Final Project

Progress- Week-1

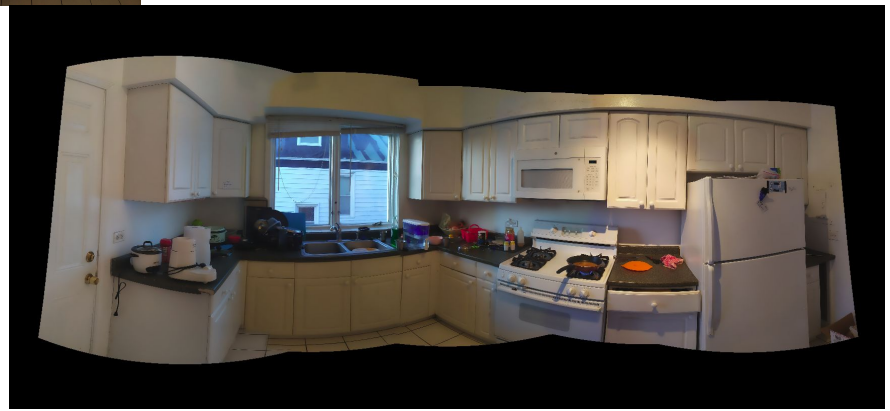
1. Proof of concept - Basic Panorama stitching
2. Setup the hardware and system
3. CAD course (Finished 2 courses in Basic)
4. Gone through code and skimmed through papers

Proof of concept - Basic Panorama stitching (A zero code exercise - openCV out of box)

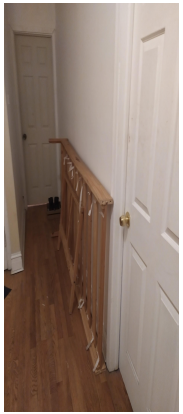
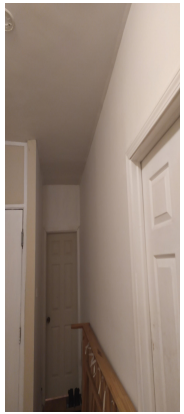
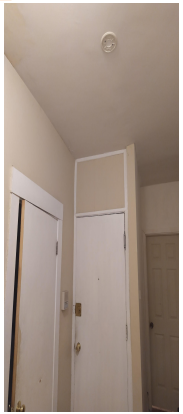
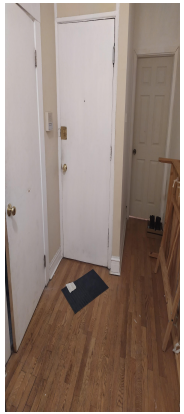








Proof of concept - Basic Panorama stitching (A zero code exercise - openCV out of box)

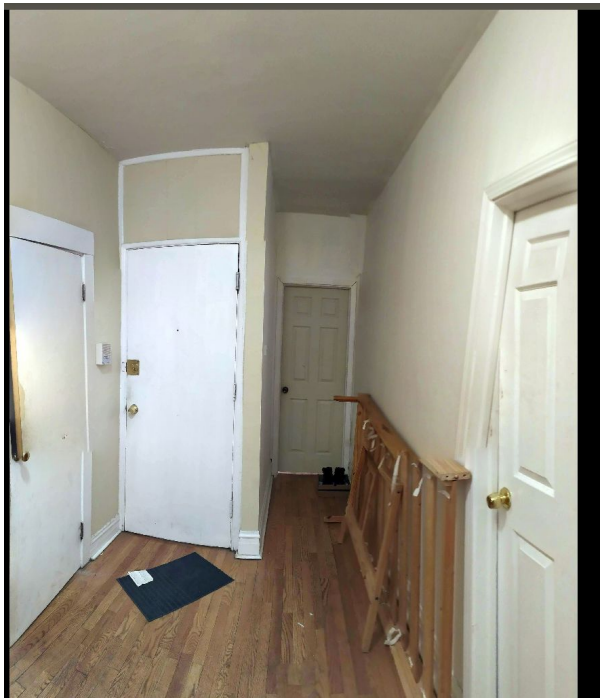





Timing analysis

No of Images	Runtime
3	0.6 S
4	8.1 S
5	11.2 S
6	Failure many times

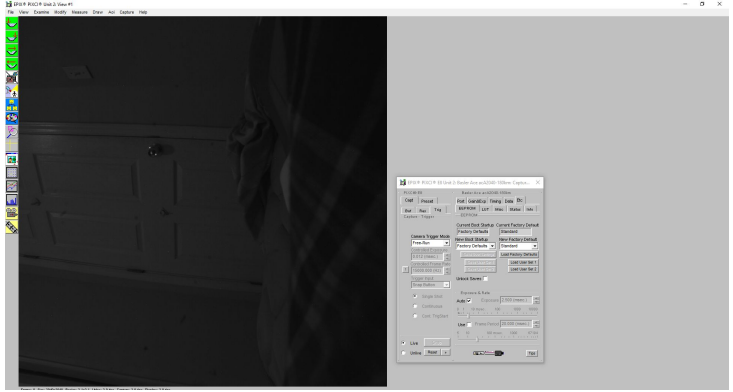
- It is still unclear to me, why runtime would shoot out from 0.6 S to 8.1 S when moving from 3 to 4 images. I will need to spend more time on this, if needed
- Some possible explanations could be
  - Combinatorial run time
  - Run time could actually depend on the nature of images

Google is too smart nowadays



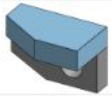
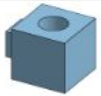
Hardware and system setup

*Running as fast as possible to minimise downtime



1. A camera can be viewed with the system.
2. Problems
    a. Only one PoCl cable is working (Not 100 % sure, It might be one of the cameras as well)
        i. **Action taken:** Got in touch with Basler and their distributing partner machine vision store. Expected lead time - 2-3 weeks. These are expensive (5m - $174)
        ii. Amazon - delivery by April end
    b. No Wifi in the system
        i. **Action taken:** Ordered a usb wifi card
    c. Missing power cord
        i. German was kind to help me out.

On Shape progress

	**revolve_exercise** Main	10:03 PM Apr 18	me	me
	**exercsie2** Main	7:08 PM Apr 18	me	me
	**keyhole_design** Main	6:41 PM Apr 18	me	me
	**trail** Main	5:25 PM Apr 18	me	me

Code readup

1. My analysis of the coding infrastructure so far
    a. The driver code for frame grabbers on windows looks great and should be reusable
    b. No code so far on image stitching, Some progress on multi View reconstruction but very hard coded (Will lose more than what I can gain)
    c. Gantry robot code may be reusable (Is not useful to me at this stage. Will decide later).

```
// Assign Parameters' Value
x1_step1_xoffset << 64.901778151504, -8.42209010747119, 130.189970511086;
x1_step1_gain << 0.0688013705172384, 0.0739789493547617, 0.0762121584900576;
b1 << -0.84478751913874262, 0.62482150249037582, -0.52602559945415883, 0.68828147510444948, 0.024600675497650798, -0.61150543662932011, 1.313895903284739, -0.813448568495311, 0.48353152862540655, 0.904985314767961
b2 << -0.144840488872127, -0.0614561747222241, 0.0867888925487940;
IW1_1 << -0.0693775059374704, -0.898235121775880, 0.141430277664874,
    -0.507919663751378, -0.247756823567594, -0.807591548820987,
    0.411856934158734, 1.17101363357360, 0.0621108190725948,
    -0.140658720314116, 0.389867218217533, 0.680401201935144,
    -0.474985401996206, 0.192974772528405, 0.728134838467866,
    0.799505410400201, -0.0938881053618274, -0.242755509709435,
    0.151009494364861, 0.143599900468799, 0.918347954839469,
    -0.285210395188397, 0.747275783134957, 0.749685485657332,
    0.00460007717146710, -0.526688404879907, 0.304056241681414,
    0.888294323570919, -0.0917992029087001, -0.305748002062219;
LW2_1 << 0.199923352051252, -0.372035688711793, -0.219354459133026, 0.0343504617404127, 0.192897226766938, 0.706831207678236, 0.538999456639056, 0.292301089276802, -0.123530168564321, 0.768370626724965,
    0.655304808299187, -0.709335925135639, -0.447538845512842, 0.480105893632707, 0.00822693786532985, -0.399009670716074, 0.877534246312236, 0.727588309502385, -0.0384329926091908, -0.323787141842497,
    0.587893354358607, -0.0525796610607098, -0.180044112613425, -0.765476390881385, 0.369534113176288, 0.0638993512151596, 0.0352104798223909, -0.539873284397187, 0.750257393621354, 0.0348348453881138;
y1_step1_gain << 0.0666666666666667, 0.1000, 0.0800;
y1_step1_xoffset << -5, 10, -35;
```

```
MatrixX3d cameraMatrix::Triangulate() {
    MatrixX2d matchedPts( 7 * 2, 2 ); // 2*N by 2 Matrix
                                      // Format: x11,  y11     | - Matched points pair #1
                                      //         x12,  y12     ~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                      //         x21,  y21     | - Matched points pair #2
                                      //         x22,  y22     ~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                      //         ...   ...     ...........................
    matchedPts << 1743.0, 859.0,      // | - Matched points pair #1
        82.0, 905.0,                  // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1665.0, 852.0,                // | - Matched points pair #2
        165.0, 908.0,                 // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1593.0, 846.0,                // | - Matched points pair #3
        249.0, 912.0,                 // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1751.0, 842.0,                // | - Matched points pair #4
        410.0, 911.0,                 // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1833.0, 841.0,                // | - Matched points pair #5
        488.0, 910.0,                 // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1915.0, 839.0,                // | - Matched points pair #6
        566.0, 909.0,                 // ~~~~~~~~~~~~~~~~~~~~~~~~~~~
        1964.0, 842.0,                // | - Matched points pair #7
        611.0, 912.0;
```

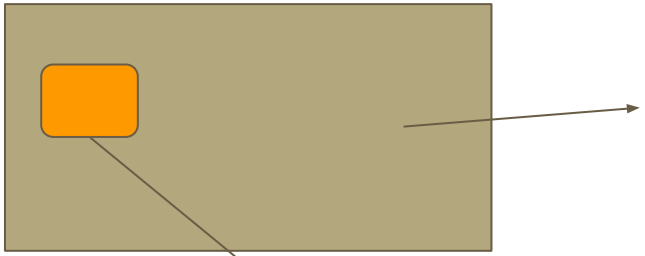
```
// Assign Camera Matrix
camMatrix[0] << 2252.96101573020, 0.0, 0.0,
    0.0, 1870.66489933229, 0.0,
    534.866659707105, 803.932066715002, 1.0000,
    0, 0, 0;

camMatrix[1] << 623.745637198259, -546.685240671789, -0.634762728031709,
    -162.033780856155, 1850.86475470962, 0.0129411181628372,
    2852.31684862298, 899.455589640930, 0.772598735801735,
    -4096144.90909403, 208348.130376731, 616.676581224207;
```

Some decision choices insights

1. Why a NIR camera?
2. Why windows? Can I try setting up the whole system in ubuntu?
3. Why C++? Due to speed?
4. Was the process very compute intensive at any point? (Frame grabbers. GPU)
5. Why not marker based pose estimation? (Just to ensure that all options have been thought and consciously eliminated)
6. How to go from poses to behaviour?
7. How costly is a mistake? A very important deep learning question. What is the impact of a false positive? A high accuracy system with some false positives or a low accuracy system with zero false positives?

Some decision choices insights

8. What pixel /mm is okay? To decide mounting height?



Field of view (1) - 242 cm x 217 cm

Minimum satisfactory view of the object interest of a known size (2)

1. Pixel area (object of interest) / actual area (object of interest) = pixel per mm
2. Field of view / pixel per mm = camera resolution
3. Play around with mounting height and focal length to capture the field of view of interest at a realistic and not so awkward height

Note: In our case, our camera resolution but we have a varifocal length and mounting height as DoFs

Some decision choices insights

9. What kind of pose estimation is needed 2D or 3D?

If 2D → we can begin training quickly with just few images and labelling. Will 2D data be enough for making the necessary behaviour predictions?

If 3D → How to acquire 3D ground truth? A stereo reconstruction might be a necessary precursor before we step in.

Task bucket list (1 week is roughly 30h)

1. Get image stitching to run faster for our narrow use case - May require hardware setup first - 30h
2. Setup drivers in ubuntu and switch to ubuntu system - 20h
3. Multiview reconstruction.
    a. Do 3-D reconstruction from multi view datasets available online - 3 weeks
    b. Integrate and test with our cameras - 1 week
4. Train Deepcut for pose prediction - 3.5 weeks (learning curve due to tensorflow and processing video sequences)
    a. Test this on real lab data and integrate into the system - 1 week
5. Design two camera mounts one for image stitching and one for image reconstruction - 20h
6. Setup the total hardware - 20h
7. How to predict behaviour from pose graph movements - Not sure of estimate - need more time for exploration

Plan - week-2 (30 h)

1. Setup drivers in ubuntu and switch to ubuntu system - 20h
2. Design camera mount for image stitching - 5h

Exection - week-2 (30 h)

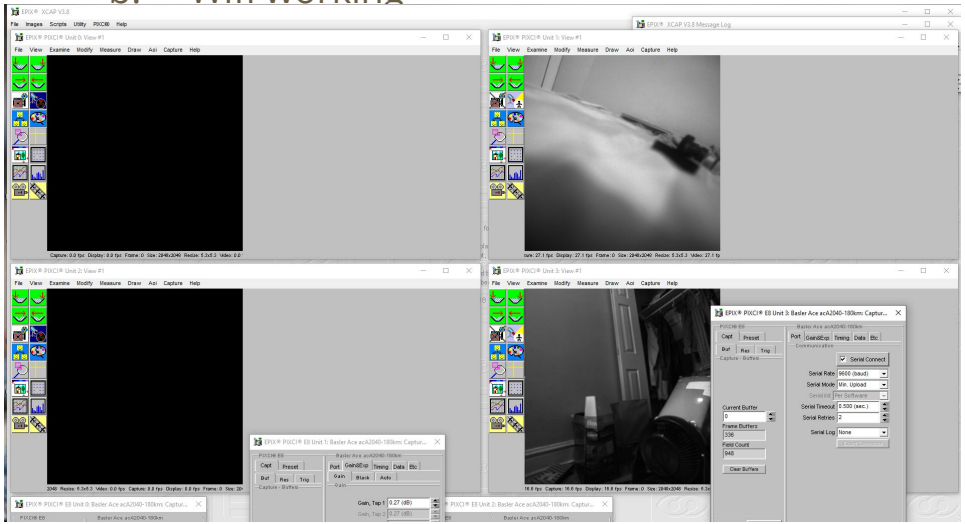
1. Setup drivers in ubuntu and switch to ubuntu system - 20h
   a. ~~Ubuntu and wifi driver installation done - 10h~~
   b. XCAP and XClib driver installation - In progress
2. Design camera mount for image stitching - 5h (An underestimation)
   a. ~~Learn parts based modelling in on shape. - 5h~~

Progress - Week 2

1. Installed Ubuntu and wifi drivers - 10h
2. Checked problem with Camera and PoCL wires
    a. Its not PoCL wires that’s not working: It is the camera
3. Got to learn more on parts based design in on shape - 5h
4. In progress - Driver installation in Ubuntu - 5h (Should be done by in another 5-6 hrs)

Hardware statues

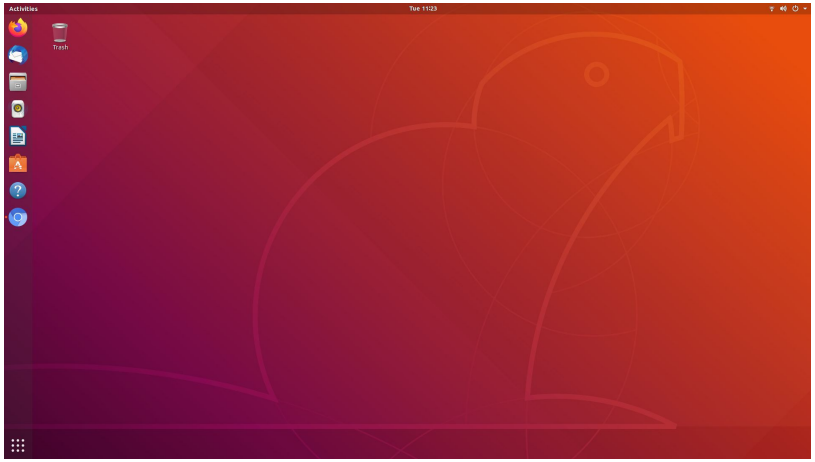
1. Camera C2 and C1 are not working
2. Computer is up and running
    a. Ubuntu installed
    b. Wifi working



Camera	PoCL wires	Status
C4	C1F1	Working
C4	C2F2	Working
C4	C3F3	Working
C2	B1F1	Not Working
C2	B2F2	Not Working
C2	B3F3	Not Working
C3	C1F1	Working
C3	C2F2	Working
C3	C3F3	Working
C1	C2F2	Not Working
C1	C1F1	Not Working
C1	C3F3	Not Working

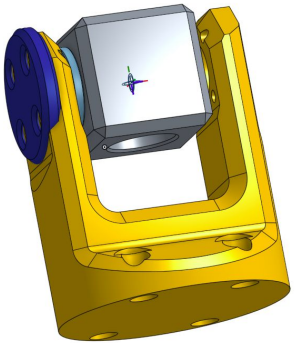
System and Drivers

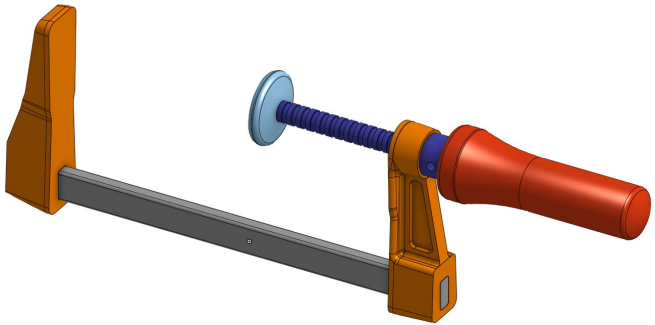
1. Ubuntu 18 setup
2. Driver installation in progress



3D modelling progress - Just learning parts based modelling

	Name	Modified ▾	Modified by	Owned by
	CAD_Assembly3 Main	10:10 PM Yesterday	me	me
	CAD_Assembly2 Main	7:17 PM Yesterday	me	me
	CAD_Assembly' Main	4:58 PM Yesterday	me	me





Plan next week

1. Design a basic holder for the camera - It might not meet our specifications but this is just a trial before fitting in the details -10 h

2. Design / Think about a system which based on initial startup should be able to stitch images together with known transformation. A trail run before getting actual mounts - 10h
3. Try out open source implementation of deep lab cut

   (or)

2. Get started with 3D reconstruction - two camera based triangulation- 18 h

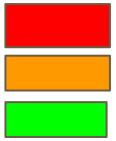
3. A survey on the precision of stereo system's reconstuctionaccuracy

Proposed plan

1. Design a basic holder for the camera - It might not meet our specifications but this is just a trial before fitting in the details -10 h
2. Design / Think about a system which based on initial startup should be able to stitch images together with known transformation. A trail run before getting actual mounts - 10h
3. Try out open source implementation of deep lab cut -5h

Progress status - Week 3

Legend

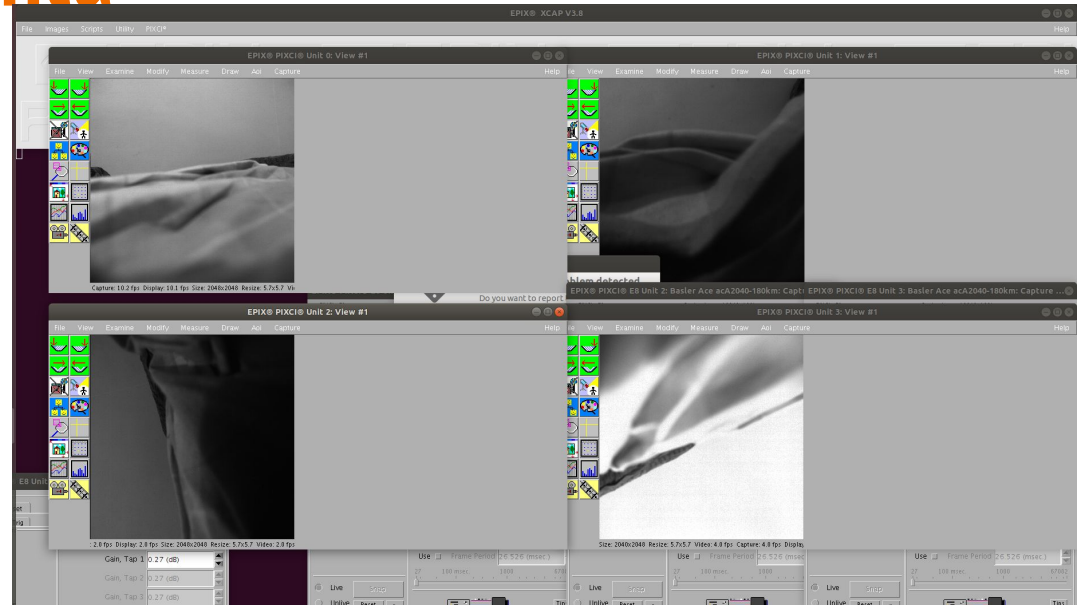


- Did not make any progress
- Partially completed (In progress)
- Fully completed

1. Design a basic holder for the camera - It might not meet our specifications but this is just a trial before fitting in the details -10 h
2. Design / Think about a system which based on initial startup should be able to stitch images together with known transformation. A trail run before getting actual mounts - 10h
   a. ~~Drivers are installed verified - XCAP works on Ubuntu~~
   b. ~~Wrote basic driver code to interface framegrabber through XClib~~
   c. Stitch images together with images on camera (pushed to next week)
3. ~~Try out open source implementation of deep lab cut -5h~~

XCAP works on Ubuntu and all cameras / PoCL cables works on ubuntu

1. Had to check licenses
2. Had installation problems
3. Had to learn xcap usage
4. Had to learn XCLIB driver level code

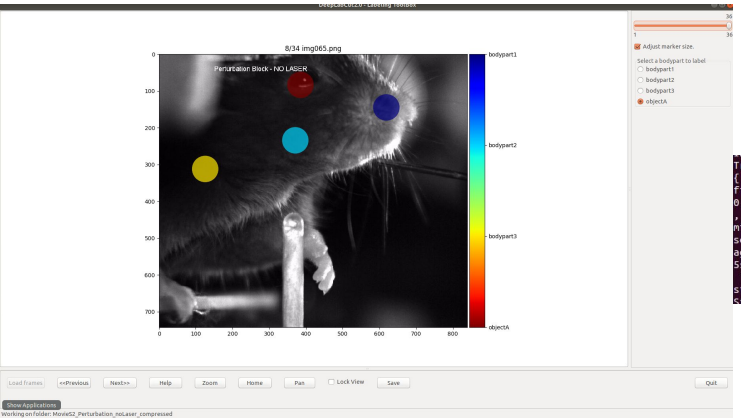


C++ code for interacting with frame grabber code

1. Completely avoided the previous code (It saves images to disk and then reads it for processing, lots of extra generic thing)
2. Wrote a short and specific code for our specific use case

Just dabbled at deep lab cut

```
In [2]: deeplabcut.create_new_project('demo', 'senthil', ['/home/senthilpalanisamy/work/courses/final_
   ...: project/DeepLabCut/examples/cloned-DLC-repo/examples/Reaching-Mackenzie-2018-08-30/videos/Movl
   ...: eS2_Perturbation_noLaser_compressed.avi'])
Created "/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/videos"
Created "/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/labeled-data"
Created "/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/training-datasets"
Created "/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/dlc-models"
Attempting to create a symbolic link of the video ...
Created the symlink of /home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/cloned-DLC-repo/examples/Reaching-Mackenzie-2018-08-30/videos/MovieS2_Perturbation_noLaser_compressed.avi to /
home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/videos/MovieS2_Perturbation_noLaser_compressed.avi
/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/videos/MovieS2_Perturbation_noLaser_compressed.avi
Generated "/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/config.yaml"

A new project with name demo-senthil-2020-05-04 is created at /home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples and a configurable file (config.yaml) is stored there. Change the param
eters in this file to adapt to your project's needs.
 Once you have changed the configuration file, use the function 'extract_frames' to select frames for labeling.
. [OPTIONAL] Use the function 'add_new_videos' to add new videos to your project (at any stage).
Out[2]: '/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/config.yaml'

In [3]: path_config = '/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/config.yaml'

In [4]: deeplabcut.extract_frames(path_config, 'automatic', 'kmeans')
Config file read successfully.
Do you want to extract (perhaps additional) frames for video: /home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/cloned-DLC-repo/examples/Reaching-Mackenzie-2018-08-30/videos/MovieS2_P
erturbation_noLaser_compressed.avi ?
yes/noyes
Extracting frames based on kmeans ...
Kmeans-quantization based extracting of frames from 0.0  seconds to 8.53  seconds.
Extracting and downsampling... 256  frames from the video.
256it [00:00, 324.93it/s]
Kmeans clustering ... (this might take a while)
```




```
Training parameter:
{'stride': 8.0, 'weigh_part_predictions': False, 'weigh_negatives': False, 'fg_fraction': 0.25, 'weigh_only_present_joints': False, 'mean_pixel': [123.68, 116.779, 103.939], 'shuffle': True,
fix': '/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04/dlc-models/iteration-0/demoMay4-trainset95shuffle1/train/snapshot', 'log_dir': 'log', 'gl
0.8, 'location_refinement': True, 'locref_stdev': 7.2801, 'locref_loss_weight': 0.05, 'locref_huber_loss': True, 'optimizer': 'sgd', 'intermediate_supervision': False, 'intermediate_supervisi
, 'regularize': False, 'weight_decay': 0.0001, 'mirror': False, 'crop_pad': 0, 'scoremap_dir': 'test', 'batch_size': 1, 'dataset_type': 'default', 'deterministic': False, 'crop': True, 'cropa
minsize': 100, 'leftwidth': 400, 'rightwidth': 400, 'topheight': 400, 'bottomheight': 400, 'all_joints': [[0], [1], [2], [3]], 'all_joints_names': ['bodypart1', 'bodypart2', 'bodypart3', 'obje
set': 'training-datasets/iteration-0/UnaugmentedDataSet_demoMay4/demo_senthil95shuffle1.mat', 'display_iters': 1000, 'init_weights': '/home/senthilpalanisamy/anaconda3/envs/DLC-CPU/lib/python3
ages/deeplabcut/pose_estimation_tensorflow/models/pretrained/resnet_v1_50.ckpt', 'max_input_size': 1500, 'metadataset': 'training-datasets/iteration-0/UnaugmentedDataSet_demoMay4/Documentation
5shuffle1.pickle', 'min_input_size': 64, 'multi_step': [[0.005, 10000], [0.02, 430000], [0.002, 730000], [0.001, 1030000]], 'net_type': 'resnet_50', 'num_joints': 4, 'pos_dist_thresh': 17, 'pr
 '/home/senthilpalanisamy/work/courses/final_project/DeepLabCut/examples/demo-senthil-2020-05-04', 'save_iters': 50000, 'scale_jitter_lo': 0.5, 'scale_jitter_up': 1.25, 'output_stride': 16, 'c
stride': 2}
Starting training...
```

Some thoughts

An end to end infrastructure

1. Frame sampling from video using k-means clustering
2. Labeling frames using wxPython GUI
3. Train network on labelled images
4. Might be forced to use docker/ Anaconda. Dependence of wxPython is annoying

Some observations

1. The camera is monochrome and NIR. Hopefully we don’t miss out on anything useful.
2.

Plan for the quarter

Quarter Goal: Have infrastructure up and running for 2D image stitching, MultiView reconstruction and Pose prediction

Week No	Proposed Plan
5	Image stitching, Mount for the camera in Onshape, Install GPU drivers and test GPU usage for ubuntu Some trial on Deep Pose prediction
6	Multiview reconstruction on online dataset. Design a 80/20 mount for both image stitching and multiview reconstruction. Network setup and training infrastructure for deeplab cut.
7	MultiView reconstruction on online dataset. Look into Camera extrinsic and intrinsic calibration required
8	Multiview reconstruction from Camera. A look into 3D pose prediction networks. Integration of debugging tools.
9	Multiview reconstruction from Camera. Look into camera synchronisation issues
10	Integrate and test the system the camera system

Plan for next week

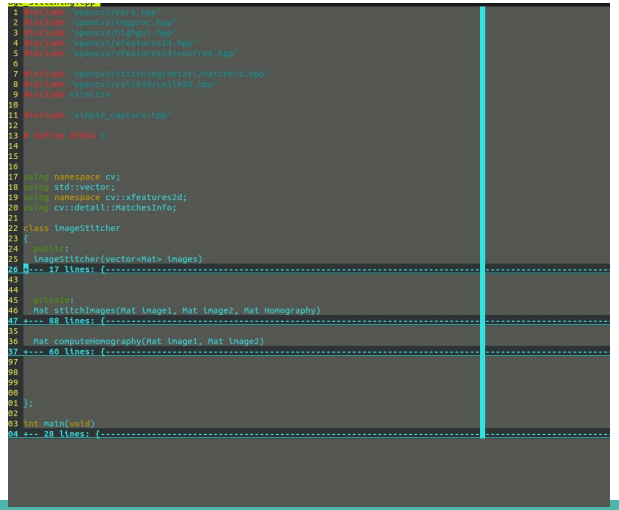
1. Image stitching - 5h
2. Mount for the camera in Onshape - 20h
3. Install GPU drivers and test GPU usage for ubuntu - 5h
4. Some trial on Deep Pose prediction - 5h

Plan for next week

1. Image stitching - 5h (15 h)
2. Mount for the camera in Onshape - 20h
3. Install GPU drivers and test GPU usage for ubuntu - 5h (5h)
4. Some trial on Deep Pose prediction - 5h

Image stitching pipeline constructed from scratch

1. OpenCV image stitchers don’t allow access to the homography matrices directly or atleast its not immediately obvious after digging through their code base for a few hours.
2. Constructed an image stitching pipeline from scratch. Tested it on offline images.
3. Built the pipeline for interfacing Camera.



PÁTEK 15. ČERVENCE 2016 · ČÍSLO 164 CENA 18 KČ

ÚSPORY NA DŮCHOD. Fondy Čechům prakticky nic nevydělávají ...10

PELHŘIMOVSKÝ
deník
vysočina
www.pelhrimovsky.denik.cz

NOVÝ ŠÉF BIS?
České agenty povede muž se „školou" Jamese Bonda
strana 9

TV MAGAZÍN
Přehledný televizní program na celý týden
uvnitř

ŽIRAFÍ SAMEC ZUBERI SE PO TŘECH LETECH ODVÁŽIL VEN

STRACH JE PRYČ. Zuberi, největší, nejvyšší a nejtmavší ze tří žirafích samců v jihlavské zoologické zahradě, se po třech letech konečně odhodlal a už navštěvuje i venkovní výběh. Chovatelé mu ale jeho strach museli pomoci překonat kapkou násilí. Vypozorovali totiž, že se Zuberi bojí vozíku, ze kterého leští obří skla vnitřního výběhu žiraf. A tak ho pomocí tohoto vozíku do venkovního výběhu poprvé před měsícem a v podstatě vytlačili. Takto to chovatelé praktikovali celý týden, než Zuberi pochopil, že mu venku nic nehrozí. Teď už se do výběhu odváží sám.
Foto: Jan Černo ...2

Dobrý den, Vysočino

Před 25 lety (tedy 15. 7. 1991, kdy ještě existovalo Československo) vznikla Policie České republiky. Zmutovala z tehdejších českých složek Sboru národní bezpečnosti (SNB).

Po těch 25 letech nám sedí v parlamentu lidé jako Zdeněk Ondráček, bývalý příslušník Pohotovostního pluku SNB (dnes předseda sněmovního podvýboru pro Policii ČR), či Bronislav Schwarz, poslanec, jenž (ANO, už v demokratických časech) řídil kdesi na severu městskou policii, muž, který teď klidně prohlásí: „Radši si obětuju trošku svobody, když vím, že budu mít pořádek a že mi na ulicích nebudou lítat vandalové a nebudou mi tady neorganizovaně šířit paniku v manifestacích!" – Nedávno jsem četl *souhrnnou situační zprávu* Ministerstva vnitra (za 1. čtvrtletí 2016) o extremismu v ČR. Bylo to dost šílené čtení. Ne, duch SNB neoživá. On žije.

Roman Streichsbier

Předškoláky letos na školu přípravná třída nepřipraví

Rada města Pelhřimova zrušila na svém červnovém zasedání zřízení přípravné třídy

DENISA ŠIMANOVÁ

Pelhřimov – Základní škola Osvobození v Pelhřimově letos přípravnou třídu neotevře. O jejím zrušení rozhodla na svém červnovém zasedání rada města.

Důvodem byl malý počet přihlášených žáků. „Rada města Pelhřimova na svém zasedání dne 22. června rozhodla o zrušení zřízení přípravné třídy pro školní rok 2016/2017 v Základní škole Pelhřimov, Osvobození z důvodu nesplnění zákonných podmínek ke zřízení této třídy," odůvodnila zrušení přípravné třídy mluvčí pelhřimovské radnice Alexandra Knapová.

Jak ale řekla ředitelka tamní Základní školy Osvobození Vladimíra Madronová, dostatečný počet přihlášených dětí v přípravné třídě měli. „Maminka jednoho přihlášeného žáka poslala takový neoficiální e-mail, kde přihlášku do přípravné třídy svého syna zrušila. Tato zpráva se pak nejspíš dostala i na úřad," řekla Vladimíra Madronová.

ZMĚNA ROZHODNUTÍ

V pondělí 20. června ředitelka uspořádala pro rodiče informační schůzku ohledně přípravné třídy, kam přišla i maminka, která zmíněný e-mail poslala. „Maminka si to nakonec rozmyslela a na pondělní schůzce své rozhodnutí změnila," dodala Madronová s tím, že všech deset přihlášek zaslala s předstihem na Městský úřad v Pelhřimově.

Dodala také, že na stole měla navíc ještě jednu přihlášku, u které čekala s podpisem na rodiče. Přihlášku pak ještě před středeční zasedáním rady poslala na úřad. „Já jsem ani nevěděla, že přípravnou třídu budou na zasedání rady projednávat. V programu o tom totiž nic nepsali. Kdybych to věděla, poslala bych ji už v úterý," přemítala.

V době zmíněného zasedání, podle slov pelhřimovského místostarosty Josefa Kocha, měli členové rady k dispozici pouze devět přihlášek. „Pokud bychom měli den po zasedání či den před ním deset přihlášek, je to jedno. Rozhodující je, kolik jich máme přímo v den zasedání rady. Zřízení přípravné třídy ale určitě podpoříme i příští rok," informoval Josef Koch.

dokončení na straně 2

PELHŘIMOVSKO

Želiv opět rozezní Musica Figurata

Želiv – V létě proběhne osm koncertů duchovní hudby. Festival navštíví světoznámý soubor Musica Florea....3

POČASÍ
DEN 15/19
BIO 1
více uvnitř

9 771802 086028 16164

Výstava mapuje rozvoj velké firmy

Žďár nad Sázavou – Historii strojírenského podniku Žďas ve Žďáře nad Sázavou mapuje nová výstava, nazvaná Cesta k pramenům. Zájemci ji najdou ve žďárském regionálním muzeu, kde se včera uskutečnila slavnostní vernisáž.

Strojírenský a metalurgický podnik Žďas patří k největším zaměstnavatelům na Vysočině. Výstava mapuje dobu jeho vzniku a rozvoje od roku 1951, kdy zahájil výrobu. Strojírenský kolos přinesl do dříve agrárního kraje práci tisícům nových dělníků. Lidé do žďárského podniku přicházeli z nejrůznějších míst Československa a vedle pracovních zkušeností si s sebou přinášeli i vůli začít v malém městě svůj nový život. Vzájemná vazba mezi podnikem a městem se za roky existence firmy stala realitou. „Dnes je Žďár nad Sázavou průmyslovým centrem svého okresu. Byly roky, kdy slova Žďárák a Žďasák měla takřka stejný význam," říká marketingový pracovník Žďasu Stanislav Růžička. Výstava v Regionálním muzeu města Žďáru nad Sázavou potrvá do 4. září. (sva)

HISTORIE. Nová výstava mapuje život ve žďárské firmě Žďas. Strojírenský kolos zahájil výrobu v roce 1951.
Foto: ČTK/ Luboš Pavlíček

VÝLET:

V Jemnici otevřeli tajemné podzemí

Jemnice – S letními prázdninami se otevřelo tajemné podzemí starobylého kostela sv. Víta v Jemnici na Třebíčsku. Tamní prostory odkazují nejspíš až na 15. století, kdy v místech působili františkánští mniši. Kostel byl poté přestavěn v pozdně gotickém stylu a stejně jako do jeho podzemí lze během léta nahlédnout i do jeho interiéru. Návštěvníkům se kostel i s podzemím otevírá každý den kromě pondělí od 9 do 12 a odpoledne od 13 do 16 hodin. (kč)

VZDĚLÁVÁNÍ:

Youtubeři lákají ke studiu techniky

Přibyslav – Společnost ACO Industries Přibyslav zahájila projekt, který má přivést mladé lidi ke studiu technických oborů. Využívá přitom internetu a sociálních sítí. V první fázi oslovila trojici mladých youtuberů, kteří prostřednictvím svých zveřejněných videí motivují absolventy základních a středních škol k dalšímu studiu techniky. Další budou následovat. Firma, která na Vysočině zaměstnává na 700 lidí, technické kádry potřebuje. (lz)

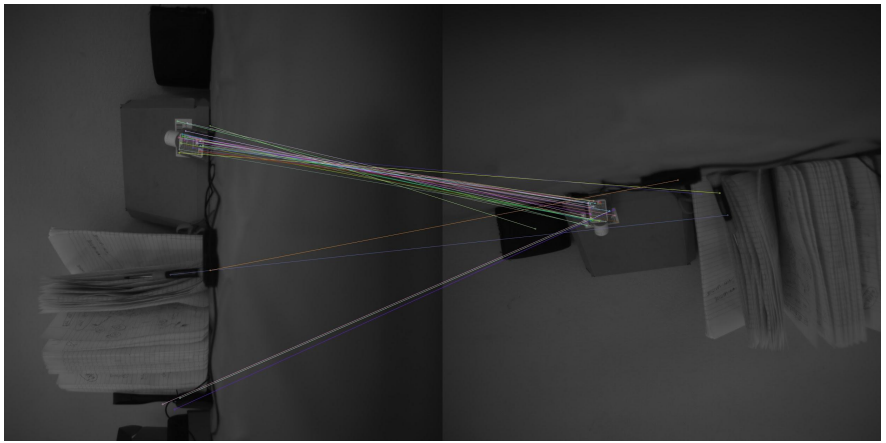
VÝSTAVA:

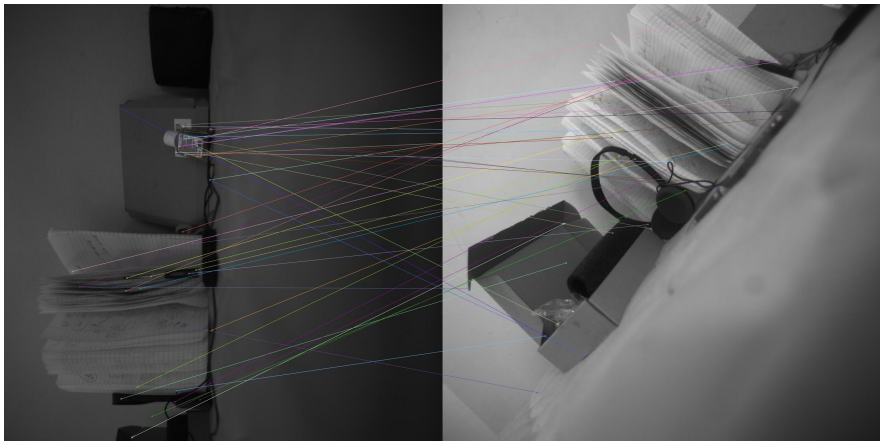
Překrásné lilie zavoní v Žirovnici

Žirovnice – Během nadcházejícího víkendu bude zámecký areál v Žirovnici hostit největší výstavu lilií v republice. Výstava, jež nese název Zahrada Vysočiny, ozdobí zámecké interiéry květinami nejen od tuzemských, ale i slovenských a polských pěstitelů. Na výstavu mohou zájemci zavítat mezi 9. a 16. hodinou. Zítřejší i nedělní program navíc od 15.00 doplní pěvecké a taneční vystoupení dětského folklorního souboru Javorníček z Brna. (dš)

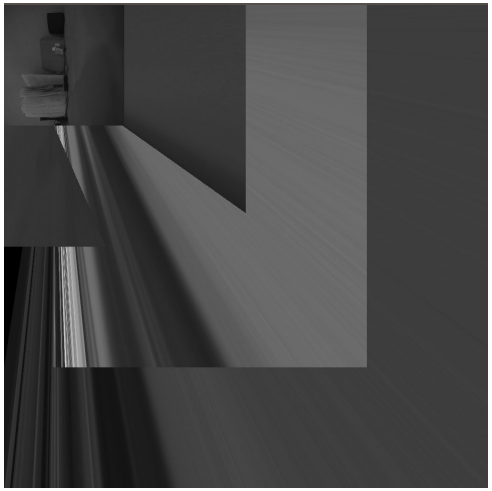
Problem identified so far

1. Key point matching sometime fails.
2. Time to control camera lighting more tightly
    a. Auto Exposure is too dim (Haven't tried out histogram equalisation techniques)
    b. Manual exposure is not adaptive to my room lighting changes
3. All these should be sorted out next week





Tried Image stitching by acquiring camera images- but fails - Debugging in progress



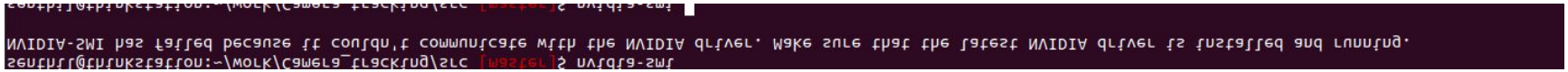
Updates on previous week question

1. Can we reduce the frame rate? The present system doesn't have a notion of frame rate. It's an asynchronous capture. We pass a software trigger to each camera and acquire images from the frame grabber buffer.
2. In short, we cannot acquire more than what we can afford to process
3. The issue of the need to have a synchronous capture will be taken up as and when it becomes necessary

Driver setup for GPU

Went through the CUDA driver installations instructions and installed CUDA drivers

Nvidia CUDA driver is not detected even after installation. Have to explore more



Plan for next week

1. Complete image stitching pipeline - 10h
2. Mount for camera from Onshape - 10h
3. Try out Multi View reconstruction on online datasets -10 h (Just get started)

Quarter Plan

Week No	Proposed Plan
5	Image stitching, Mount for the camera in Onshape, Install GPU drivers and test GPU usage for ubuntu Some trial on Deep Pose prediction
6	Multiview reconstruction on online dataset. Design a 80/20 mount for both image stitching and multiview reconstruction. Network setup and training infrastructure for deeplab cut.
7	MultiView reconstruction on online dataset. Look into Camera extrinsic and intrinsic calibration required
8	Multiview reconstruction from Camera. A look into 3D pose prediction networks. Integration of debugging tools.
9	Multiview reconstruction from Camera. Look into camera synchronisation issues
10	Integrate and test the system the camera system

1. Complete image stitching pipeline - 10h
2. Mount for camera from Onshape - 10h
3. Try out Multi View reconstruction on online datasets -10 h (Just get started)





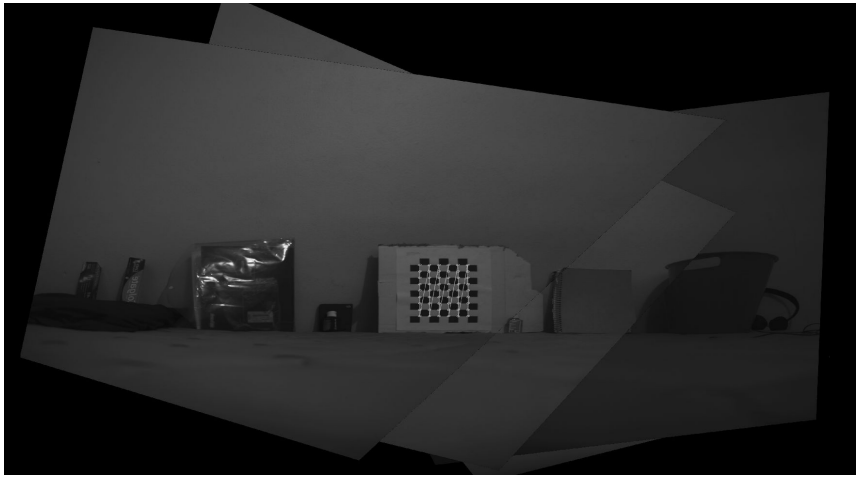



Image stitching video results

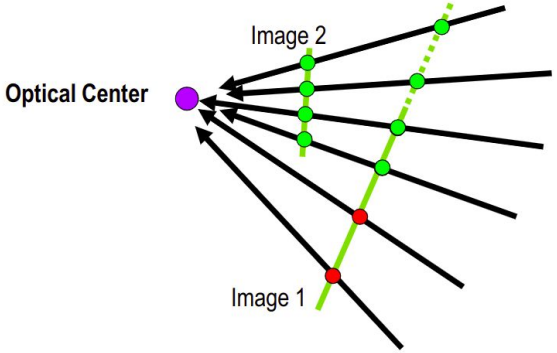
Timing analysis

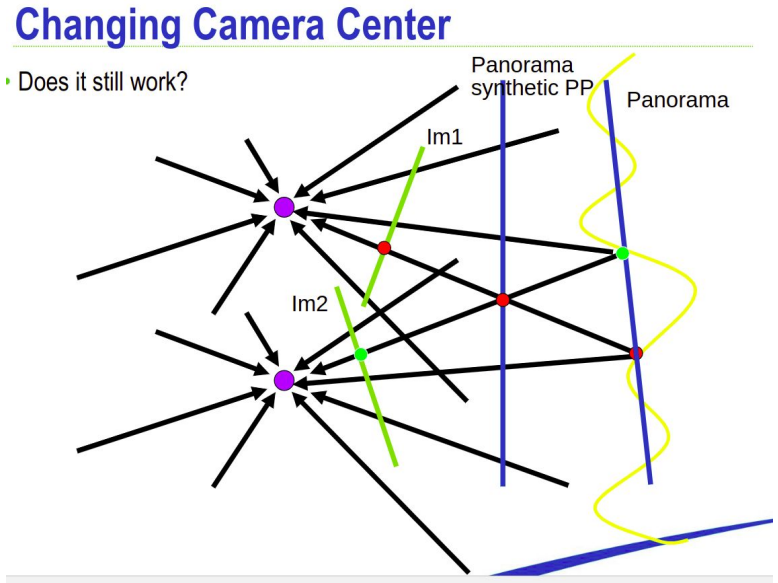
Image Acquisition time - around 200 ms

Image stitching time - Around 1.5 S

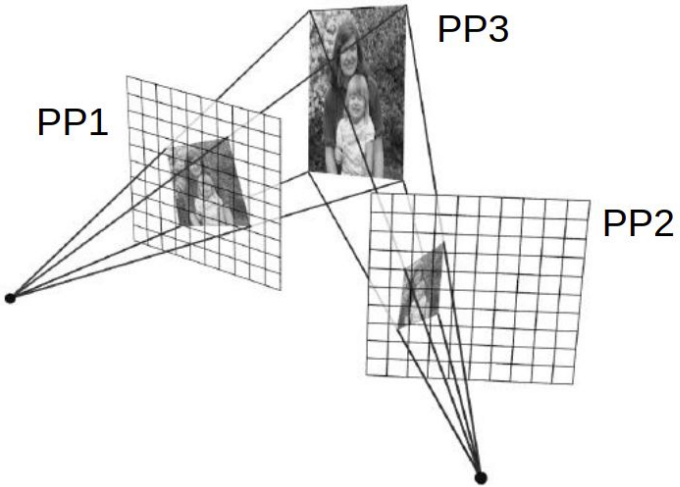
Should be able to bring this down to 0.6 Second. Below that would be difficult for this image size

Multi View image stitching is an ill-posed problem when compared to Panorama stitching - Projective ambiguity



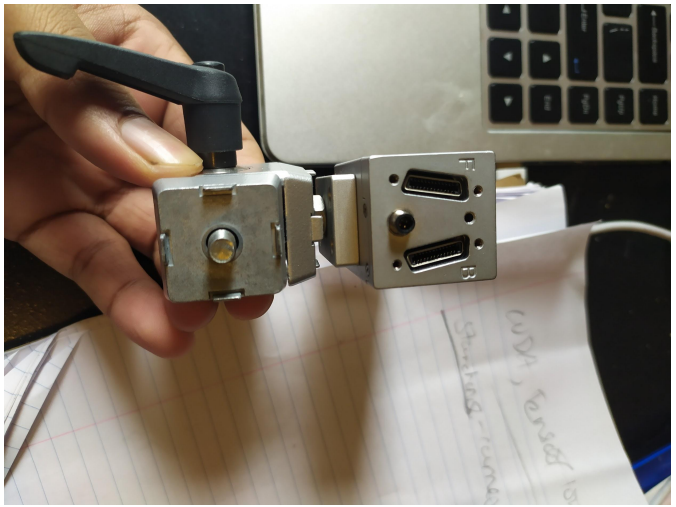


Simplification -The scene is almost planar from a far away view. Example: Aerial image stitching



Camera mounting

1. Can we order a few T-slots? Just to speed up things



GPU Setup progress

1. Conflicts with Packages
    a. Deeplab cut runs on tensorflow 1.8
    b. Tensorflow 1.8 runs with CUDA 9.x
    c. CUDA 9.x doesn’t seem to be well supported on Ubuntu 18.
    d. Spending some time to figure out if CUDA 10.x is backwards compatible (Seems like not)
2. In the worst case, we could use a docker.

Plan next week

1. Multiview Reconstruction Get a basic version working- 15h
2. Synchronous image capture - 8h
3. Clean up Image stitching code - 2h
4. Look more into mounting options - 5h

Quarter Plan

Week No	Proposed Plan
5	Image stitching, Mount for the camera in Onshape, Install GPU drivers and test GPU usage for ubuntu Some trial on Deep Pose prediction
6	Multiview reconstruction on online dataset. Design a 80/20 mount for both image stitching and multiview reconstruction. Network setup and training infrastructure for deeplab cut.
7	MultiView reconstruction on online dataset. Look into Camera extrinsic and intrinsic calibration required
8	Multiview reconstruction from Camera. A look into 3D pose prediction networks. Integration of debugging tools.
9	Multiview reconstruction from Camera. Look into camera synchronisation issues
10	Integrate and test the system the camera system

Proposed plan

1. Multiview Reconstruction Get a basic version working- 15h
2. Synchronous image capture - 8h
3. Clean up Image stitching code - 2h
4. Look more into mounting options - 5h

Executed Plan

1. Multiview Reconstruction Get a basic version working- 15h
2. Synchronous image capture - 8h
3. Clean up Image stitching code - 2h
4. ~~Look more into mounting options - 5h~~

Had health issues and was forced to rest 24 * 7 for 2.5 days. From there on, it was a catch up with my 3 courses and hence, I was able to spend only 6-7 hrs of project time this week

Continuous capture

1. Continuous capture runs in 30 ms (for all 4 images put together)
2. Note that this is not equivalent to 1000 / 7 = 143 frames per second.(Infact , for this experiment the camera was running at only 71 fps)
3. The camera can run at a max frame rate of 187 fps, the PIXCI E8 framegrabber can support frames rates upto 400 fps. So the theoretical max frame rate is 187 fps

Synchronous capture

1. No documented references in PIXCI library
2. What is this ? Hardware or Software?
3. Still Synchronous captures is a necessity:
   a. Assume 70 fps, rodent speed: 3.5 m / sec.
   b. The maximum offset between 2 cameras can be 14.2 ms. The maximum movement disparity is 0.49 cm or 4.9 mm.

Quote Date	Ship Via	F.O.B.	Customer PO Number	Payment Method
04/23/14	Pick Up	EX WORKS: Buffalo Grove		NET30
Entered By	**Salesperson**	**Ordered By**		**Resale Number**
Charlie	Charlie Dijak			E9990-4055-06

Order Quantity	Approve Quantity	Tax	Item Number / Description	Unit Price	Extended Price
4	4	Y	LENS	500.00	2,000.0
			LENS - To Be Determined	<8.00 %>	-160.0
			Warehouse: MAIN		
4	4	Y	CAMERA	2,226.00	8,904.0
			BASLER acA2040-180km 2048 x 2048 @ 187 fps Mono	<8.00 %>	-712.3
			Warehouse: MAIN		
4	4	Y	TRIPOD MOUNT	39.00	156.0
			BASLER TRIPOD MOUNT PLATE w 1/4-20 SOCKET	<8.00 %>	-12.4
			Warehouse: MAIN		
8	8	Y	CBL-MCL-5M U of M : Each	135.00	1,080.0
			CameraLink cable, MiniCL(SDR) & CL(MDR) conn, 5 meter	<8.00 %>	-86.4
			Warehouse: MAIN		
4	4	Y	PIXCI-E8 U of M : Each	1,295.00	5,180.0
			PIXCI E8 PCIe Board	<8.00 %>	-414.4
			Warehouse: MAIN		
1	1	Y	SYNCHRONIZATION	250.00	250.0
			4 CAMERA SYNCHRONIZATION	<8.00 %>	20.0
			Warehouse: MAIN		
1	1	Y	XCAP-STD-V3.8 U of M : Each	1,495.00	1,495.0
			XCAP STANDARD V3.8, SINGLE USE	<8.00 %>	-119.6
			Warehouse: MAIN		

Print Date	04/23/14
Print Time	09:44:19 AM
Page No.	1

Printed By: Charlie

Continued on Next Page

EPIX IMAGING PRODUCTS ARE MANUFACTURED IN THE UNITED STATES OF AMERICA

Plan next week

1. See if stitching time can be reduced (from 600 ms) - 8 h
    a. Time profile code
    b. OpenCV build options - Its already in release mode. Code is still in debug mode.
    c. Parallel processing / Pipeline architecture
2. Get started on MultiView reconstruction - 10h
3. Synchronous capture - 7h
4. CUDA GPU setup - 3h

We must have been using a hardware based trigger

Izaak -- Question about triggering the 4 synchronized cameras. Do you plan to use a signal generator to control the frame rate of the cameras? If you do, then what type of signal does your generator provide? Our frame grabbers are designed to accept an LVTTL (Low voltage [3.3 volts] trigger pulse. If you are using some other type of signal then we need to know.

Such an external signal is not required. The frame rate and sequence duration can be set in software. Once the frame rate and sequence duration are set, then the system will sit and wait for a single "start sequence capture" trigger pulse. We will provide a pushbutton on a cable that will trigger all 4 frame grabbers / cameras at the exact same time. We typically provide this pushbutton on a 2 meter cable. Typically the person responsible for triggering sequence capture is sitting at the computer keyboard and so a longer cable isn't needed.
BUT -- if there is an advantage, in your application, to having a longer cable -- 3 to 6 meters -- then let me know. We can provide a longer length cable. For example, you can have the cameras armed and ready to capture -- completing the setup at the keyboard -- but then maybe you want the robot to be in a certain position before you start capturing images. Maybe this position isn't easily observable when sitting in front of the computer. This pushbutton gives you the freedom to get out of the chair and move around the tank -- depending on how long we make the cable. The cable length won't change the system price -- so let me know if you want a longer cable.


MIC IN
LINE OUT
LINE IN
TRIGGER
B4
F4
BASE 0
FULL 0

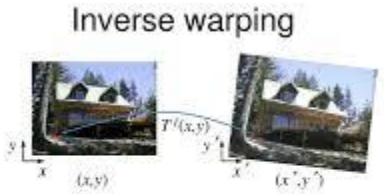
Image stitching time down to 180 ms

1. 1.5 S to 600 ms (Found an algorithmic optimisation)
    a. Previous strategy:
        i. Calibration strategy: The output image size if unknown. So use a very big canvas. Find homography relating each image with the canvas Finding the bounding box for image inside the canvas.
        ii. Online stitching: Use the estimated homography to warp each image onto the “Big canvas”. Then use the bounding box to crop the image from the canvas.
        iii. Modified strategy: Modify the Homography so that we know how to transform from the original image to the cropped image.
2. Switched to Release mode: 600 ms to 250 ms
3. Removed unwanted debug Read / Write File operations: 250 ms to 180 ms

Scope for further improvement - Custom warp function

1. Do inverse warping for each image and transfer pixels from the warped image to the final canvas

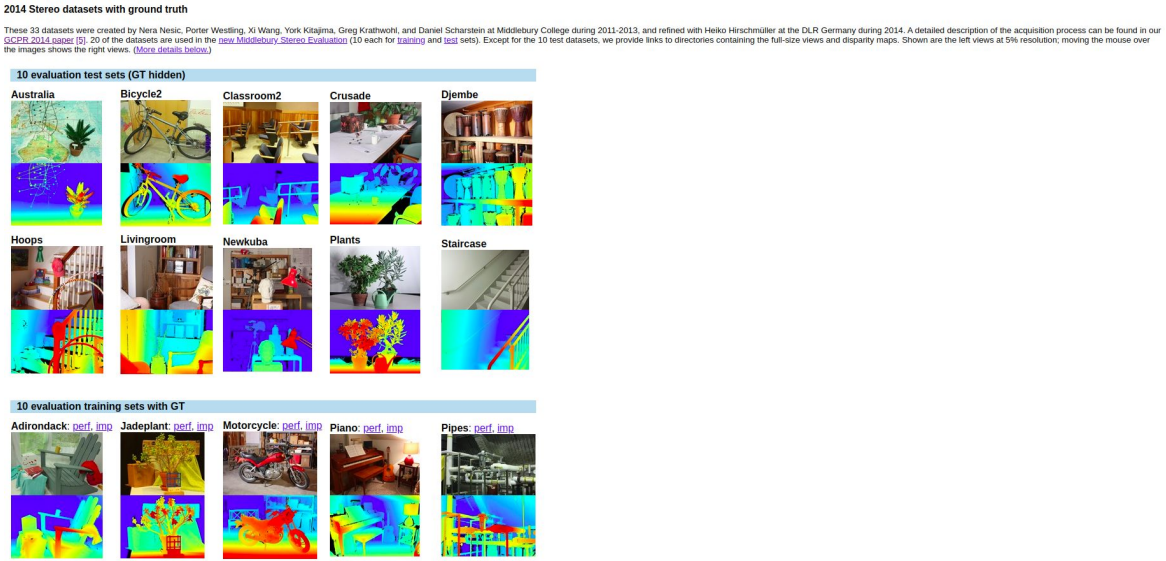
Idea: Write a custom warp function that directly uses all homographies and predicts the values of pixels using inverse warp



Expected result: It is definitely algorithmically faster but not sure about the end gain since OpenCV uses a lot of inbuilt SIMD instructions to speed, which we may lose by writing a custom function

MultView Reconstruction

1. Dataset exploration
2. Got started with basic implementation but no showcasable progress



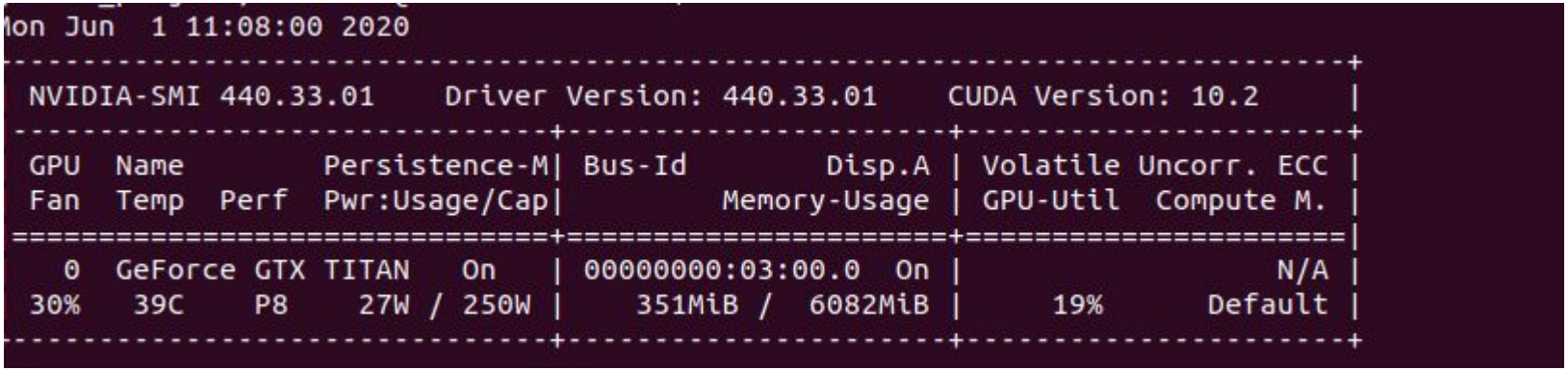
2014 Stereo datasets with ground truth

These 33 datasets were created by Nera Nesic, Porter Westling, Xi Wang, York Kitajima, Greg Krathwohl, and Daniel Scharstein at Middlebury College during 2011-2013, and refined with Heiko Hirschmüller at the DLR Germany during 2014. A detailed description of the acquisition process can be found in our GCPR 2014 paper [5]. 20 of the datasets are used in the new Middlebury Stereo Evaluation (10 each for training and test sets). Except for the 10 test datasets, we provide links to directories containing the full-size views and disparity maps. Shown are the left views at 5% resolution; moving the mouse over the images shows the right views. (More details below.)

10 evaluation test sets (GT hidden)

10 evaluation training sets with GT

Tested CUDA installation by running a small nueral network



Plan for next week

1. Continue with MultiView Stereo - 10h
2. Check OpenCV build options for image stitching - 1h

Quarter Plan

Week No	Proposed Plan
5	Image stitching, Mount for the camera in Onshape, Install GPU drivers and test GPU usage for ubuntu Some trial on Deep Pose prediction
6	Multiview reconstruction on online dataset. Design a 80/20 mount for both image stitching and multiview reconstruction. Network setup and training infrastructure for deeplab cut.
7	MultiView reconstruction on online dataset. Look into Camera extrinsic and intrinsic calibration required
8	Multiview reconstruction from Camera. A look into 3D pose prediction networks. Integration of debugging tools.
9	Multiview reconstruction from Camera. Look into camera synchronisation issues
10	Integrate and test the system the camera system

Explored and read on parallel processing options with OpenCV

TBB, IPP, CUDA, openMP - parallel architecture.

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D WITH_TBB=ON -D
WITH_OPENMP=ON -D WITH_IPP=ON -D BUILD_EXAMPLES=OFF -D
WITH_NVCUVID=ON D WITH_CUDA=ON D BUILD_DOCS=OFF -D
BUILD_PERF_TESTS=OFF -D BUILD_TESTS=OFF -D WITH_CSTRIPES=ON -D
WITH_OPENCL=ON      -D CMAKE_INSTALL_PREFIX=/usr/local  -D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules     -D
OPENCV_ENABLE_NONFREE=ON ..
```

Parallel processing to speed up

1. Used pthreads and opencv parallel for to parallelise computations.
2. Final time after optimised OpenCV build - 130 ms
3. Final time after parallel processing - 80 ms

Overview of all optimisation for imagestitching speed up

1. Original time - 1.6S
    a. Algorithmic optimisation (Modifying homography to use the appropriate canvas) - 0.6 S
    b. Using right build flag (Release mode and disabling all debug flags) - 250 ms
    c. Disabled all file, read and write operations and debug result processing - 180 ms
    d. Using optimised OpenCV build - 130 ms
    e. Using parallel processing - 80 ms

Quarter-1 Progress summary

1. Studied the whole system and understood the whole system (Pieced together information from presentations, mails, system files, interactions through lab meetings, customer care)
2. Wrote code for driver interface for using Basler cameras
3. Constructed image stitching system and optimised it to run under 80 ms.

Quarter -2 - high level goals

1. Deploy the image stitching system in lab.
2. Fine tune synchronous image capture.
3. Use deep network to track the trajectory of the rodent (Possibly deep labcut)
4. Deploy the system in lab and test.

Lab access needed for:-

1. Deploying and testing image stitching system in lab
2. Synchronous capture
3. Data collection for rodent for our use case
4. Integration testing

Offline work:-

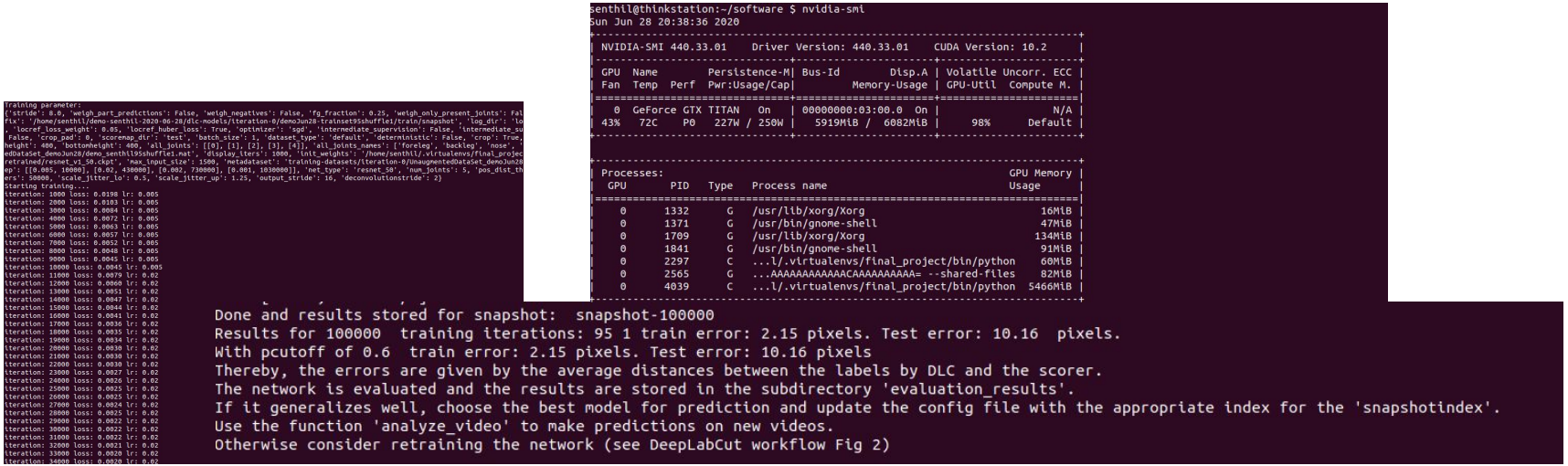
1. Deep network training and pipeline construction
2. Synchronous capture (May be possible depending on the hardware used)

Weekly Progress

1. Deeplab cut trains on system (Trained a model for mouse pose tracking)
2. Stereo reconstruction from two views

Deep lab cut

1. Sorted all dependencies and finally got it perfectly working
   a. Cuda version compatible with Deeplabcut and OpenCV (10.0)
   b. Tensorflow version (1.14), CuDNN (7.7), Nvidia driver (440), WxPython (4.0.7.post2)
2. Trained a network on the example videos
3. Comparing and contrasting other approaches: Deep learning based object tracking, classical vision based object tracking.



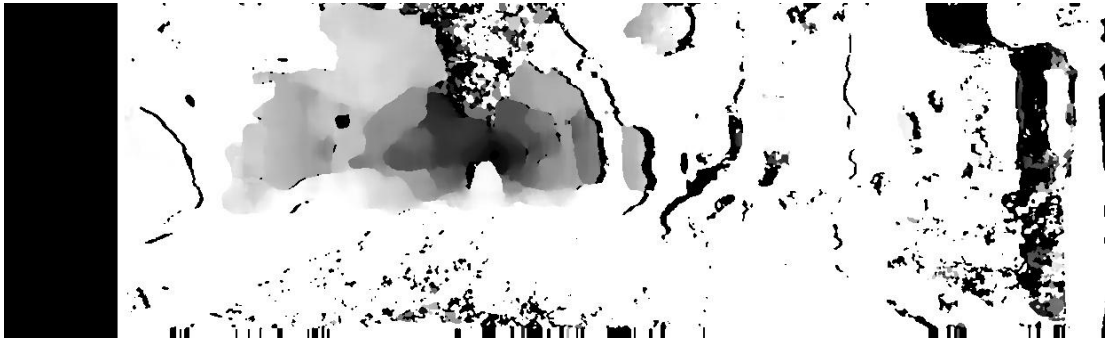
Two routes

1. The Deep learning route (DeepLab Cut or Any object detection network)
    a. Train a model
    b. If time is too much, reduce model size
    c. Quantise models
    d. Try more efficient architectures
2. The classical vision route (Faster but may not be as robust)
    a. Have costly object detection route
    b. Have a cheap tracking step (in a few ms)

Trained Networks on a video close to our use case

1. The video may be subject to copyright
2. It can in fact be an illegal use (for now).

Disparity estimation from two views



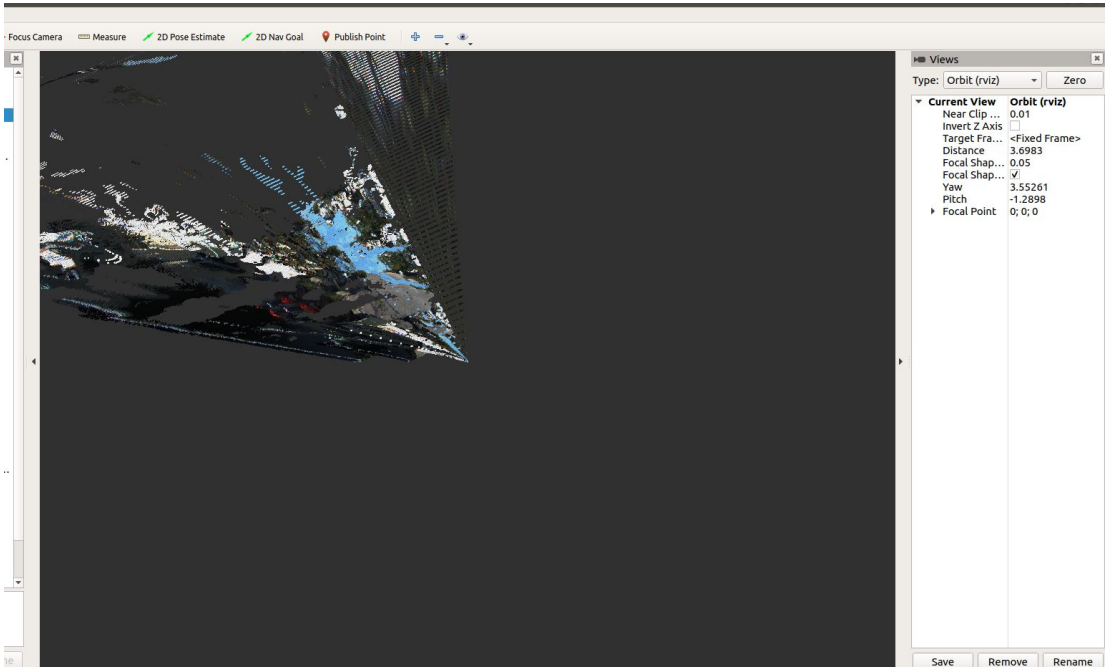
Problems with the system

1. The system hangs / freezes during operation.
2. The specs are powerful enough to ensure that we shouldn’t have this.
3. This happens in both OS (Ubuntu and Windows)
4. This should be a hardware issue (loose components like RAM / overheating etc..)

Plan for next week

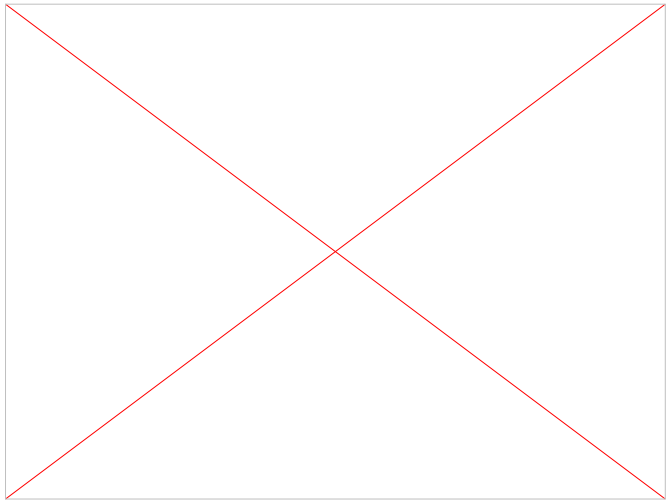
1. Figure out the hardware issue - 5h
2. Get Deep lab cut running on the sample video and draw the position tracking over time - 10h
    a. Benchmark timings of inference
3. Refine stereo reconstruction and Visualise point clouds - 5h

Progress Made - Point cloud visualisation



System Check

1. Temperature of all components is under control
   a. GPU - Around 57 to 60 C (Specs say it can last as high as 100 C)
   b. CPU - Around 50 C (Can last as high as 60 C)
2. RAM is good. Did memory test
3. GPU card is good.
4. Motherboard - Hard to do software tests
5. CPU - Hard to do software tests
6. Hard Disk - Good



Planning to switch everything about Deep learning training to MSR system

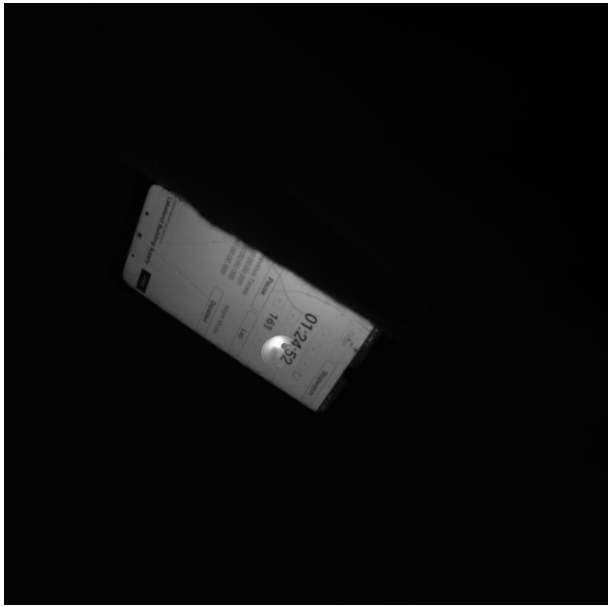
Synchronisation

1. Went through basic tutorials on Synchronisation

Plan for next week

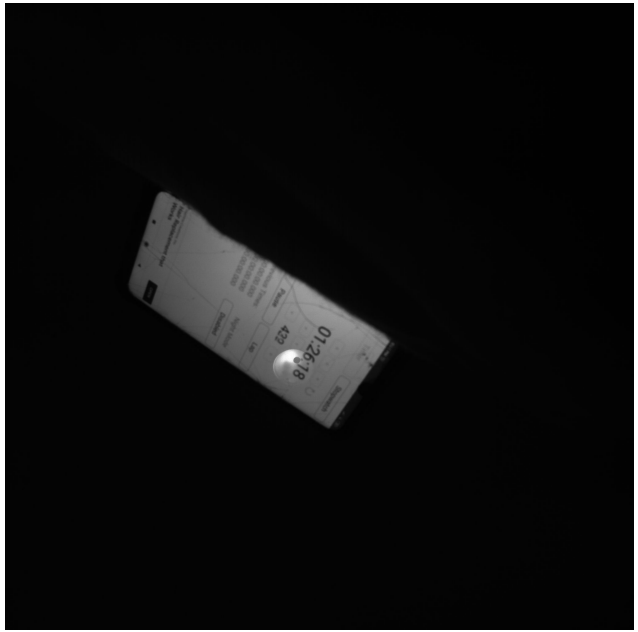
1. Read up more on camera synchronisation - 5h
2. See if there is a way to quantify amount of dis synchronisation between camera and validate if this is too low or high for our use case - 10h
3. Set up all training infrastructure in MSR system.

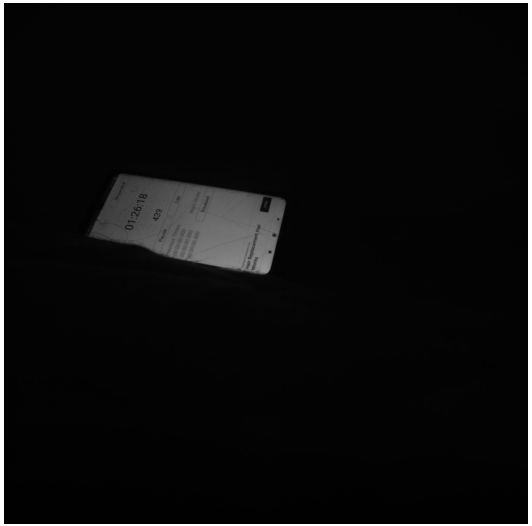
Camera synchronisation progress

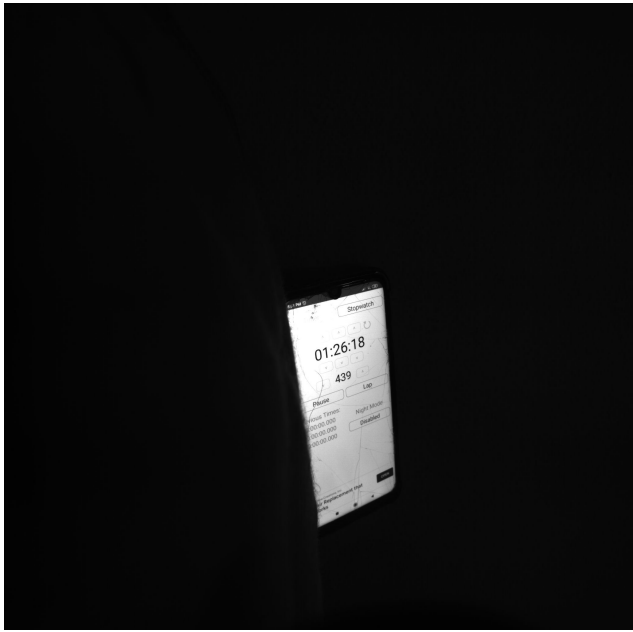


Capture times - 161, 162, 176

Max diff - 15 ms







Capture times - 432, 439, 439
Max diff - 7 ms

What's happening?

Our first attempt - Sequential wait and capture (dosnap)

Flow - Start capture 1, wait until capture 1 ends, start capture 2, wait until capture 2 ends etc... (heavily dependent on integration time, which depends on lighting etc..)

Our second attempt - Sequentially initiate all captures and collect all the images later.

Flow - Start capture 1, start capture 2, start capture 3, start capture 4, get image 1 , get image 2, get image3, get image 4 etc...

Can there be any delay between different image capture start times? Yes

How bad can this delay be? (About 10 - 20 ms based on empirical measurements)

Does this mis-synchornisation difference matter?

1. Lets take the worst case to be 20 ms off.
2. Average speed of mice - 3.5 meter per second
3. Distance moved - 3.5 * 0.02 = 7 cm

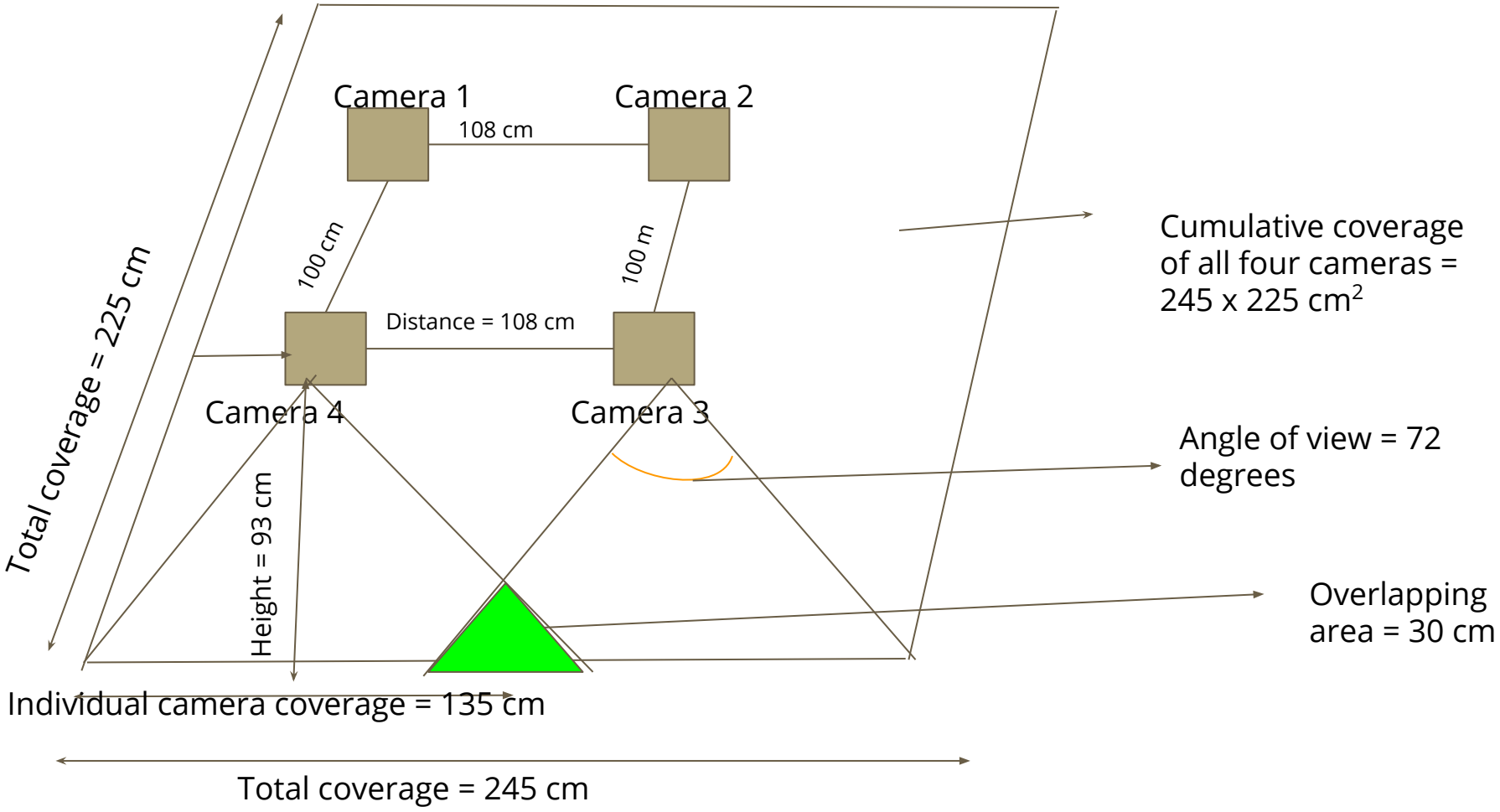
Is this big ? For 2D tracking not a big deal. Worst case - Mice’s position flickers with 7 cm uncertainty when it is switching between different camreas

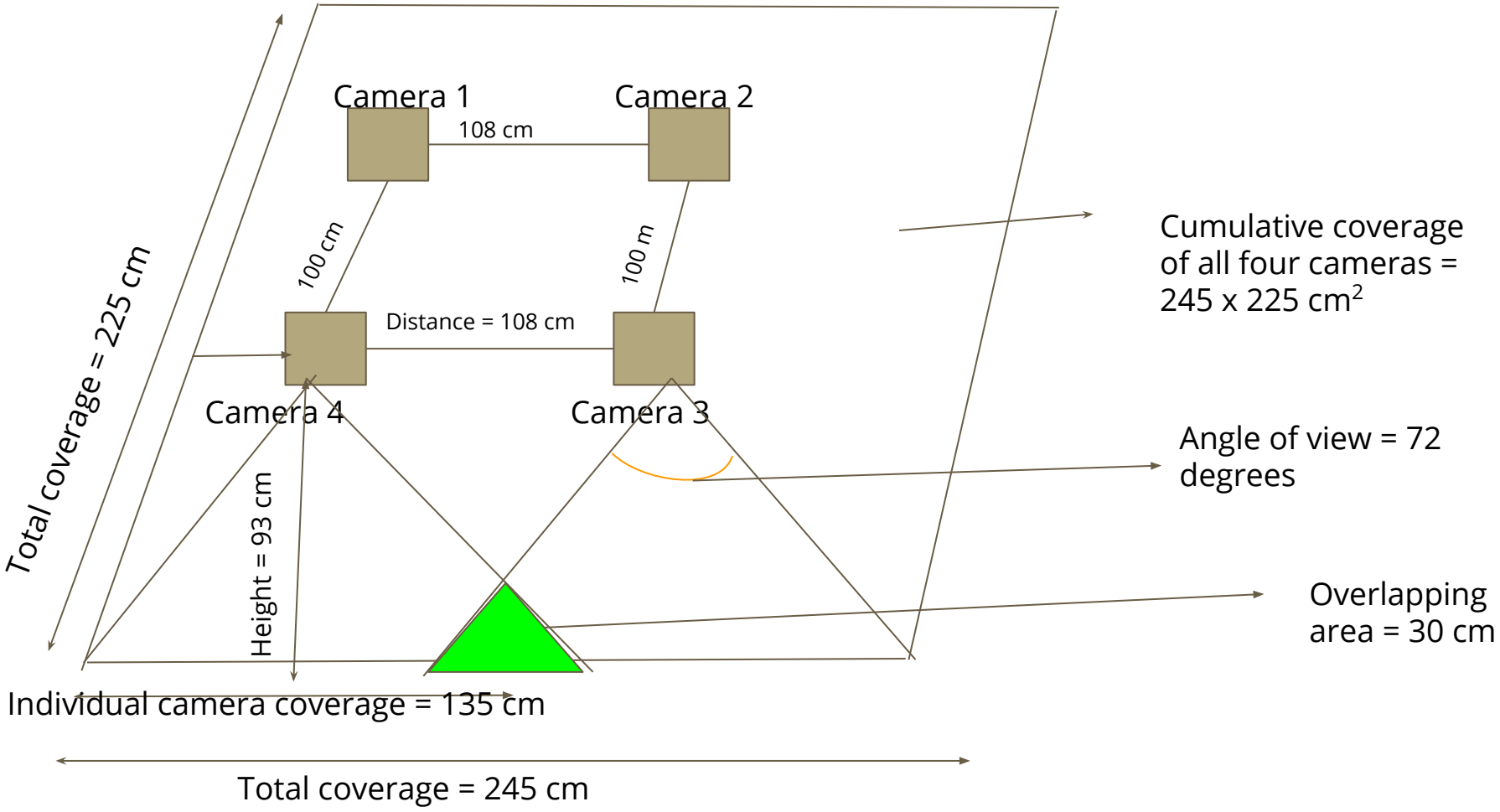
For 3D tracking? Big deal. We will not be able to do reliable triangulation.

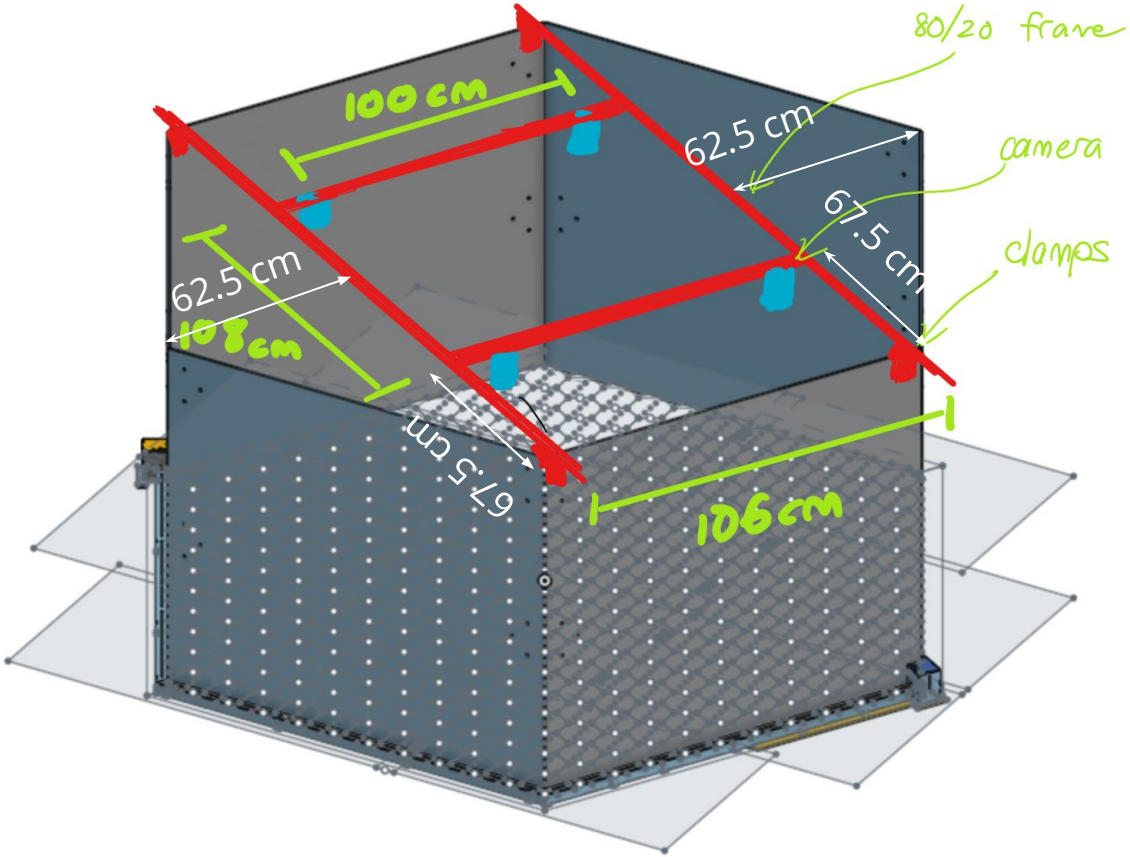
Answer - In the short run no, In the long run yes.

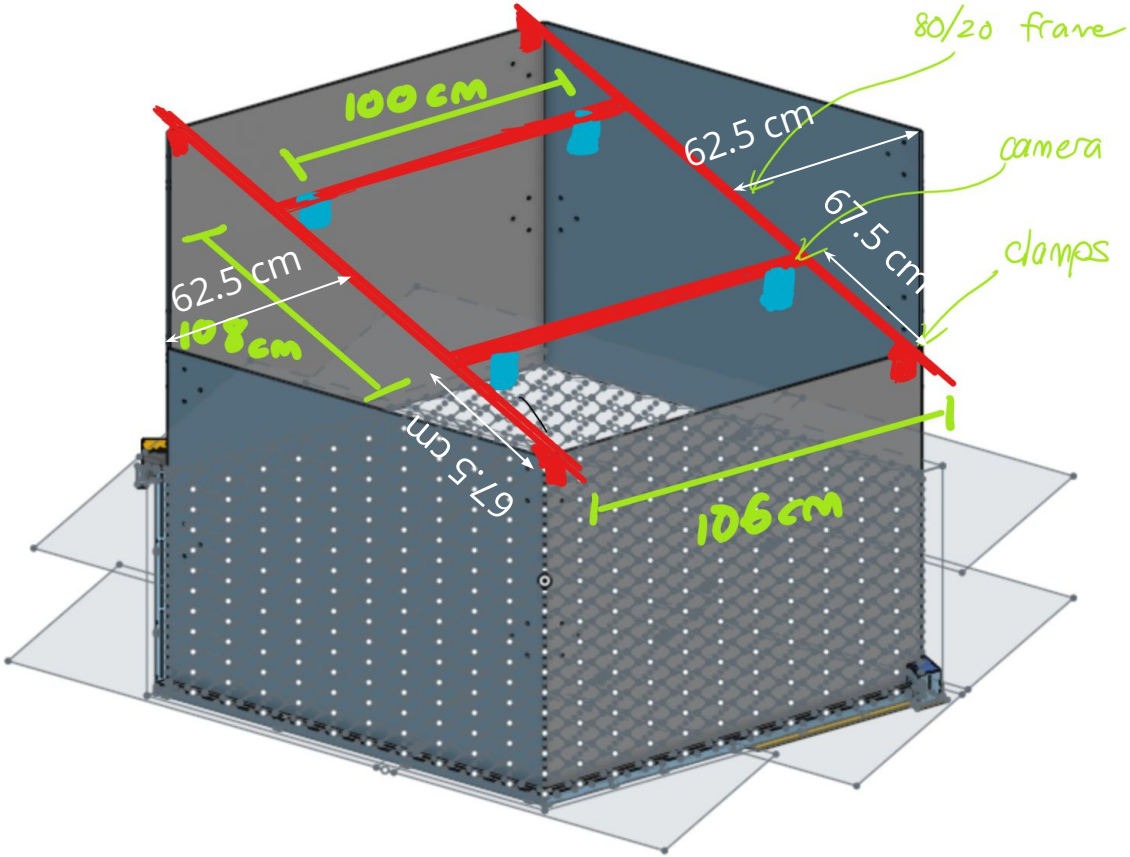
Next week Plan

1. Have deeplab cut based tracking running on a different video
2. Interaction with Pixci about synchronisation??


Camera 1
Camera 2
108 cm
100 cm
100 m
Distance = 108 cm
Camera 4
Camera 3
Total coverage = 225 cm
Height = 93 cm
Individual camera coverage = 135 cm
Total coverage = 245 cm
Cumulative coverage of all four cameras = 245 x 225 cm^2
Angle of view = 72 degrees
Overlapping area = 30 cm

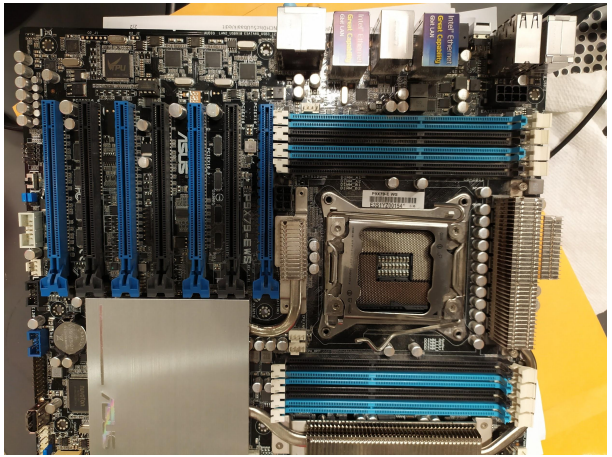

Camera 1
Camera 2
108 cm
100 cm
100 m
Distance = 108 cm
Camera 4
Camera 3
Total coverage = 225 cm
Height = 93 cm
Individual camera coverage = 135 cm
Total coverage = 245 cm
Cumulative coverage of all four cameras = 245 x 225 cm^2
Angle of view = 72 degrees
Overlapping area = 30 cm

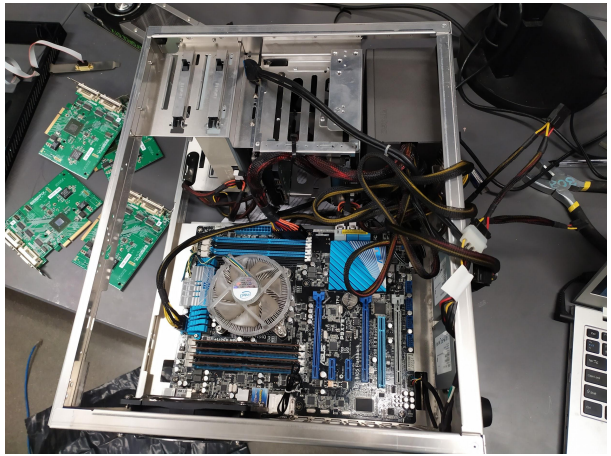

80/20 frame
camera
clamps
100 cm
62.5 cm
67.5 cm
62.5 cm
108 cm
67.5 cm
106 cm

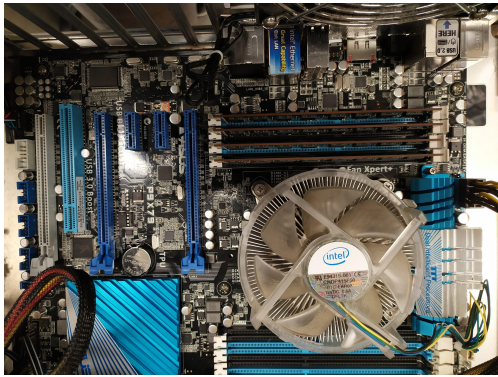

80/20 frame
camera
clamps
100 cm
62.5 cm
67.5 cm
62.5 cm
108 cm
67.5 cm
106 cm

Last week progress

1. Mounted the new motherboard only to realise that it isn’t compatible with what we want







Last week progress

1. C++ inference vs c++ wrapper for python
    a. C++ inference doesn’t give any reasonable speed up since 99% of all deep learning libraries (irrespective of Tensorflow or pytorch) is written in C++. Less than 1 % of the deep learning code actually runs in native python. Therefore the speed is negligible or minimal.
    b. C++ inference poses problem - Not well documented, weird issues reported
    c. C++ wrapper for python inference - pybind11

Progress over summer

1.Deeplab cut training on a random mice video

2.System checks and component ordering

3. Experiment to check the camera synchronisation offset

4. Camera mounting specifications and solution to camera mounts.

5. Some information exploration on inference pipelines

Plan for next week

1. Inference pipeline setup - 20h

Two proposals:-

1. Reassemble the old motherboard, work with the annoying hardware and establish the image stitching system on maze

Explore alternatives:-

1. Detection followed by tracking pipeline setup.
2. Background subtraction based mice tracking

Progress

1. Exploration on embedding python in C++ - 15h
    a. Boost.Python
    b. Pybind11
2. Boilerplate code on a simple example to embed python in C++ using pybind11 - 15h
3. Exploration on object tracking and checking on a sample code of object tracking using C++ - 15h

Vidoes outputs

Plan for next week

1. Setup the inference pipeline - 60h
    a. Image acquisition using camera drivers
    b. Image stitching using openCV
    c. Mice pose detection using python embedding
    d. Object tracking using openCV
    e. Reinitialise mice pose upon detection failure

Progress this week

1. Inference for a single image - 25h
2. System assembling - 7h
3. Exploration on fast object tracking methods - 10h

Inference pipeline

1. Demo
2. Disappointing stat - 1s per image for pose detection. Have to debug deeper on why this is so slow
3.

Assembled PC

1. Yet to check if it works properly

Exploration on fast object tracking methods.

1. MOOSE tracker - (50 - 60 FPS on normal CPU).
2. Yolo V5 (400 FPS)
3. Background subtraction (CUDA implementations can run upto 1000 FPS)

Plan next week

1. Verify if the image capture system works fine
    a. Bring system into the maze and start constructing system
    b. Quicker we get data / video, the quicker we can build models
    c. Quicker we start putting the system together, the quicker we notice problems
    d. Follow up on synchornised capture

Progress this week

1. System works - Was able to train a network on our system. I haven’t seen any instance of system hanging during the past 4-5 days of operation.
2. Speed analysis of different methods.

BackGround Subtraction

Deep Lab cut Inference time

Inference Time - 16-20 ms

FPS - 50-60 FPS

Object Tracking with CUDA verison of OpenCV

BackGround Subtraction GPU version

A proper OpenCV build gives a lot of optimisation

TBB, IPP, CUDA, openMP - parallel architecture.

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D WITH_TBB=ON -D
WITH_OPENMP=ON -D WITH_IPP=ON -D BUILD_EXAMPLES=OFF -D
WITH_NVCUVID=ON D WITH_CUDA=ON D BUILD_DOCS=OFF -D
BUILD_PERF_TESTS=OFF -D BUILD_TESTS=OFF -D WITH_CSTRIPES=ON -D
WITH_OPENCL=ON      -D CMAKE_INSTALL_PREFIX=/usr/local  -D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules     -D
OPENCV_ENABLE_NONFREE=ON ..
```

Comparison

Method	Speed	FPS	Scope for improvement
Background subtraction	1.4 ms - 1.6 ms	600-700	CUDA versions claim to run as high as 1200 FPS. We may be at the bottleneck of our hardware or there is some scope for improvement
Deeplab Cut	16 ms	62	We may be able to fasten it by using MobileNet instead of ResNet architecture.
Object tracking	3-4 ms	250-350	Online estimates say 400 FPS. So, we are almost close
Yolo v4	-		Expected to be 60-70 FPS

Plan for next week

1. Setup the system inside the lab and test stitching and capture system
2. Make progress on the inference pipeline.

Progress this week

1. Code refactoring and whole pipeline setup
2. Re installing drivers for frame grabbers on our system
3. Installing all dependencies on our system

Plan for next week

1. Test Image capture once in our system.
2. Setup remote access in the GPU system.
3. Set up the system in CCM and get image stitching working on the actual system.
4. Follow up with EPIX on synchronised capture.

Progress last week

1. ~~Test Image capture once in our system.~~
2. ~~Setup remote access in the GPU system.~~
3. ~~Set up the system in CCM and get image stitching working on the actual system.~~
4. Follow up with EPIX on synchronised capture.

Other items worked on:-

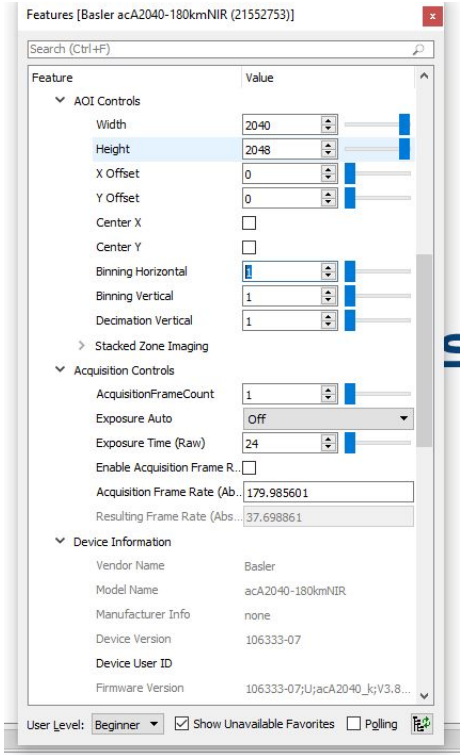
1. Testing the cameras for faster capture through pylon configuration

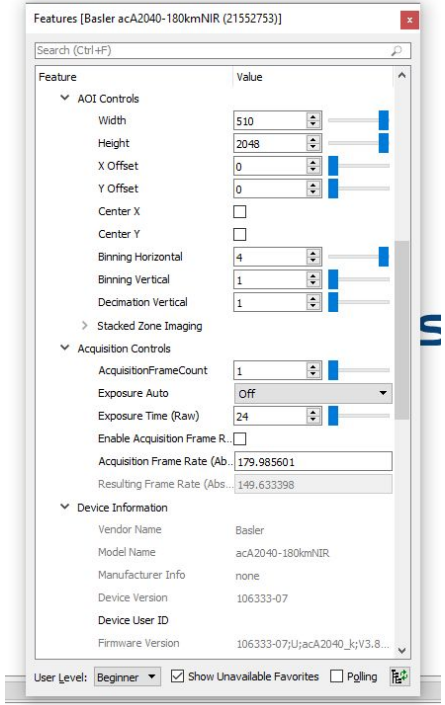
Output for our system setup

Image capture implications

1. I played around with camera configurations. Their spec speed of 180 fps seems hard to reach unless we choose to
    a. Run the camera in binning mode (Binning refers to a process of grouping neighbouring pixels together to obtain a single pixel). This reduces the resolution but increases the frame speed.
2. Presently an object which is roughly the same size as a mice occupies 150 x 150 pixels. With binning this will reduct to 40 x 150.
3. We will have the option to run
    a. At 37 fps for 2048 x 2048 (150 x 150 pixel size for mouse)
    b. At 75 fps for 2048 x 1024 (150 x 75 pixel for mouse)
    c. At 150 fps for 2048 x 512 (150 x 40 for mouse)

Screenshot from pylon





Missing pieces

https://www.edmundoptics.com/p/frac14-20-mounting-adapter-for-basler-ace-usb-30/30775/

Plan for next week

1. Setup a pipeline for image capture sequence (done)
    a. Will Alex / German be okay with image capture while training
    b. Analyse codes / image capture implications
2. Setup the other two cameras and test the whole image stitching system
3. Setup Pylon viewer in Linux (A pain now since its only available in windows)
4. Test trigger and get back to EPIX for synchronisation clarification.

Plan for next week

1. ~~Setup a pipeline for image capture sequence~~
    a. Will Alex / German be okay with image capture while training
    b. Analyse codes / image capture implications
2. ~~Setup the other two cameras and test the whole image stitching system~~
3. ~~Setup Pylon viewer in Linux (A pain now since its only available in windows)~~
4. Test trigger and get back to EPIX for synchronisation clarification. (Sequence capture doesn't work)

Installed all 4 cameras

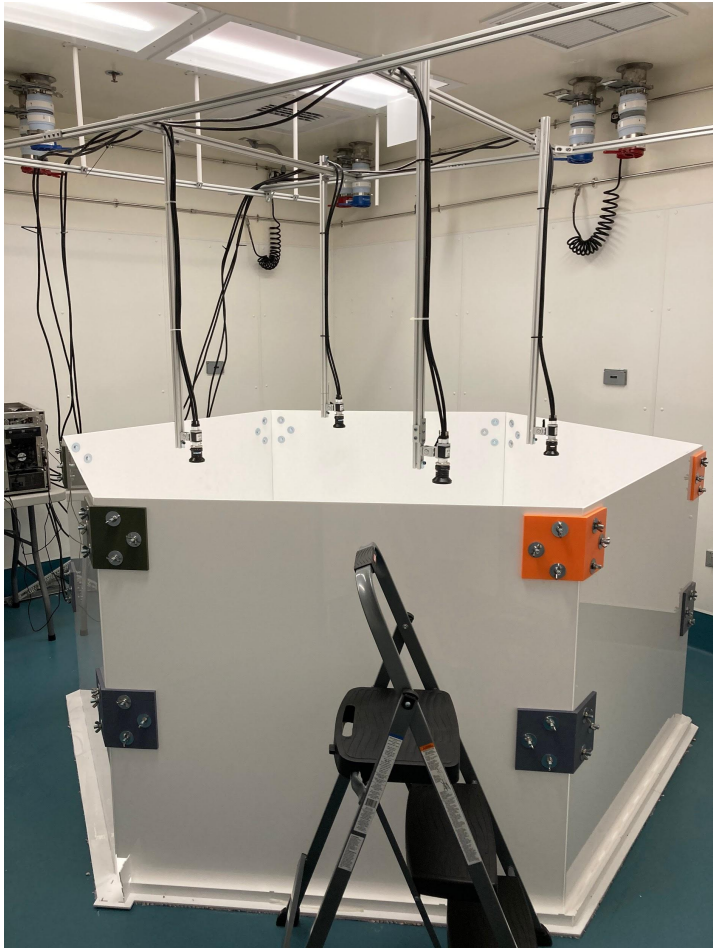


Image stitching output

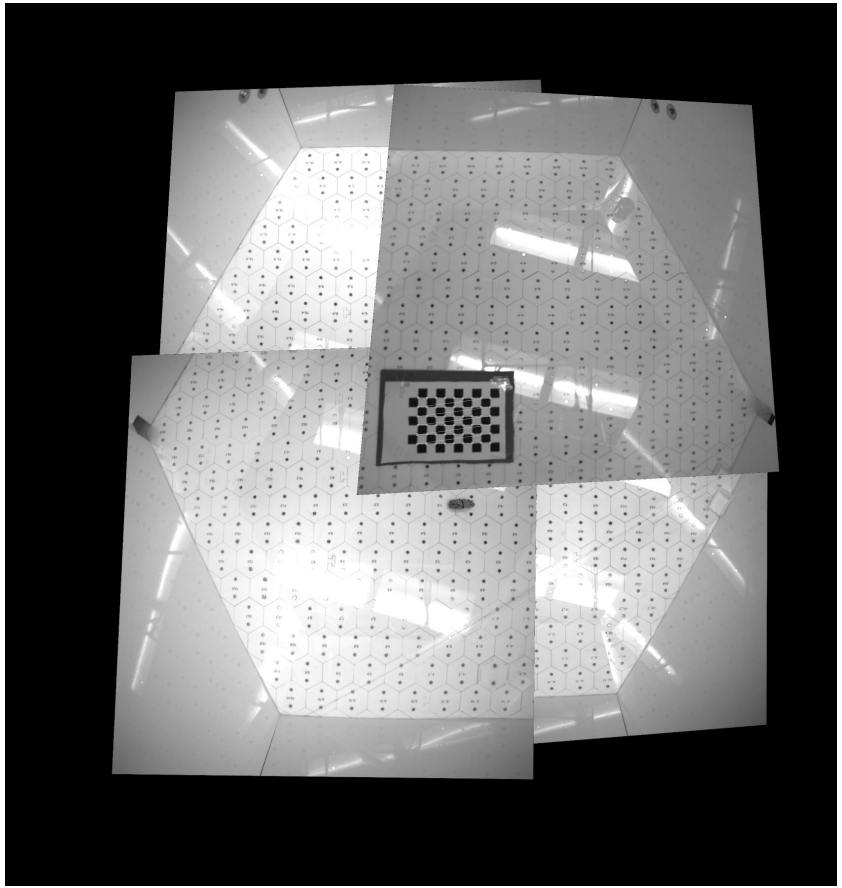


Image read / write

1. Analysed different codes and narrowed down on H.265 video codes
2. Video from 4 camera can now be stored as a video in lossless compression format
3. We can now store images from all cameras and videos

Pylon image viewer - setting

Hello Senthil.

Which Camera Link frame grabber are you using? If using one of our PIXCI frame grabbers, our XCAP software will allow capture of images and video; XCAP software also has custom, camera specific, controls to adjust and configure Basler Ace cameras.

PIXCI frame grabbers and XCAP are supported under x86 and x86-64 systems, as well as under many ARM systems (including nVidia TK!, TX1, TX2, Xavier, and Nano).

Regards
Howard

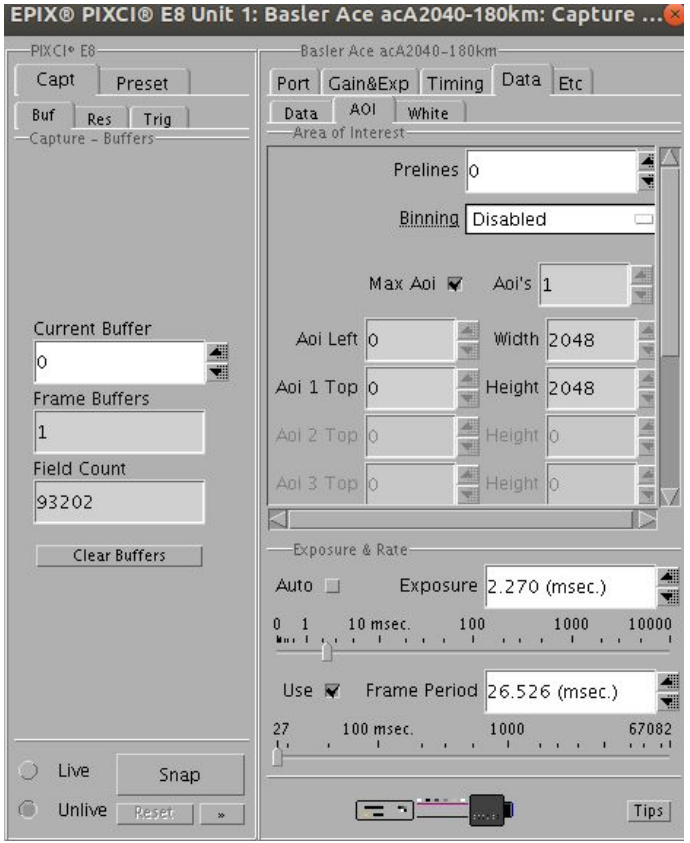
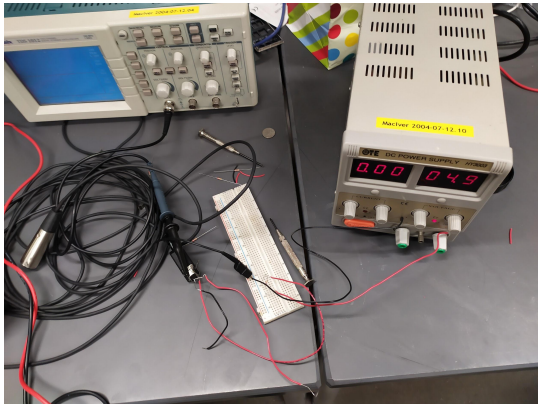


Image sequence capture

1. Trigger not received from snap button.
2. The issue could be due to
    a. Faulty snap button
    b. Problem with the sync buffer cable
    c. Software problem
    d. Problem with the frame grabber board (very less likely)
    e. Problem with the camera (very less likely)



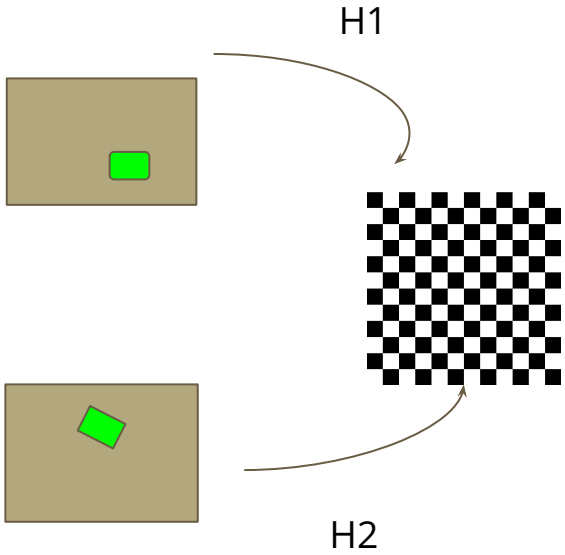
Plan next week

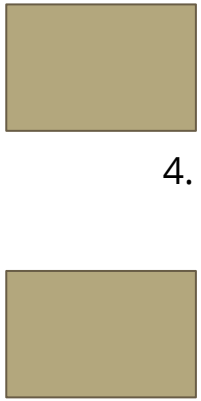
1. Design overall flow of the whole algorithm
    a. Background subtraction for initialising tracking
    b. Object tracking algorithm for tracking the object throughout
    c. Use Multi-camera tracking to track the object
    d. Use deep lab cut / background subtraction to reinitialise pose if the object is lost
2. Figure out the problem with Sequence capture
3. Get actual mice data and train a deep lab cut model

Progress this week

1. Design overall flow of the whole algorithm
    a. Background subtraction for initialising tracking
    b. Object tracking algorithm for tracking the object throughout
    c. Use Multi-camera tracking to track the object
    d. Use deep lab cut / background subtraction to reinitialise pose if the object is lost
2. Figure out the problem with Sequence capture
3. Get actual mice data and train a deep lab cut model

MultiView Object Tracking



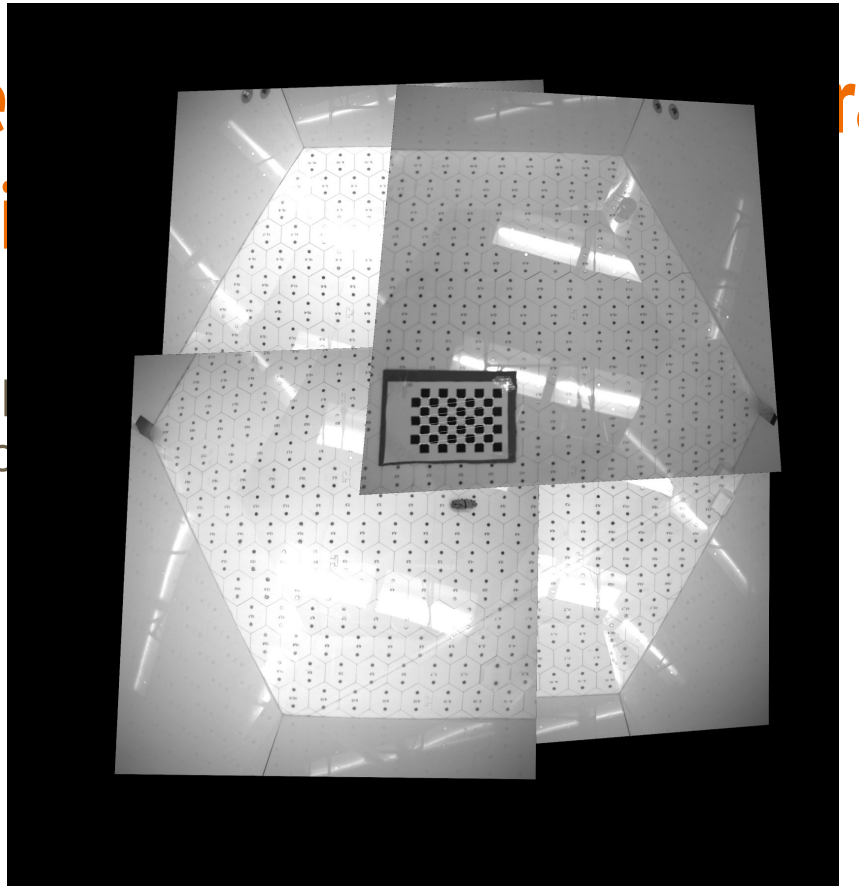


1. Homography between view1 and view2: H1 * H2.inverse()
2. Similar homography between any two views can be calculated
3. Everytime, we track an object, we can transform the bounding box to the other camera views to see if the bounding box lies within that image
4. If we find a bounding box lying within another camera view, we initialise another tracker for that camera view (therefore we can have a maximum of 4 trackers alive at the same time)

Need of multivie rate

Homography esti

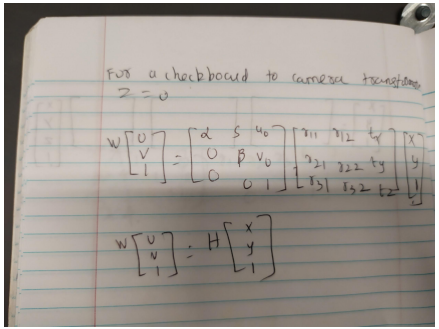
1. Not good enough.
2. Played around with
   Wasn't able to impro
3.



Can Homographies be estimated more accurately?

1. Another procedure to estimate the homographies

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & s & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



2. The plane to plane transformation between checkboard and image can be derived as given in the image
3. We could correct for lens distortion as well.

Homography estimation

1. Yet to be tested on stitching


```
{"dist": [[-0.17088236577043575, 0.10450283574783911, 0.0017341115493829569, 0.0028818779339680645, -0.029081650823638 27]], "homography": [[-1544.5287883160504, -72.3186224097207, 995.0883494555469], [7.932083332738679, -1522.2671243226625, 1002.2416335463522], [-0.0414895466791626, -0.02148228377367016, 0.9989079682334236]], "intrinsic": [[1502.7206350501465, 0.0, 1059.6370497962657], [0.0, 1501.227887257477, 1033.5198296470644], [0.0, 0.0, 1.0]], "extrinsic": [[-0.9985654635170984, -0.03297698684176849, -0.042184516231910676, -10.331506764016], [0.03384719467312018, -0.9992252147652727, -0.0200832664189511, 19.183665415568043], [-0.0414895466791626, -0.02148228377367016, 0.9989079682334236, 32.73355199031345], [0.0, 0.0, 0.0, 1.0]]}
```

Methodology suggested

1. If homography estimation is accurate, we can get a very accurate mechanism for transferring bounding boxes from one camera to another
2. If that is not accurate, we can avoid the transfer mechanism all together and use four background subtraction algorithms (Remember BG subtraction runs at 600-700 fps in CUDA)
3. If background subtraction for the whole image seems to slow down, we can have a background subtraction mechanism for the overlapping areas alone

Progress on Trigger

Multiple Devices. Uncheck all E8 devices except for the one connected to the camera to be used. Select OK > Open.

2. Configure camera settings in the “Capture & Adjust” window. When capturing from one camera, use of Free Run mode is recommended. To adjust the capture resolution, under the AOI tab uncheck “Max AOI” and adjust the “AOI Width” and “AOI Height”.

3. In the View window, select Capture > Sequence Capture > Video to Disk File.

4. A "Video to Disk File: Capture & Adjust" window will pop up. Click Browse, browse to the location to save the sequence to, and enter a file name for the image sequence. Click Accept.

5. If the “File Write Protect” box is checked, uncheck it.

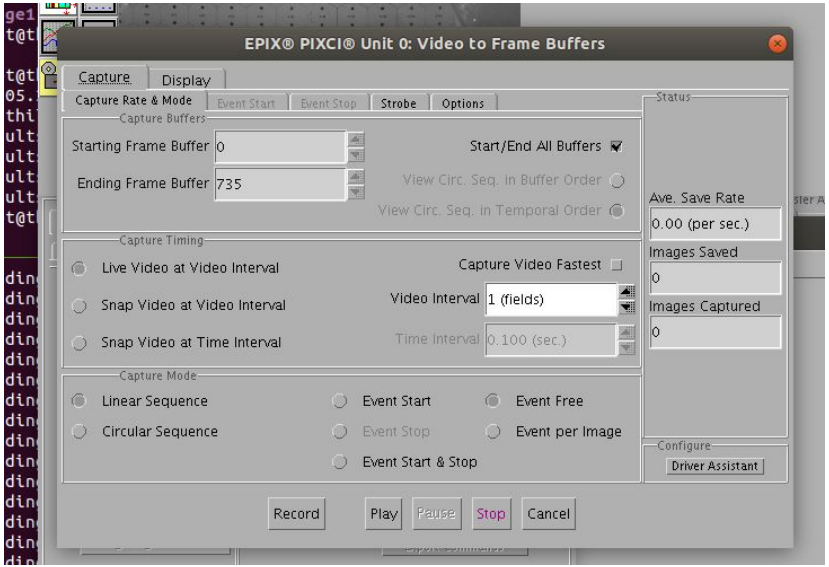
6. In the "Video to Disk File: Capture & Adjust" window, select the Capture tab.

7. Enter the number of images to be captured in the "Images to Save" box. Note that XCAP reports how much drive space the sequence will occupy after “Required Disk Space”.

8. Make sure "Live Video at Video Interval" is selected, and make sure the Video Interval is set to 1. This ensures that all images coming from the camera will be captured.

9. Click Record to begin sequence capture.

Page 5 / 7



Absence of a synchronised capture could affect object transfer mechanisms

2 / 135 * 2048 = 30 pixels

Plan for next week

1. Get mice running and record videos
    a. Try out Homography based transfer mechanism vs background subtraction based mechanisms to compare and contrast the two methods
2. Progress on trigger

Progress this week

1. Captured live mice data
2. Ran background subtraction based methods on live mice data and did timing analysis

Video output

Timing analysis

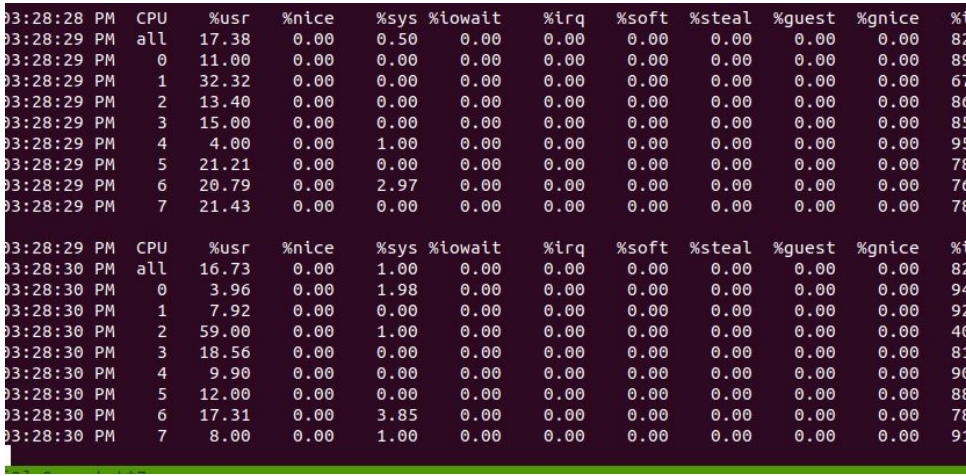
Total time taken for the algorithm = 30-32 ms

Fps = 30-33

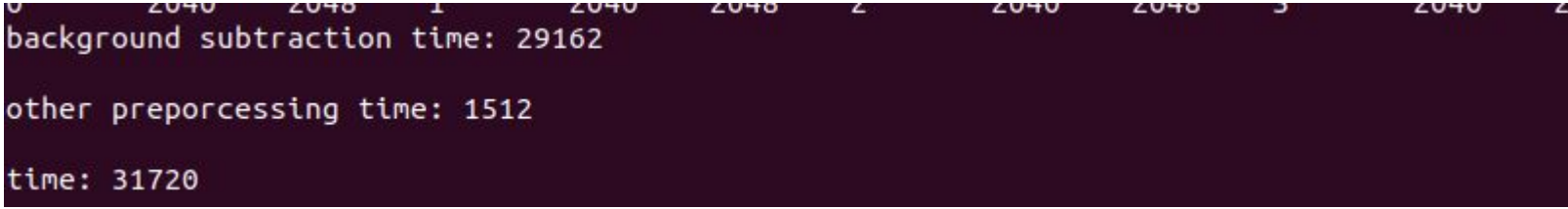
```
fps: 31.6396
0       2040    2048    1       2040    2048    2       2040    2048    3       2040    2048    here
time: 31463

fps: 31.7834
0       2040    2048    1       2040    2048    2       2040    2048    3       2040    2048    here
time: 31557
```

Can multithreading help? No

1. Almost the same time (30-33 ms)
2. 29-32 fps
3. Timing break down
   a. Background subtraction - 29 ms
   b. Other processing - 1-2 ms



03:28:28 PM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%i
03:28:29 PM	all	17.38	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	82
03:28:29 PM	0	11.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	89
03:28:29 PM	1	32.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	67
03:28:29 PM	2	13.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	86
03:28:29 PM	3	15.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	85
03:28:29 PM	4	4.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	95
03:28:29 PM	5	21.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78
03:28:29 PM	6	20.79	0.00	2.97	0.00	0.00	0.00	0.00	0.00	0.00	76
03:28:29 PM	7	21.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	78
03:28:29 PM	CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%gnice	%i
03:28:30 PM	all	16.73	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	82
03:28:30 PM	0	3.96	0.00	1.98	0.00	0.00	0.00	0.00	0.00	0.00	94
03:28:30 PM	1	7.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	92
03:28:30 PM	2	59.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	40
03:28:30 PM	3	18.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	81
03:28:30 PM	4	9.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	90
03:28:30 PM	5	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	88
03:28:30 PM	6	17.31	0.00	3.85	0.00	0.00	0.00	0.00	0.00	0.00	78
03:28:30 PM	7	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	91



```
background subtraction time: 29162

other preporcessing time: 1512

time: 31720
```

Total time in contention

1. Capture time - 30 ms (since frame rate was around 37.5 fps)
2. Algo time - 30 ms
3. Total time - 60 ms
4. Fps - 16-17 fps

Scope for improvement

1. Reduce the resolution of the image through binning
    a. Try capture at 150 fps (6-7 ms for capture)
        i. 13 - 14 ms for capture (1024 x 1024)
        ii. 6-7 ms for capture (512 x 512)
    b. Try the whole algorithm at lower resolution
        i. May be 15 ms for 1024 x 1024
        ii. May be 7-8 ms for 512 x 512

Plan for next week

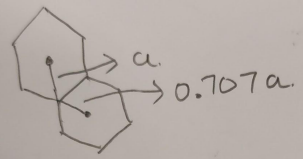
1. Downscale images and check the whole system throughput
2. Add support for visualising mice trajectory
3. Associate mice position with

McCormick
Northwestern Engineering

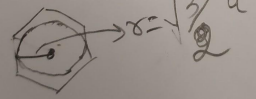
Robert R. McCormick School
of Engineering and Applied Science
Northwestern University
www.mccormick.northwestern.edu

Proof :-

At the boundary.



Safe zone.

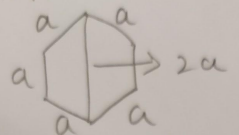


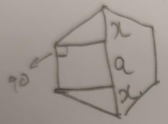
Assumed correct % $= \dfrac{\pi \times 0.86^2 a^2}{\frac{3\sqrt{3}}{2} a^2} \times 100.$

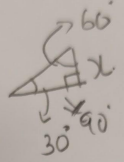
$\simeq 89\%$

Robert R. McCormick School of Engineering and Applied Science
Northwestern University
www.mccormick.northwestern.edu

Fact 1:-







$\sin 30 = \frac{x}{a}$

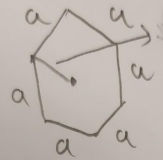
$x = a/2$

$y = a + 2x$

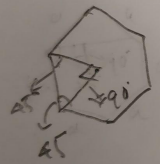
$y = a + a$

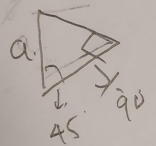
$\boxed{y = 2a}$

Fact 2:-



$\frac{a}{\sqrt{2}} \simeq 0.707a$





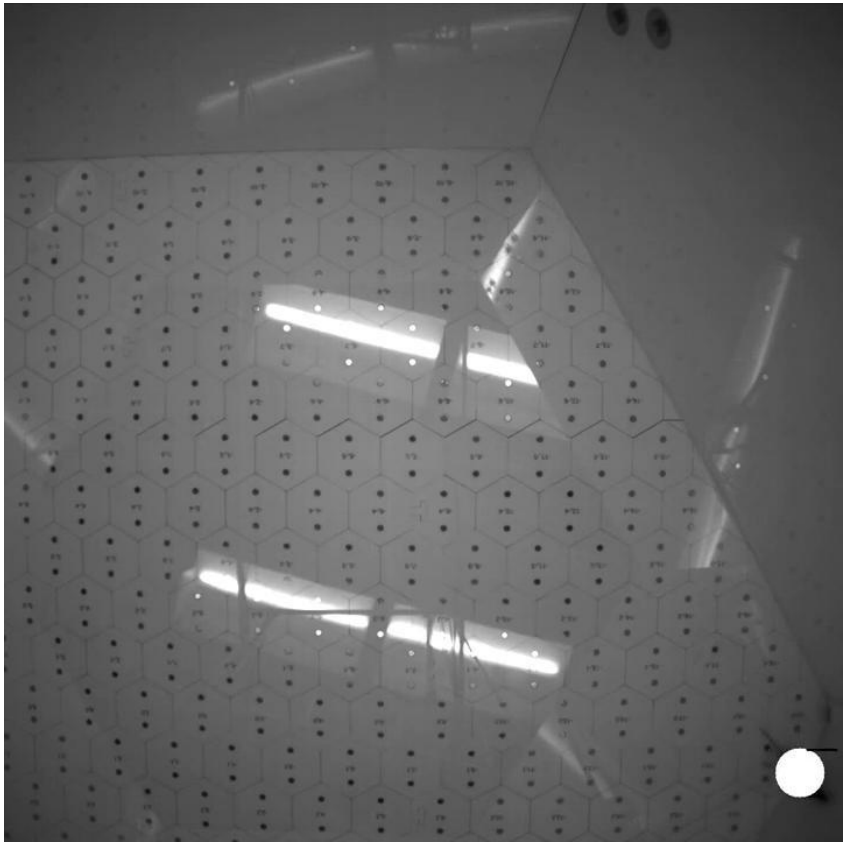
$\sin 45 = \frac{x}{a}$

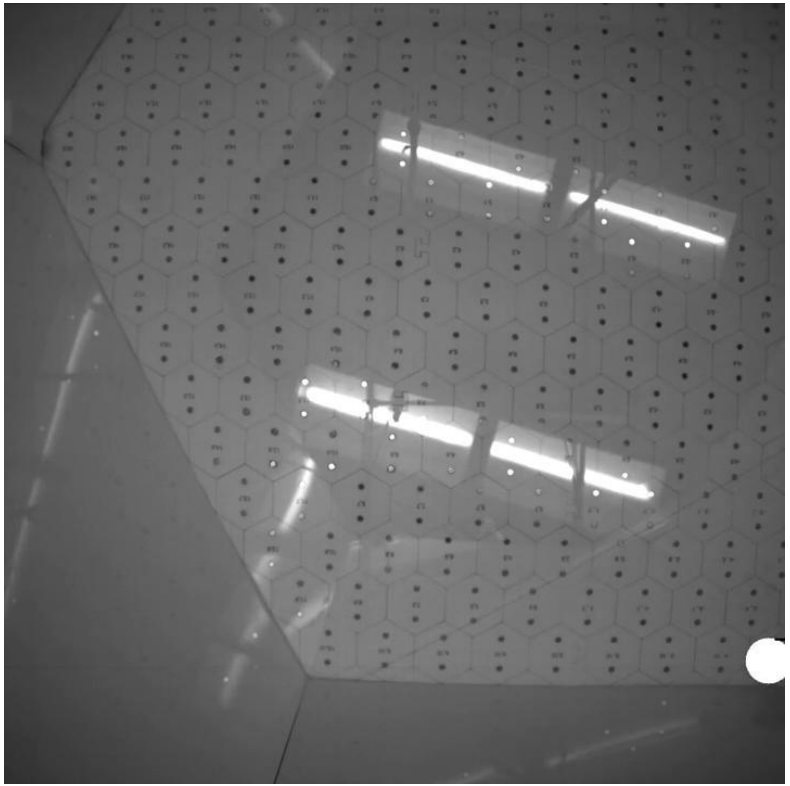
$x = \frac{a}{\sqrt{2}} = 0.707a$

Progress last week

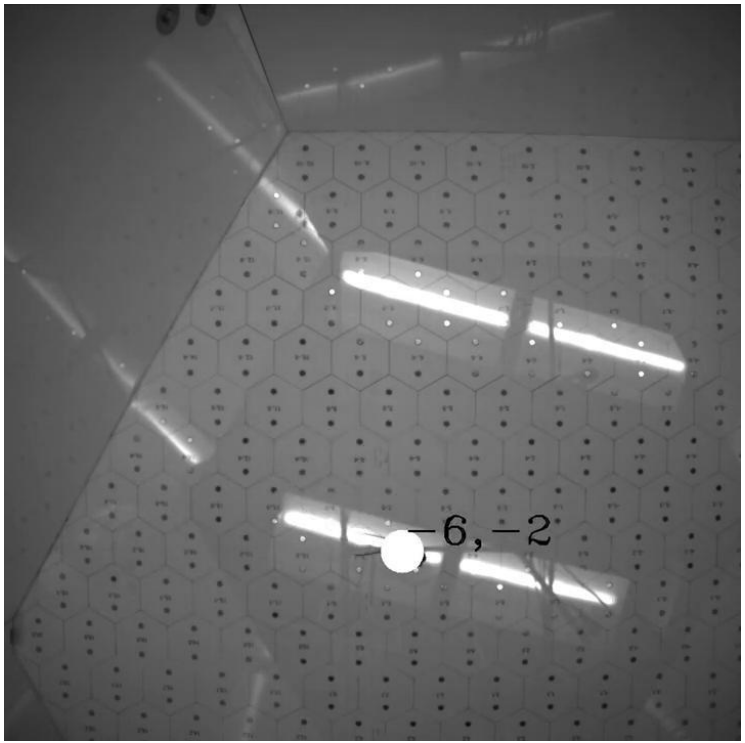
1. Did cell association, reduced image capture resolution through binning and tested the whole system
    a. The system now runs at at least 140 fps (max of 7.1 ms processing)
2. Trained deeplab cut model on our actual lab data
    a. The model seems good
3. Performed lens distortion calibration for our camera images.
    a. Images after lens correction look really good!
    b. Yet to test the result with camera calibration

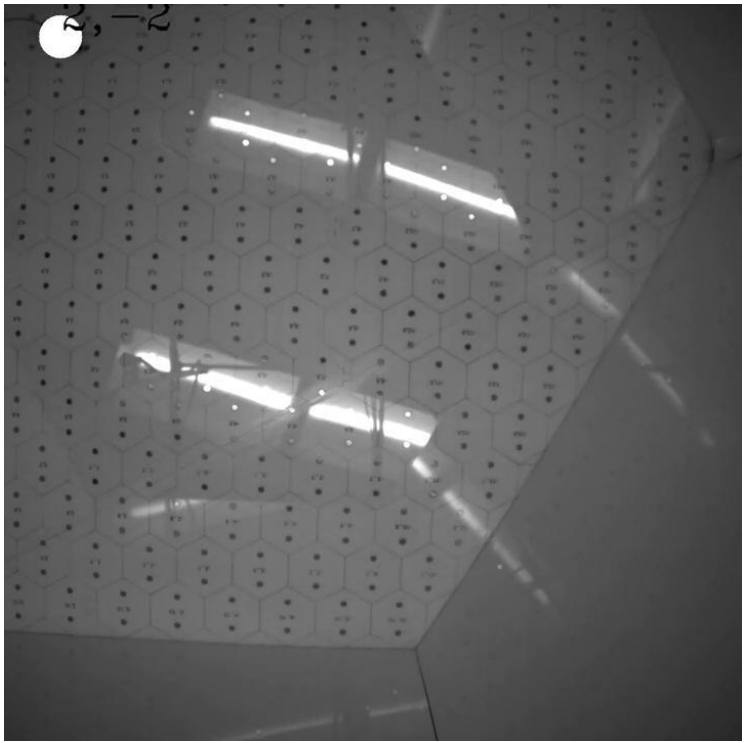
Background subtraction video outputs



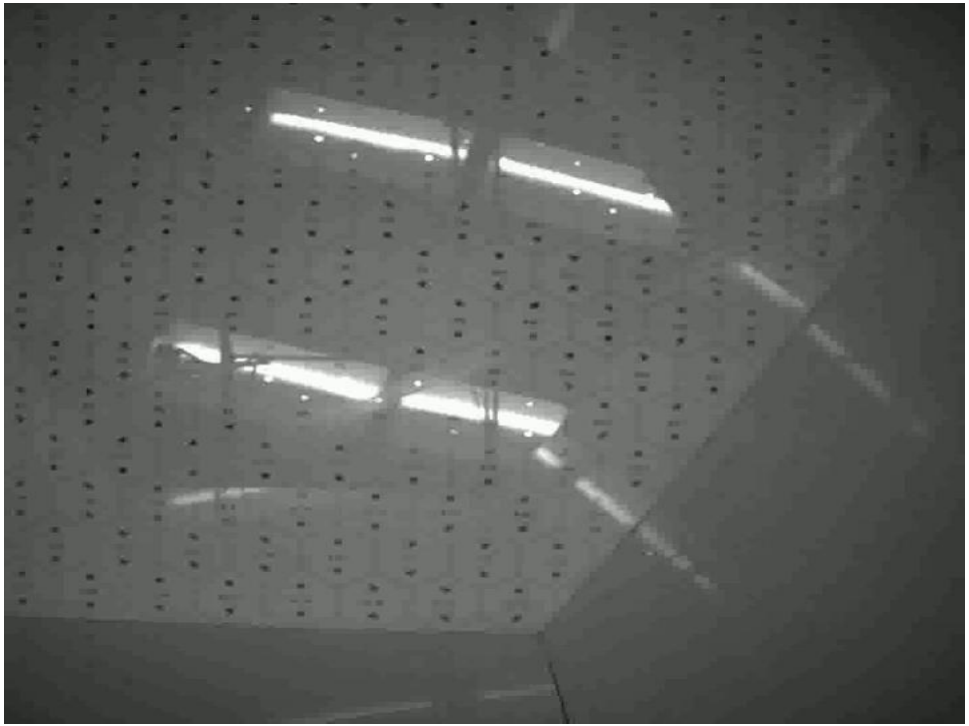


Background subtraction video outputs





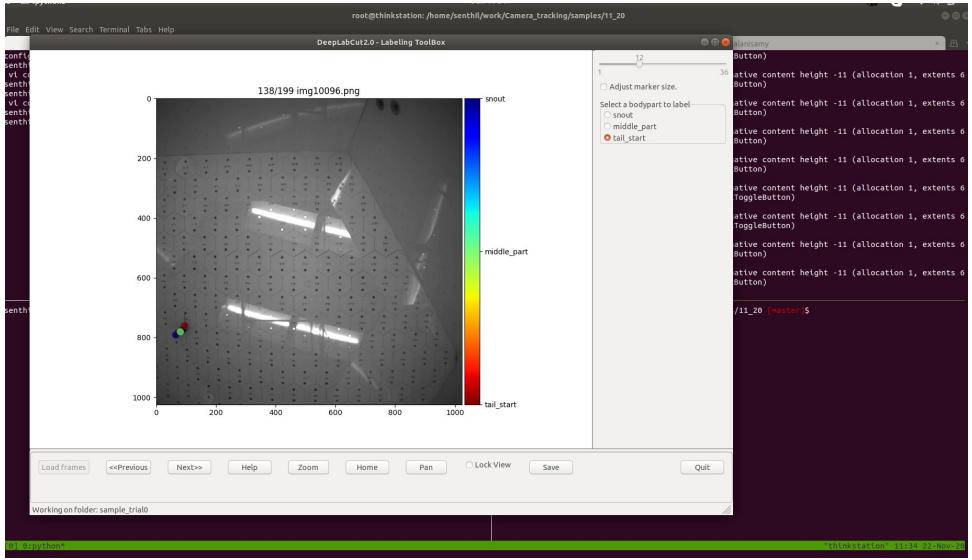
DeeplabCut outputs



Lens distortion correction results

Some manual works to be beware of

```
 1 31 239 6 10
 2 99 237 4 10
 3 167 234 2 10
 4 239 231 0 10
 5 312 229 -2 10
 6 386 225 -4 10
 7 463 224 -6 10
 8 539 223 -8 10
 9 616 223 -10 10
10 656 291 -11 9
11 578 290 -9 9
12 499 289 -7 9
13 423 290 -5 9
14 347 291 -3 9
15 274 295 -1 9
16 201 299 1 9
17 132 298 3 9
18 63 302 5 9
19 29 363 6 8
20 96 362 4 8
21 164 363 2 8
22 235 358 0 8
23 308 359 -2 8
24 383 357 -4 8
25 460 354 -6 8
26 538 354 -8 8
27 616 356 -10 8
28 692 356 -12 8
29 731 422 -13 7
30 654 421 -11 7
31 577 423 -9 7
32 499 420 -7 7
33 420 420 -5 7
34 344 422 -3 7
35 271 423 -1 7
36 198 424 1 7
37 128 424 3 7
38 62 424 5 7
39 26 489 6 6
40 95 487 4 6
41 161 489 2 6
42 232 490 0 6
43 306 489 -2 6
44 383 490 -4 6
45 459 488 -6 6
46 538 488 -8 6
47 616 490 -10 6
48 691 487 -12 6
49 767 488 -14 6
50 806 556 -15 5
51 731 557 -13 5
52 654 557 -11 5
53 577 556 -9 5
54 499 556 -7 5
55 421 556 -5 5
```



1. Cell Association
2. Data labelling

Plan for next week

1. Check image stitching with lens distortion correction
2. Run the system with actual robot and see if we can reliably disambiguate ran from the robot (It should definitely be possible)
3. Try out strategies to automate data association process.
4. Multiprocessing support for the whole pipleine
    a. Whole process should happen in a fast master thread
    b. But image writing for offline processing should happen in a slower asynchronous thread.

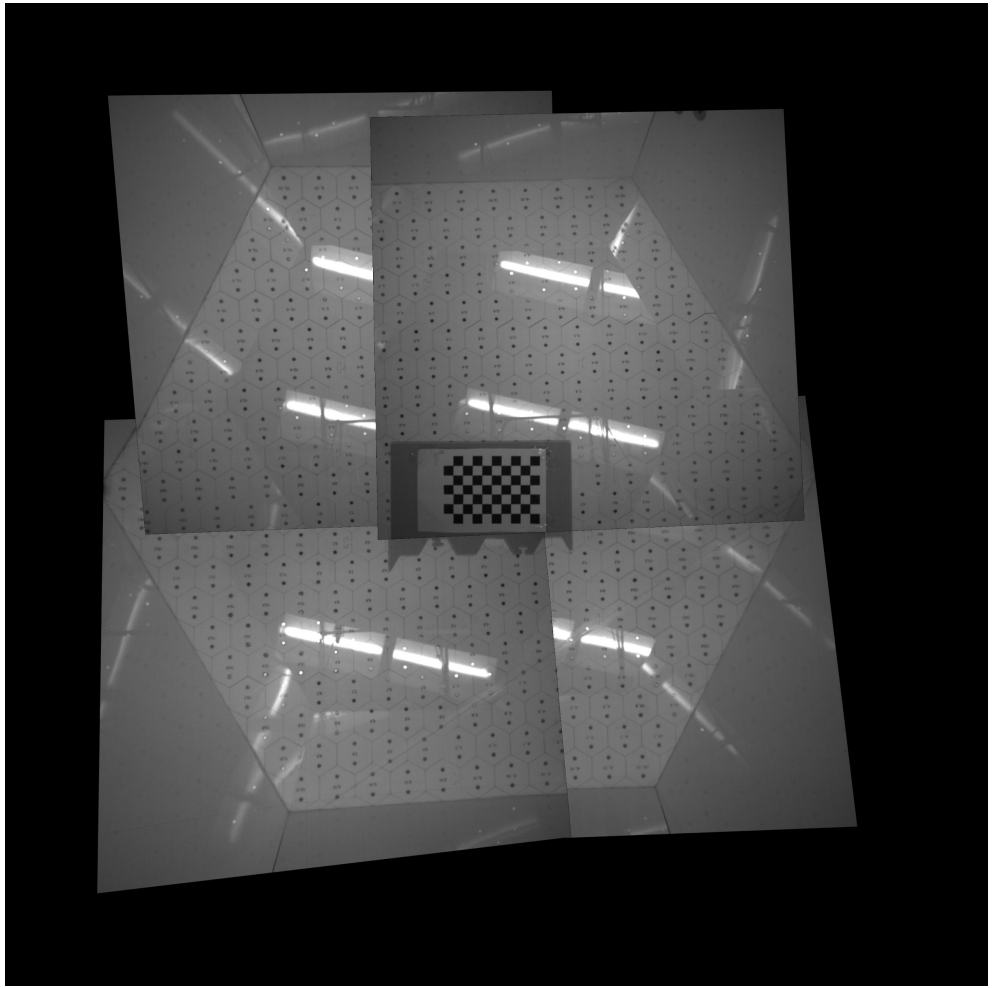
1. https://github.com/ultralytics/yolov5
2. https://www.simonwenkel.com/2020/02/13/opencv_cuda_for_background_subtraction.html
3.

ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > iOS

Progress this week

1. Multiprocessing support for data collection
2. Checking image stitching with lens distortion correction (The results are better but not the best)

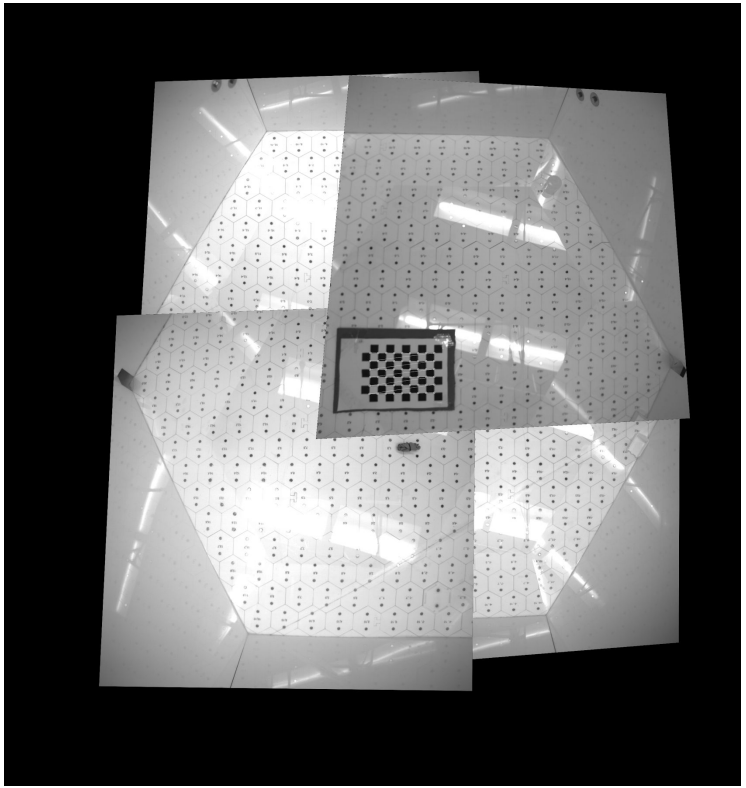
Multiprocessing support for writing files

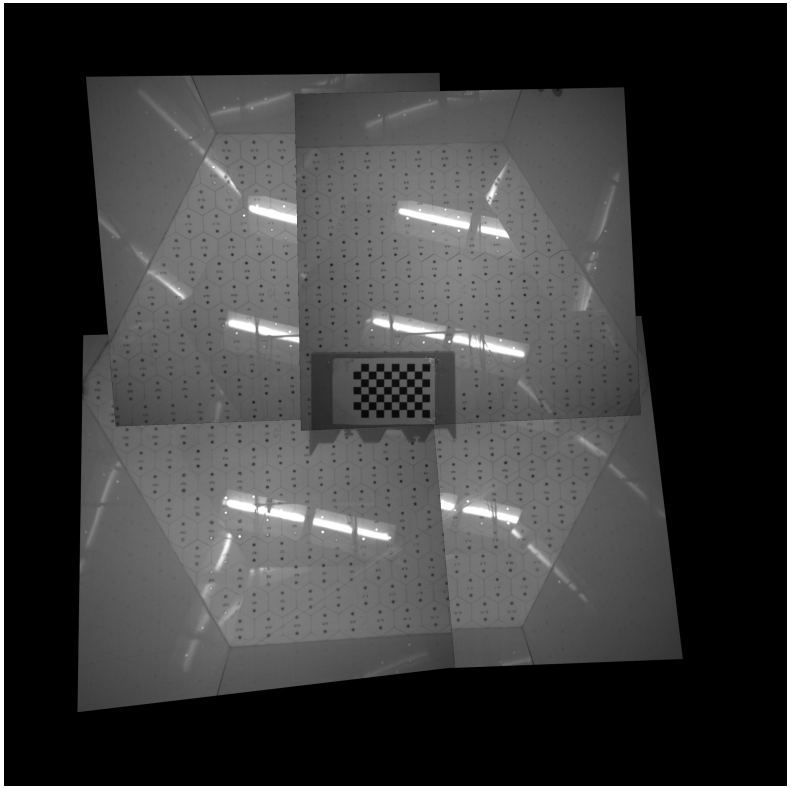


Types of video outputs

1. Raw four camera images
2. Processed four camera images (camera images with mice tracking output superimposed)
3. Stitched Image video

Previous output





Timing Analysis

1. Time required for lens distortion correction - 10 ms (for all images put together).
2. It’s better to do the online process without lens distortion correction.

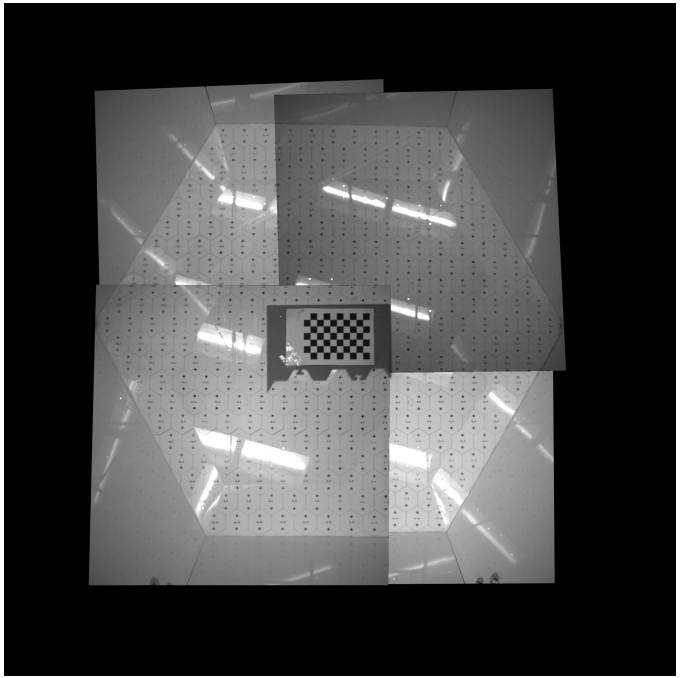
Plan next week

1. Test the whole system with both the robot and the mice moving in the scene
2. Document and Clean up code for the system.
3. Try to improve image stitching further by doing one more set of images.
4. Last week: Knowledge Transfer.

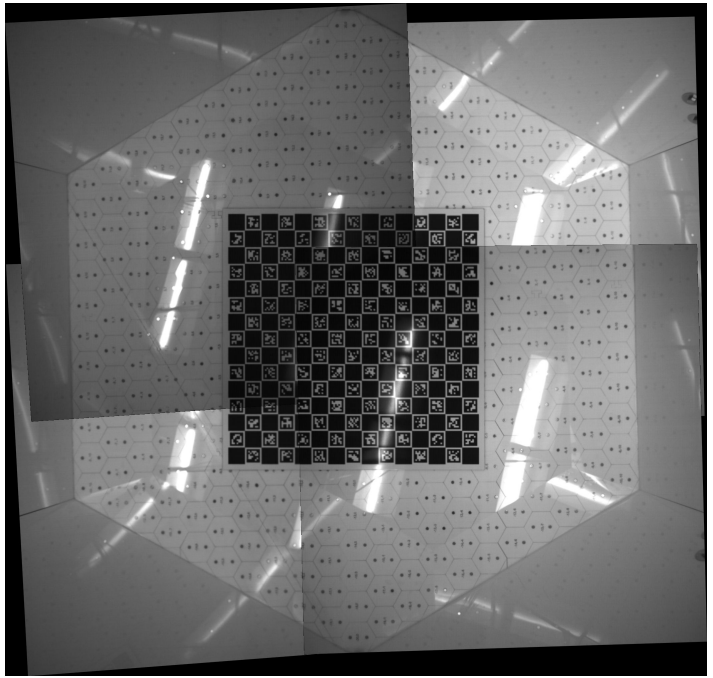
Progress last week

1. Image stitching problem fully fixed
    a. Stitching works perfectly (Thanks to Charuco boards and proper lens distortion calibration).
2. Video writing provision for stitched images added and tested with real mice and robot experiments
3. Mechanism for robot tracking added
    a. Tested tags: Too expensive
    b. Using low level computer vision techniques to track the robot with an underlying assumption - The robot is a dark square.

Image stitching progress

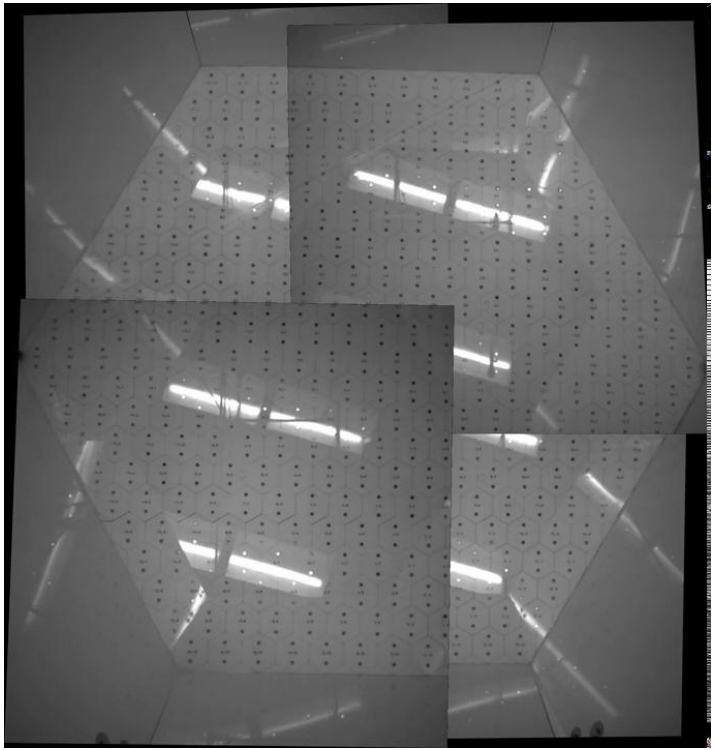


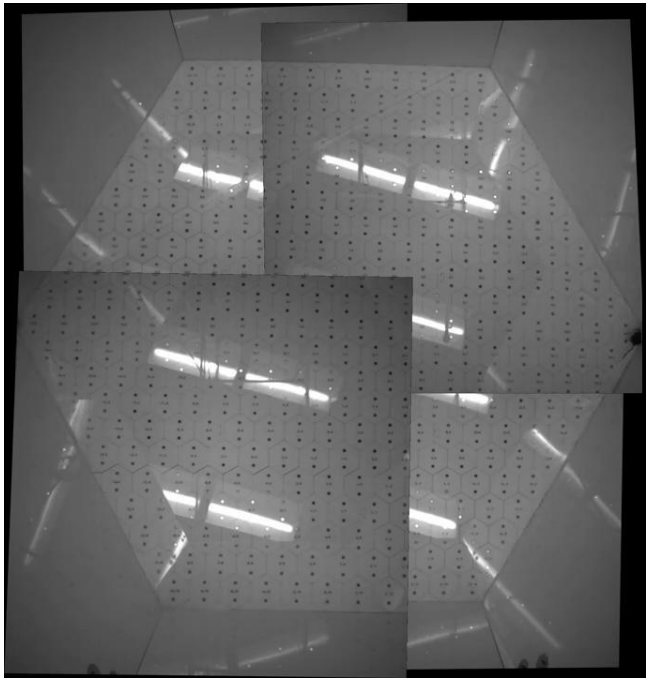
Proper lens distortion calibration



Improvement due to Charuco board

Video writing provision for image stitching added

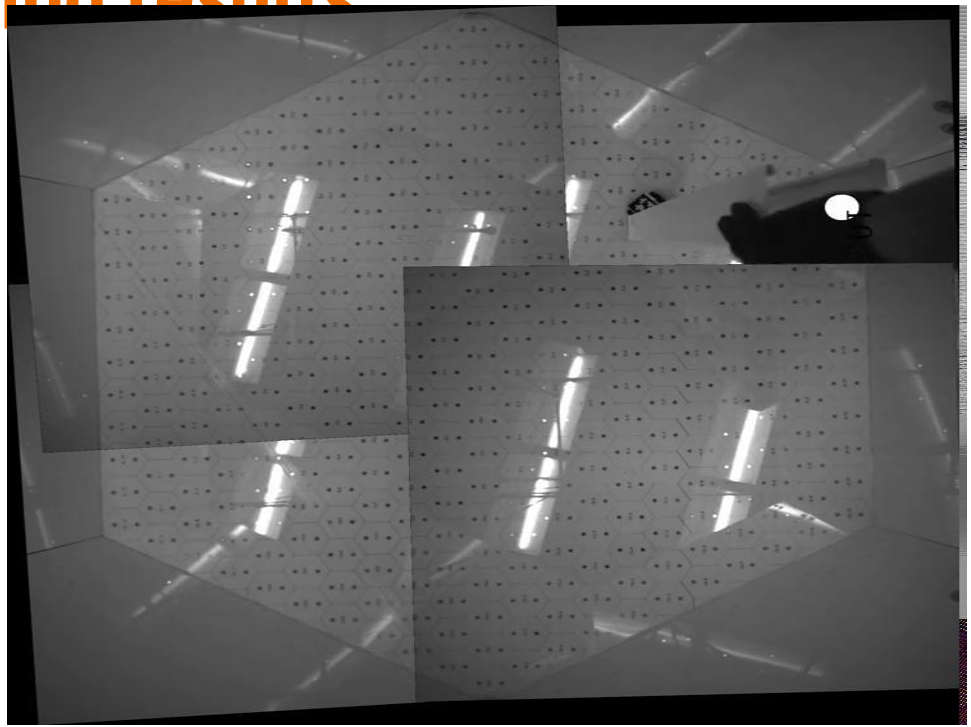




Mechanism for robot tracking

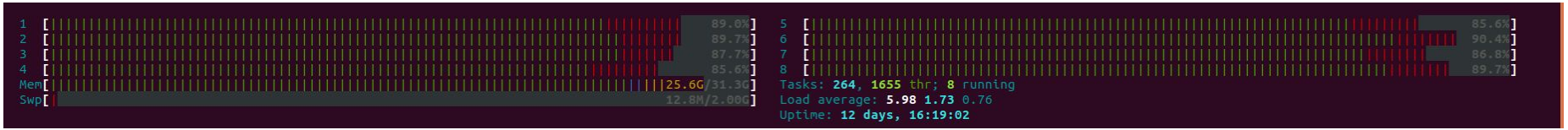
1. Tried Charco tags: Too expensive - takes more than 35 ms sometimes for a full sized image. Reduced resolution leads to degraded performance.
2. Low level computer vision image / processing - The robot is just a black square in the field of view of the camera.

Robot tracking results



Need a CPU with more cores.

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
15271 root      20   0 50.359g 331284  93780 S 683.1  1.0   1:47.57 back_ground_sub
28139 root      20   0  117272  14324   6352 S   4.3  0.0   0:02.88 sshd
13204 senthil   20   0 8154492 891504  38620 S   3.0  2.7 114:10.72 java
15386 senthil   20   0  820996  30840   9404 S   1.3  0.1  35:53.51 python3.6
21142 root      20   0   31896   5692   3164 S   1.3  0.0   0:19.33 tmux: server
 1911 senthil   20   0 5287976 742500  93856 S   1.0  2.3 171:28.40 gnome-shell
```



Speed mostly remains at around 100 fps while it falls to 40 or 50 fps sometimes.

Need a CPU with more cores.

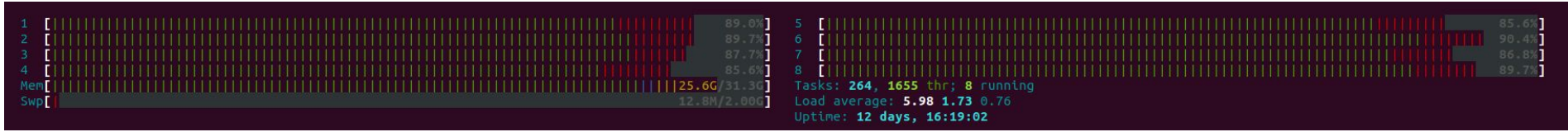
Total hypothetical cores needed

Thread counts:-

1. 4 threads for writing raw images
2. 1 thread for constructing stitched image
3. 1 thread for writing stitched image.
4. 2-3 threads for parallel pixel transfer within image stitching.
5. 4 threads for tracking robot position.

At Least 13 cores needed. Safe to order a 32 core system

Memory build up in the system



There is some unclosed system processed in our everyday work flow - 30 teamviewer process, firefox, genome processes etc...

A total recap of all the work done through this project

1. Driver, camera and hardware setup
    a. Understanding cameras, driver code of epix to construct higher level abstractions for our use
    b. Driver installation and setup.
    c. Configuring camera parameters for high speed image capture.
    d. Figuring out mounts, camera mounting heights for the whole system and setting up the system in lab
    e. Sorting out hardware issues, reassembling the system, ordering new hardware and setting up the system in the lab
    f. Failed effort on hardware based synchronisation.

A total recap of all the work done through this project

2. Multi view Image stitching.

a. Initial experiment to stitch images done remotely at home.
b. Algorithmic and code optimisation to speed up image stitching (Decoupling homography estimation, using checkerboard for homography points, compiling OpenCV with all necessary flags).
c. Multiprocessing to improve the stitching speed (The final speed for image stitching is under 100 ms - initial time was more than a second).
d. Debugging problems with stitching in lab - lens distortion, use of charuco boards.
e. Writing stitched Video using a multithreaded pipeline.

A total recap of all the work done through this project

3. High speed image tracking

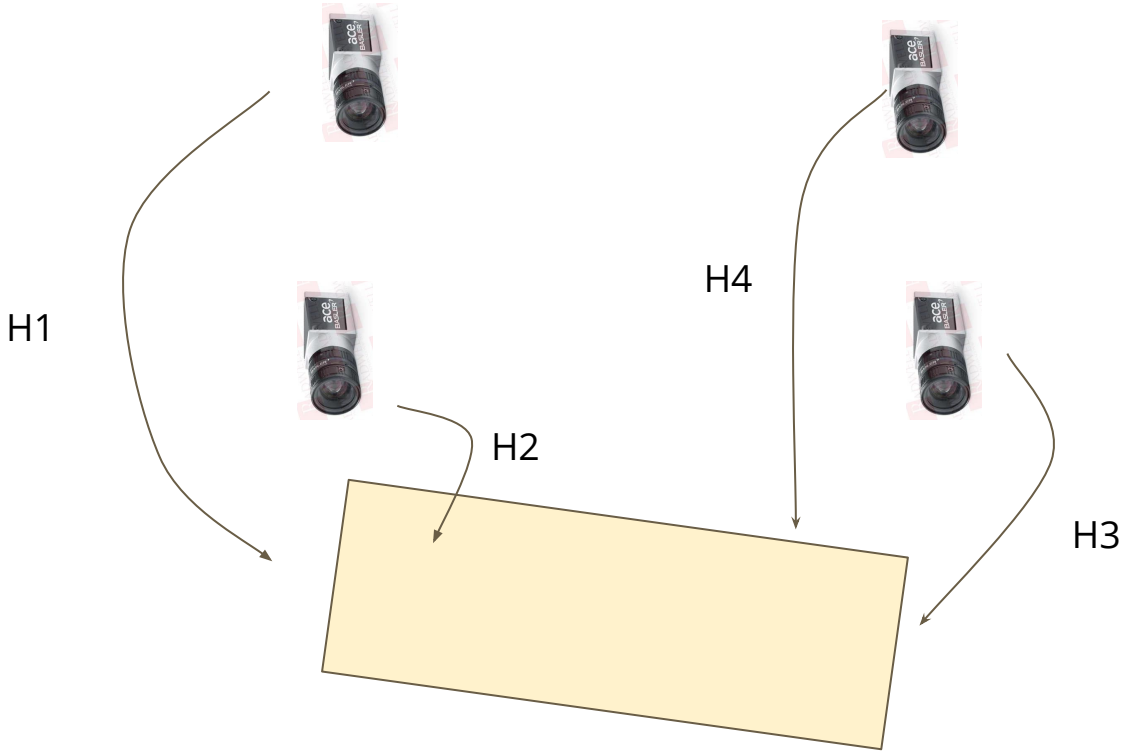
1. Initial try with deep lab cut. Trained and deployed models for deeplabcut.
2. C++ api for integrating python deeplabcut code.
3. Tried out faster mechanisms for mice tracking - object tracking, background subtraction etc.. and finally settled with CUDA enabled background subtraction to track at less than 7-8 ms.
4. Mechanisms for tracking the robot using low level computer vision.
5. Overall pipeline - A neat C++ architecture, high multithreaded and easy to use with high level abstractions

Knowledge necessary for someone to work with and debug the system when something goes wrong

1. Knowledge of hardware, drivers regarding the camera system, camera hardware, camera related processes.
2. Knowledge of homography based multiview image stitching.
3. Knowledge of computer vision algorithms at least at a high level - background subtraction, object tracking, low level computer vision.
4. Knowledge of multithreading in C++ (std::async atleast).
5. Knowledge of deep learning for using deeplabcut.
6. Knowledge of C++.
7. A look at the code base

Plan for final week

1. Do a fully integrated test - possibly today
2. Document the system and write a post for portfolio
3. Knowledge transfer and handover of the system.


H1
H2
H3
H4