Do It Yourself: Multicamera Engineering

- Precision
- Diversity
- Scalability
Can you build your own *Panoptic Studio*?
Can you build your own Panoptic Studio?
Can you build your own *Panoptic Studio*?
Can you build your own *Panoptic Studio*?

Design optimization with **budget** and **space/time** constraints
Budgetary Constraint

Total cost

Cf) Vicon/Qualisys~$60K
Budgetary Constraint

Total cost

Cf) Vicon/Qualisys~$60K

Budget allocation
Space and Time Constraint
Space and Time Constraint

Reproducible
Space and Time Constraint

Reproducible

Modular

Diversity
Space and Time Constraint

Reproducible
Modular
Fast prototyping

Open-source
Point Grey/FLIR Multiple Camera System

https://www.youtube.com/watch?v=svY2NOEBEQA
Point Grey/FLIR Multiple Camera System

USB 3.0
Point Grey/FLIR Multiple Camera System

USB 3.0
Point Grey/FLIR Multiple Camera System

GPIO™ -> Camera -> SS USB -> SS USB -> SS USB -> SS USB -> CPU

GPIO
USB 3.0
Is this suitable?

CPU

GPIO

SS

USB 3.0

Scalability
Is this suitable?

1. Sync signal attenuation
Is this suitable?

2. Limited USB 3.0 support
Is this suitable?

1280x1024 resolution~1.3 MB/image

② Limited USB 3.0 support
Is this suitable?

1280x1024 resolution ~ 1.3 MB/image
200 fps ~ 260 MB/sec

② Limited USB 3.0 support
1280x1024 resolution~1.3 MB/image
200 fps~260 MB/sec
12 cameras~3.12 GB/sec

SSD writing speed: max 650 MB/sec

② Limited USB 3.0 support
1280x1024 resolution ~ 1.3 MB/image
200 fps ~ 260 MB/sec
12 cameras ~ 3.12 GB/sec

SSD writing speed: max 650 MB/sec
28 camera max w/o SSD

7 PCIe max

② Limited USB 3.0 support
Sync signal attenuation
1. Sync signal attenuation

Parts

Precision
Sync signal attenuation

Parts

< $2k
GPIO

① Sync signal attenuation

~$25

< $2k

Parts

$
DIY: Synchronization Module
DIY: Synchronization Module

Triggering signal

2V

3.3V
DIY: Synchronization Module
DIY: Synchronization Module
ECE 101

$V = 3.3V$
\[ V = 3.3\, \text{V} \]
\[ R_d = 10\, \text{k}\Omega \]
\[ I = \frac{V}{R_d} = 0.33\, \text{mA} \]
ECE 101

\[ V = 3.3\text{V} \quad V_s = 3.0\text{V} \]

\[ R_d = 10k\Omega \quad R = 100k\Omega \]

\[ \frac{V}{R_d} = \frac{V_s}{R} \]
$V = 3.3V$

$R_d = 10k\Omega$

$R = 100k\Omega$

$R_d = 100k\Omega$
$ECE\ 101$

$V = 3.3V$

$R_d = 10\, k\Omega$

$R = 100\, k\Omega$

$R_d = 100\, k\Omega$

$R_{\text{net}} = \frac{R^2}{R+R} = \frac{R}{2} = 50\, k\Omega$
ECE 101

\[ V = 3.3V \]
\[ V = 2.7V \]

\[ R_d = 10k\Omega \]
\[ R = 100k\Omega \]
\[ R_d = 100k\Omega \]

\[ R_{net} = \frac{R^2}{R+R} = \frac{R}{2} = 50k\Omega \]
\[ V = 3.3V \quad \lim_{n \to \infty} V = 0V \]

\[ R_d = 10k\Omega \]

\[ R_{\text{net}} = \frac{R^2}{R + R} = \frac{R}{n} \]

\[ R = 100k\Omega \]

\[ R = 100k\Omega \]

\[ R = 100k\Omega \]

\[ R = 100k\Omega \]
ECE 101

GPIO

...
ECE 101

Scalability

$V = 3.3V$

$R_d = 10k\,\Omega$

$R_{var}$

$V = 3.0V$

$R = 100k\,\Omega$

AC/DC power supply

$V = 3.3V, A > 3A$

$\sim$5

GPIO

...
Sync result.
1280x1024 resolution ~ 1.3 MB/image
200 fps ~ 260 MB/sec
12 cameras ~ 3.12 GB/sec

② Limited USB 3.0 support
1280x1024 resolution~1.3 MB/image
200 fps~260 MB/sec
12 cameras~3.12 GB/sec

② Limited USB 3.0 support

Communication bottleneck
DIY: Distributed Computing

260 MB/sec
DIY: Distributed Computing

260 MB/sec  20 MB/sec

JPEG compression
Single Board Computer (SBC)

ODROID XU4
+ ARM based 8 core CPUs
+ 2GB DDR memory
+ 2x USB 3.0 ports
+ Gigabit Ethernet port
+ WIFI module
+ ~$90
260 MB/sec
JPEG compression
Modularity

Diversity

260 MB/sec

JPEG compression + CNN
Single Board Computer (SBC)

nVidia TX2
+ 256 CUDA cores
+ ARM based 4 cores
+ 8 GB memory
+ PCIe (4 USB 3.0)
+ Gigabit Ethernet port
+ WIFI module
+ ~$600
DIY: Distributed Computing

[Diagram showing the connection of cameras (SS USB) to CPUs (CPU) through a network switch (Net. Switch) and eventually to a hard drive (HD).]

- 260 MB/sec
- 20 MB/sec

- 10 Gb/sec
- Up to 60 cameras at HD/200 fps
10 Gigabit Network Switch

+ Support 48 ports
+ 10 Gb (1.25 GB) for uplink
+ ~$1,400
DIY: Distributed Computing

- 260 MB/sec
- 20 MB/sec

Net. Switch

10 Gb/sec
Up to 60 cameras at HD/200 fps

Centralized: 15.6 GB/sec
Decentralized: 1.2 GB/sec
DIY: Distributed Computing

SS writing speed: max 650 MB/sec

10 Gb/sec
Up to 60 cameras at HD/200 fps

Centralized: 15.6 GB/sec
Decentralized: 1.2 GB/sec

260 MB/sec  20 MB/sec

Net. Switch
DIY: Distributed Computing

Decentralized: 1.2 GB/sec
⇒ 3 SSD

260 MB/sec  20 MB/sec
70 Camera System Cost
70 Camera System Cost

- FLIR Blackfly S: $395
- FUJINON varifocal lens: $85
- USB 3.1 cable (5m long): $15

Optics
70 Camera System Cost

- ODROID: $60
- + eMMC module (16G): $25.5
- + Power (5V/4A): $5.5
- Server computer: ~$6,000

x35
70 Camera System Cost

- nVidia 1080 Ti: $800 x6
- nVidia TX2 x3: $600 x3

Costs are depicted for different components:
- Optics
- Computing unit

Costs range from $10k to $40k.
70 Camera System Cost

SSD & HD: $3,000

Optics
Computing unit
GPU
Storage
70 Camera System Cost

80/20 Aluminum Frame & Design: ~$1,400

Graph showing cost breakdown:
- Optics: $40k
- Computing unit: $30k
- GPU: $20k
- Storage: $10k
- Frame: $5k
70 Camera System Cost

Network switch: ~$1,400
70 Camera System Cost

Grand total: ~$60k

- Optics
- Computing unit
- Storage
- Frame
- Network
Demo:
Realtime Markerless Motion Capture
Demo system configuration

11 cameras

Client
Demo system configuration

11 cameras

Client

Net. Switch

Server
Demo system configuration

11 cameras

Client

Net. Switch

Server

1080 Ti
Demo system configuration

Source: https://www.youtube.com/watch?v=YG021vAgrng
Demo: Calibration

• 3D extrinsic parameter calibration using SfM
  – Source code available at:
    • https://github.com/hspark-umn/MulticameraSoftware.git

Image undistortion
SIFT detection
Matching
Bundle adjustment using Ceres solver
3D camera extrinsic calibration
Demo: Recording Software (Client)

- Opensource ARM based camera driver
- Two modes:
  - Secure FTP: transmitting and writing to server
  - TCP/IP: transmitting by feeding to server’s software
- Source code is available at:
  - https://github.com/hspark-umn/MulticameraSoftware.git
Demo: Recording Software (Server)

- Multithreading odroid access
- Realtime 3D body pose reconstruction
  - OpenPose for each image
  - Triangulation given camera matrices
Do It Yourself: Multicamera Engineering

Youbing Wang  Ziwei Li  Shishir Pagad  Hyun Soo Park

University of Minnesota