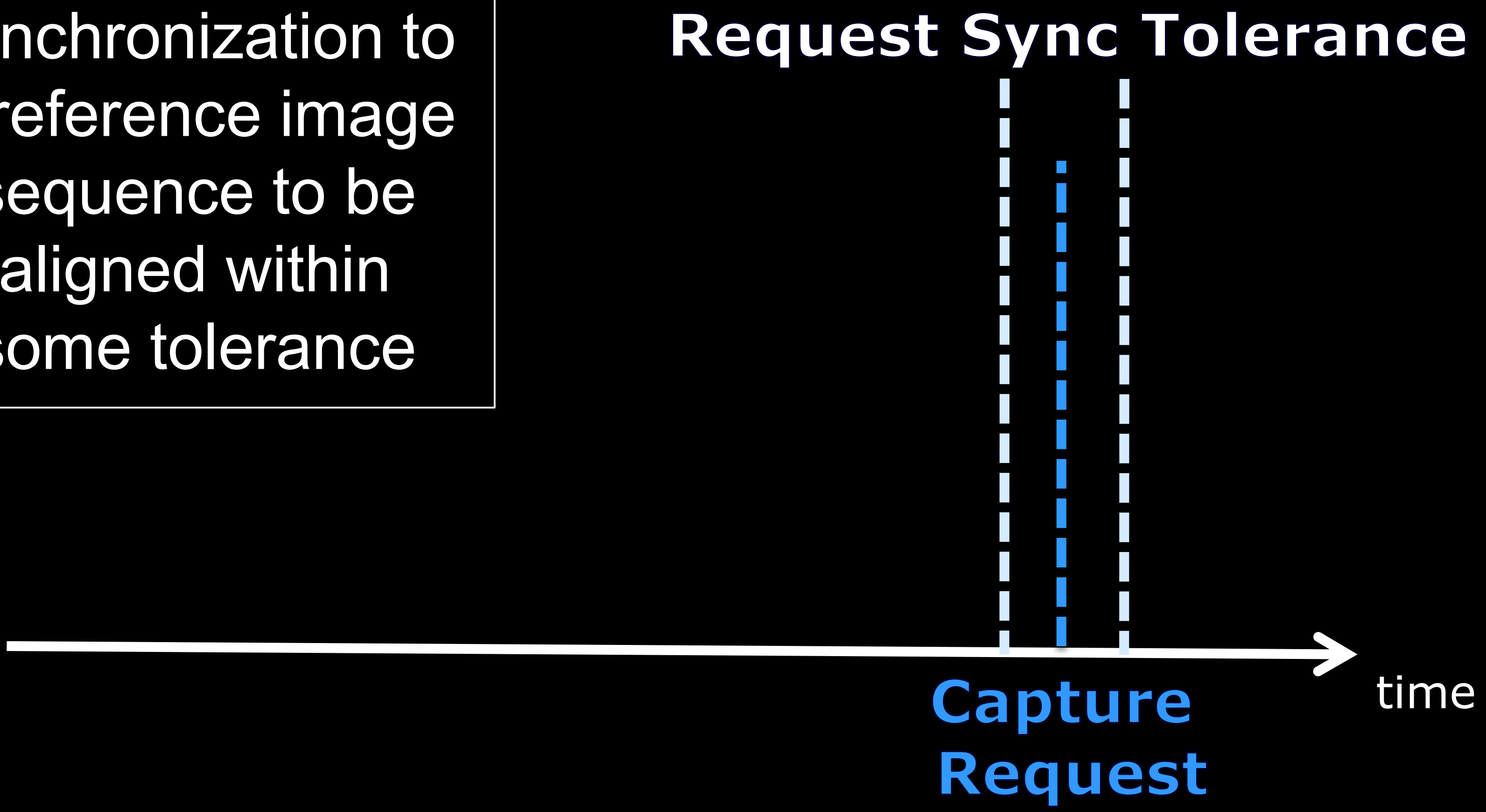
Social Sync Protocol

Estimation Step: Estimating Capture Timestamps to Measure Misalignment of Frames

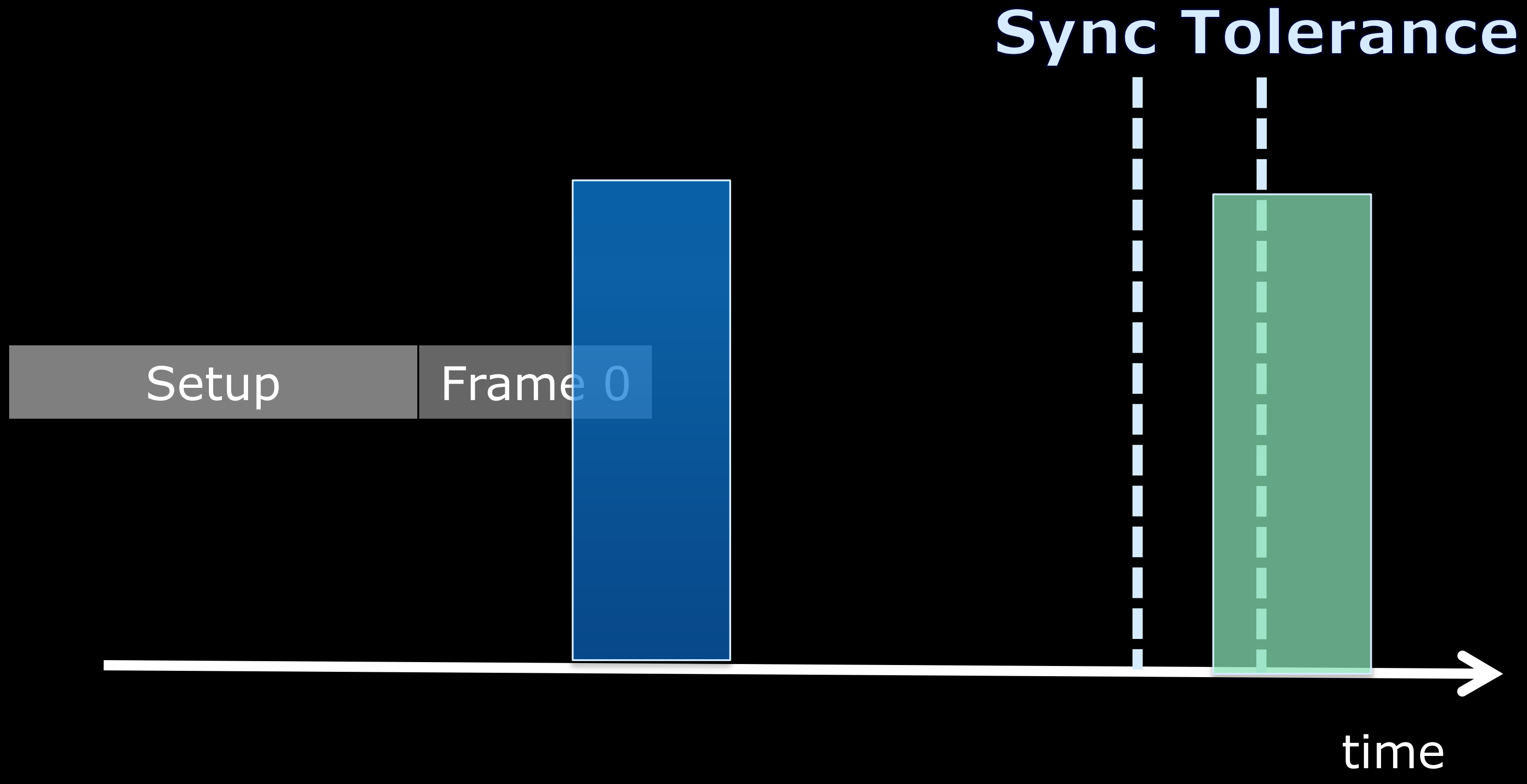
Alignment Step: Using Repeated Attempts at Launching Image Sequence to Reduce Sub-Frame Misalignment

Synchronization Tolerance

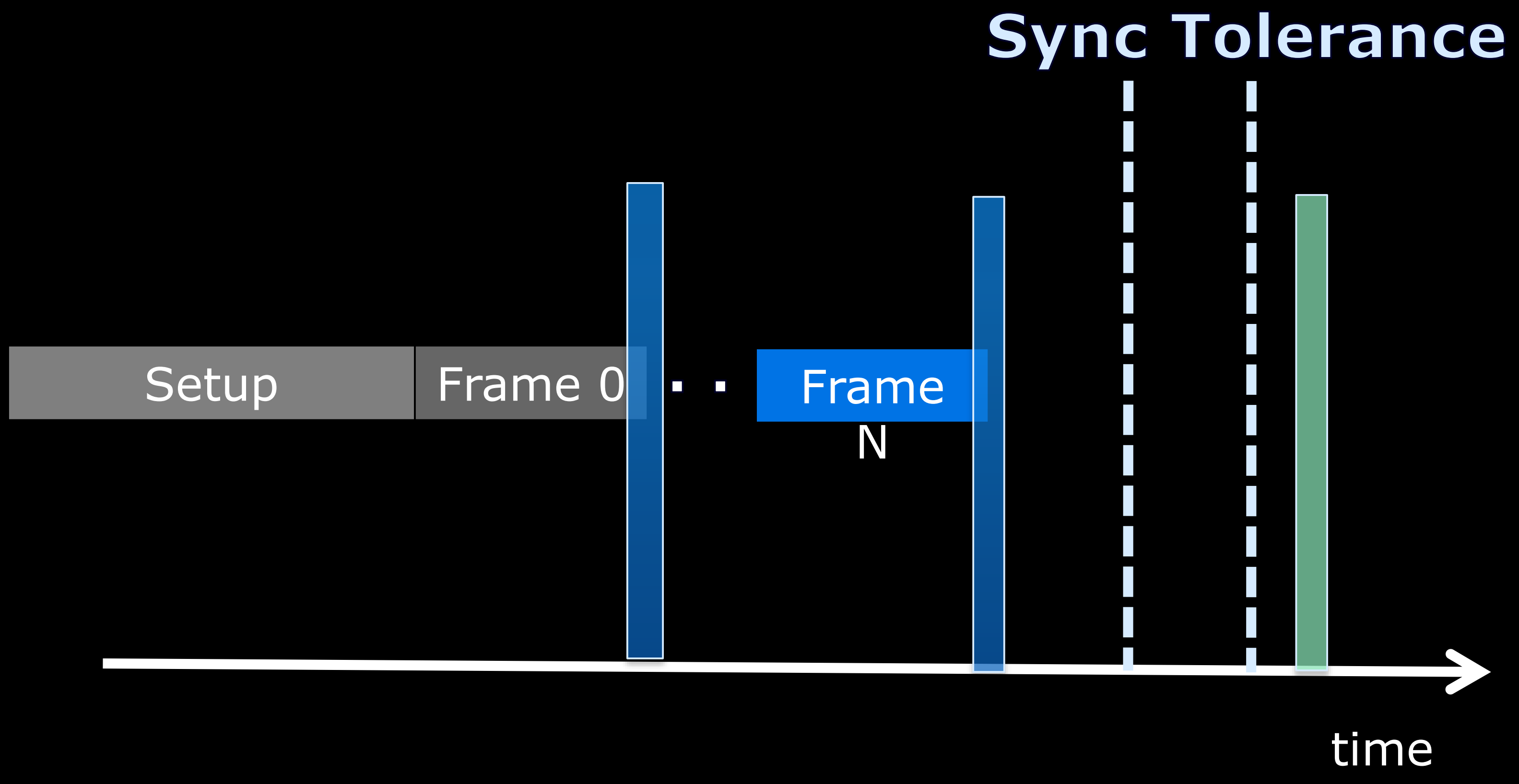
Goal: Want synchronization to a reference image sequence to be aligned within some tolerance



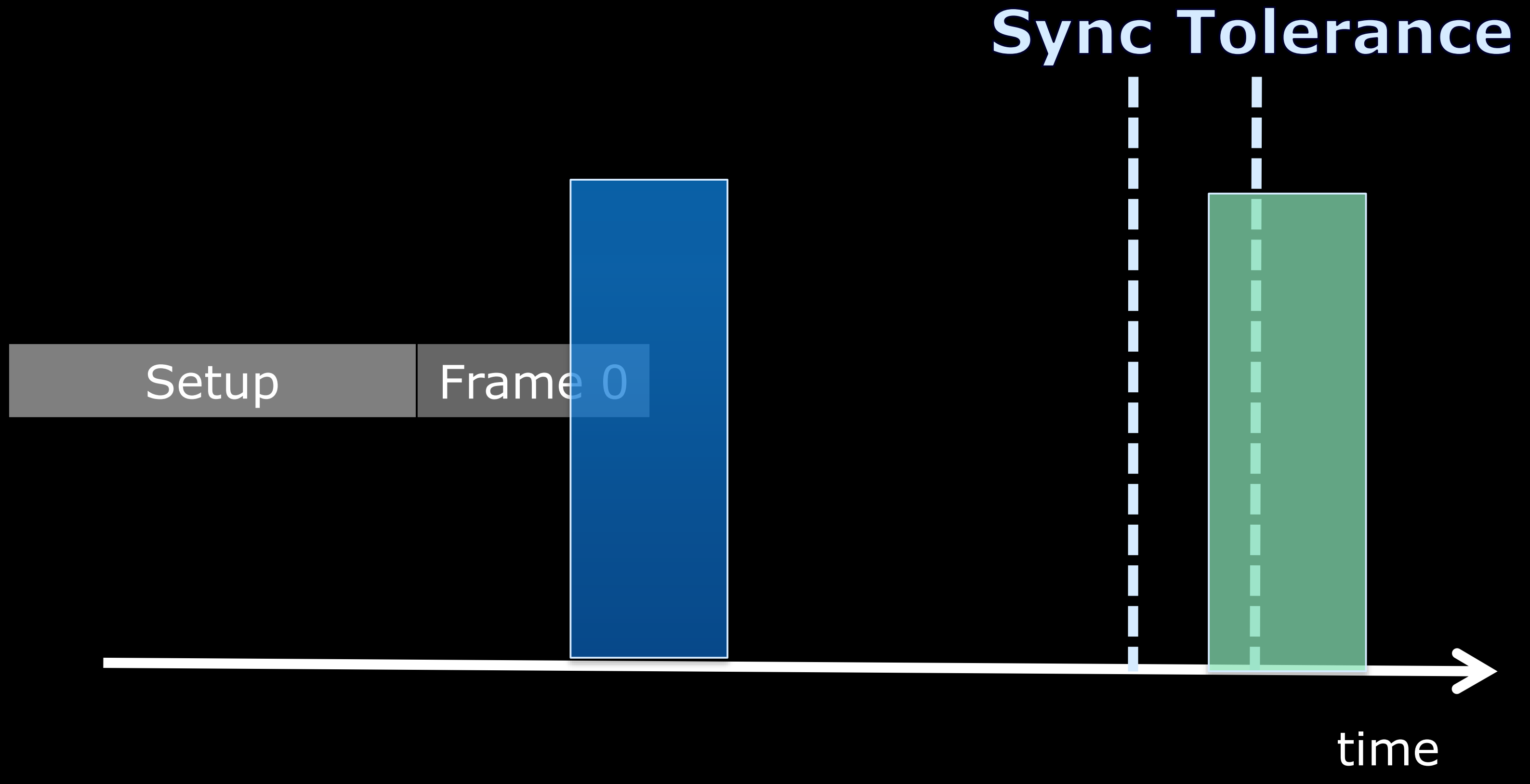
Synchronization Attempt 1



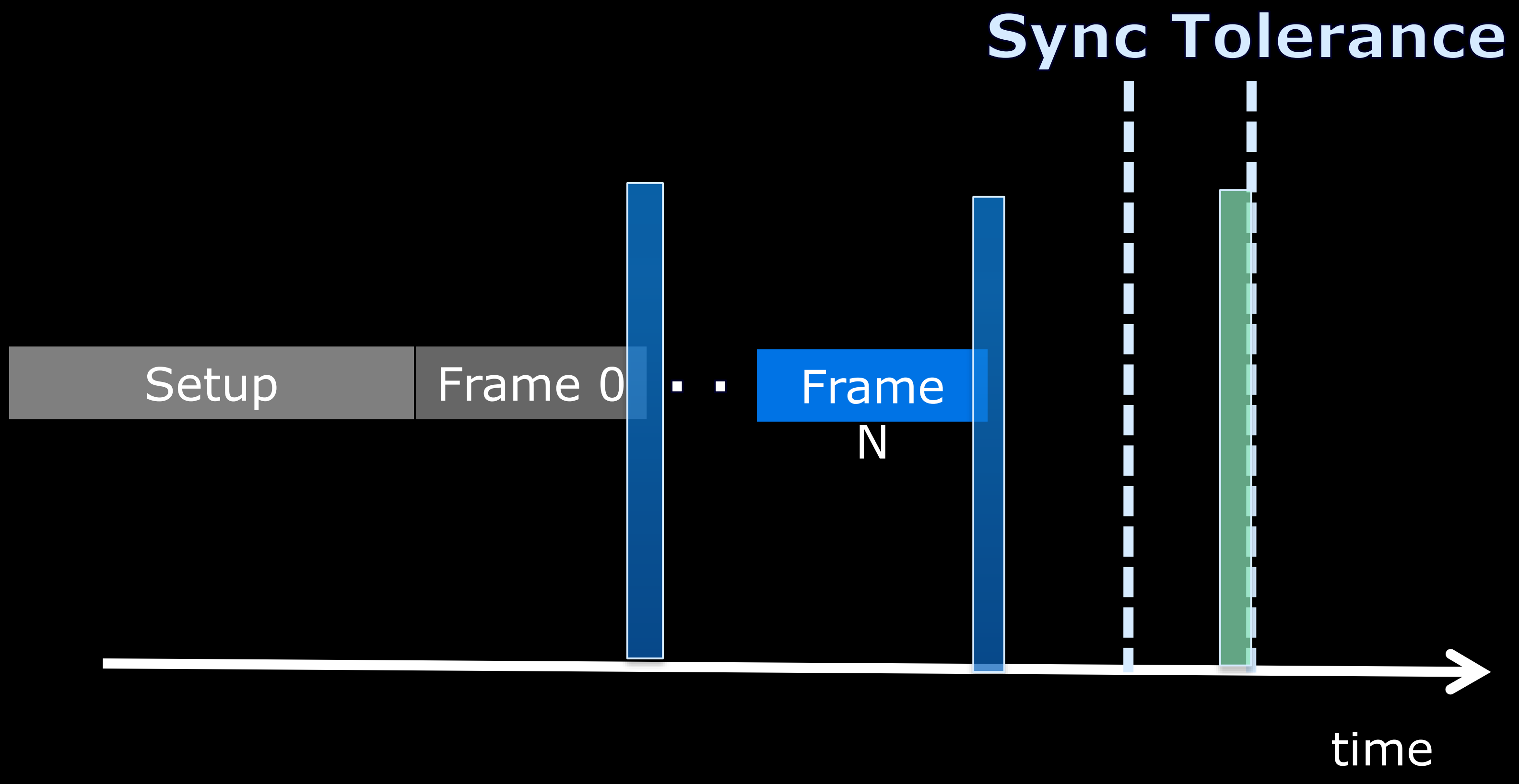
Synchronization Attempt 1



Synchronization Attempt k



Synchronization Attempt k



Obtaining Synchronization

- Let T be the worst case misalignment between frames

- Let t be the range specified by the sync tolerance, where $t \leq T$

Let $p = t/T$, be the probability of synchronization error being reduced for a single phone

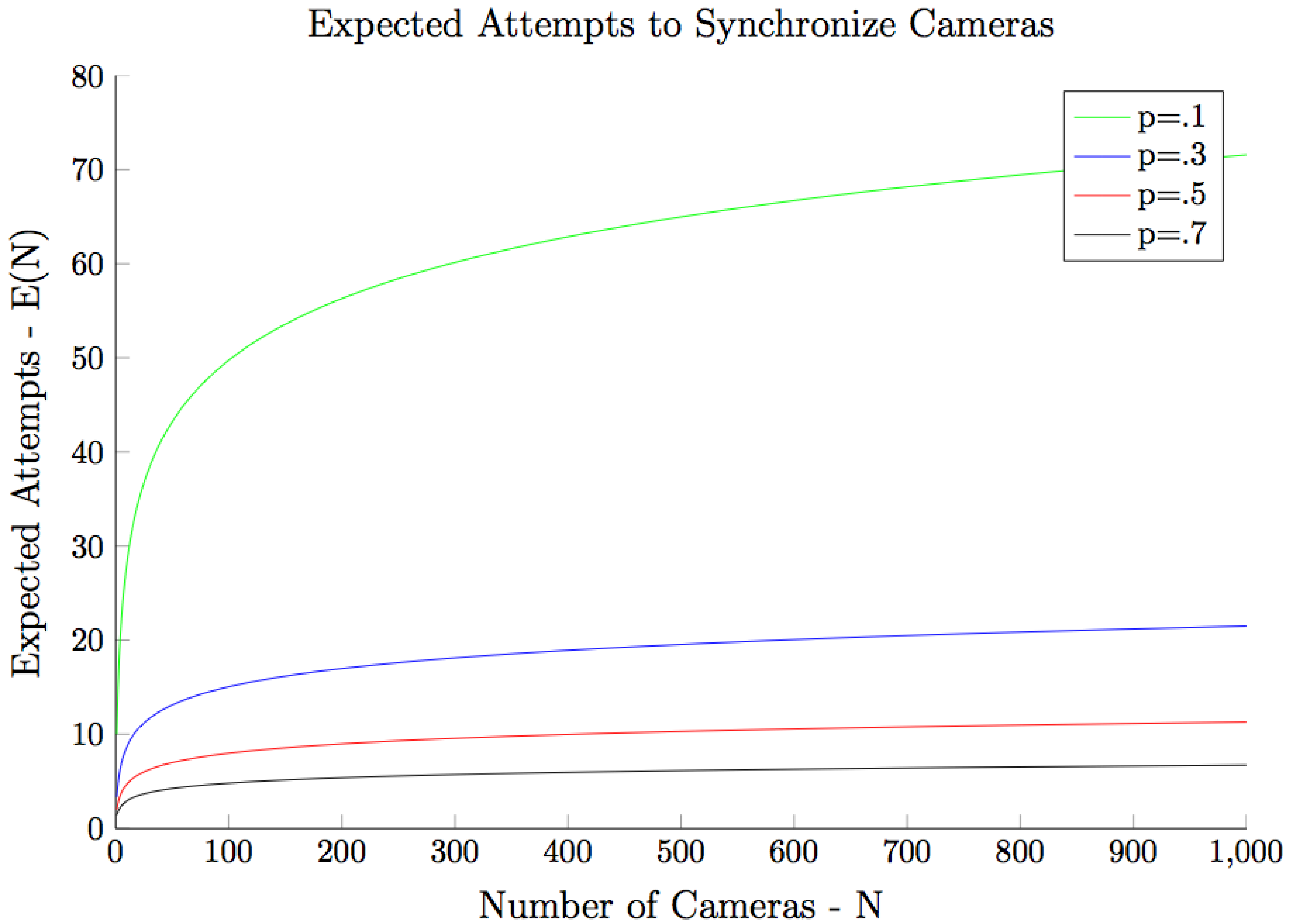
Single Camera Sync Probability After k Attempts

$P_k = 1-(1-p)^k$

Expected Number of Sync Cameras

$n*P_k$

Expected Attempts to Sync N Cameras



- Characterization
- SocialSync Protocol
- Evaluation

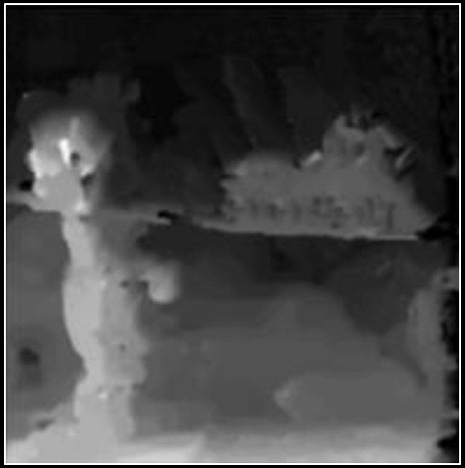
Evaluating Synchronization

Smartphones in array to reduce artifacts not related to time synchronization error

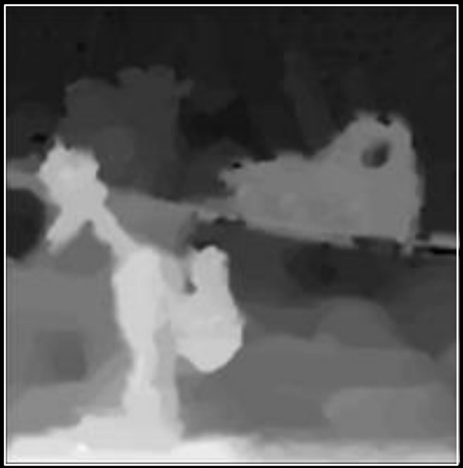
	NaïveSync	SocialSync
4 Cameras	23 ms	5 ms
8 Cameras	35 ms	6 ms

8 Cameras

Naïve Sync



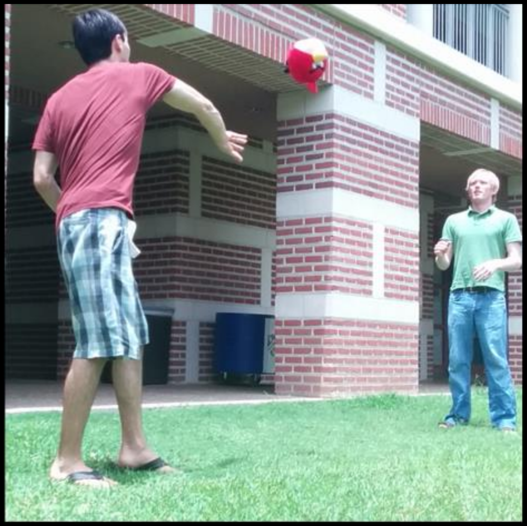
SocialSync



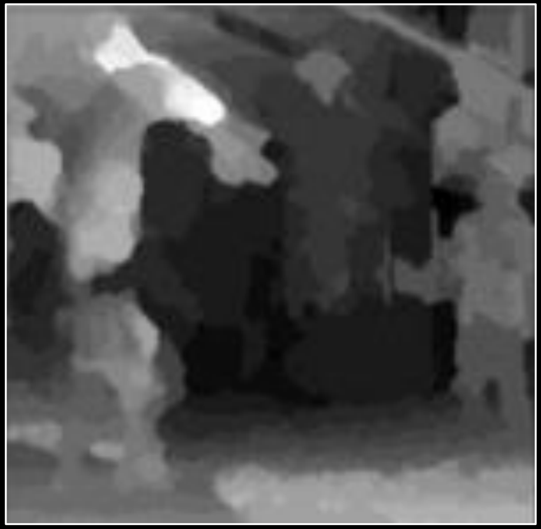
Reference Views

Depth Map

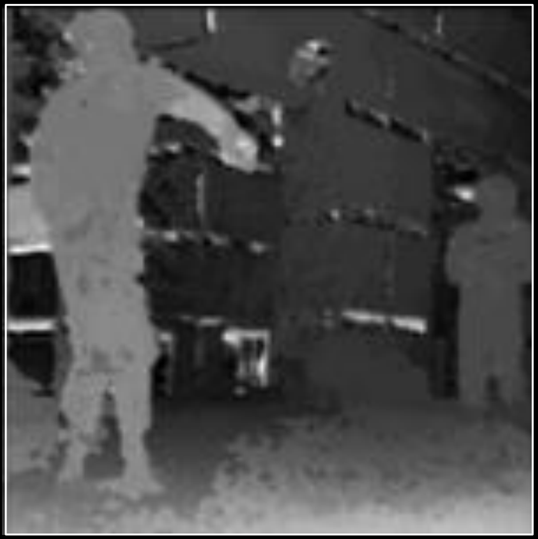
4 Cameras



Reference



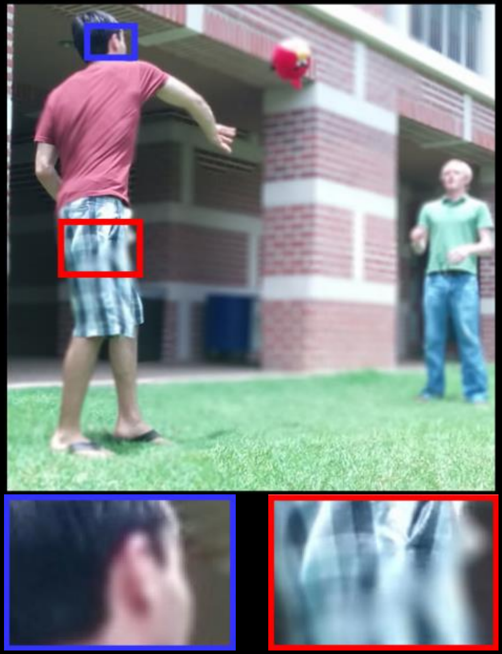
NaïveSync



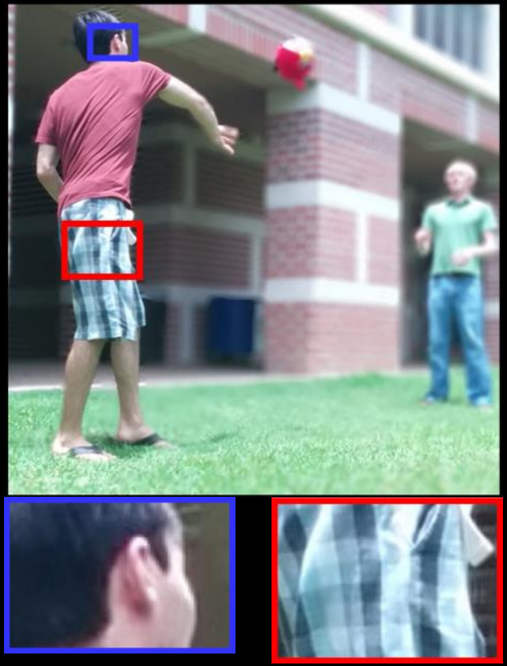
SocialSync

4 Cameras

Reference



NaïveSync



SocialSync

Conclusion

1. Characterize Android Camera
2. Developed SocialSync Protocol
3. Demonstrated SocialSync

Future Work:

- Greater control of Android camera system
- In wild implementation

Acknowledgements

1. Ashok, Lin, Ashu and their students
2. Co-Author Jason Holloway
3. LF4CV Reviewers, Paper Accepted
4. National Science Foundation