



COLLABORA

Working on Monado's Hand Tracking

Moses Turner



Open First



COLLABORA

Moses Turner

- [Freedesktop](#)
- [GitHub](#)
- [Blog](#)
- [@mosesturnervevo](#)
- moses@collabora.com

Open First





COLLABORA

Talk to me about

- Transhumanism
- Radical life extension
- Life as a virtual being
- XR, VR & AR
- Machine learning, AI and computer vision!
- Math!
- FOSS!
- Joining Collabora!

Open First





- **Motivation, prior art**
- Current status
- What's next?
- Wrapping up

Why optical hand tracking?

- 6DOF controllers are ideal for games and art, but I'm helping build the next general-purpose computing platform.
- Can't use controllers at the same time as a keyboard
- Markerless - no extra equipment besides HMD
- You can't forget your hands in a hotel
- More robust to unfavorable conditions than constellation tracking
- One input schema which we can emulate in different ways
- Largely proven out, see Mediapipe, Oculus Quest, Ultraleap, HTC
- Feels like *your hands!* Can be a lot more transformative than controllers.

Notable prior tracking pipelines

- [Ultraleap](#)
 - Works extremely well, but is closed-source and [has scary licensing](#). No-go for research purposes.
 - Requires an extra sensor that cannot be used for anything else
- [MEgaTrack](#) - Facebook
 - Our tracking is partly a replication of this. Works very well; hats off to the researchers behind this.
 - [Unofficial reimplementations of their model architectures](#)
 - Dataset is private
- [GANHands](#) - Max Planck Institut Informatik
 - Possibly SoTA for realtime egocentric RGB? I haven't kept up. Probably suitable for XR too.
 - Paper is not detailed enough, not enough of the code is published, and the dataset is not useful on its own.
 - Research institutions can request some version of the pipeline, just not me! Worth investigating!
- [Mediapipe](#) - Google
 - Anecdotal evidence that it's not good for egocentric tracking: [\[1\]](#) [\[2\]](#)
 - Questionable feature engineering; doesn't use gaussian priors
 - Dataset is private



Notable prior datasets and dataset generators

- Too much to list here. Only listing things I have some personal experience with.
- [FreiHand](#) - only 2D keypoints, no sequential data, annotations are not super accurate
- [CMU Panoptic Dataset](#) - ditto
- [InterHand2.6M](#)
 - + Very accurate 3D keypoints, sequential, includes camera calibrations, has lots of overlapping hands
 - - Cumbersomely big; has some rare but very wrong annotations that are painful to filter out.
- [GANerated Hands Dataset](#)
 - + Artificial dataset with trivially perfect annotations
 - - Only 2D keypoints, no sequential data, really weird cropping, no overlapping hands
 - --- Dataset generator and models are unpublished; you can't fix any of these problems. Please contact me if this changes!
- Everything from Max Planck Institut Informatik that tends to have licensing/publishing issues: I also wanted to use [HandTextureModel](#) and [MANO](#) but couldn't
- [Rendered Handpose Dataset](#) might be very good, gets cited a lot, haven't looked deeply into it.
- Epic Games' [MetaHumans](#) or [Mixamo](#) would be super good for generating an artificial dataset, but they have an explicit clause in their license barring you from doing AI research with it.



Our motivation

- Some good open-source datasets/pipelines, but licensing is tricky if you aren't a non-commercial research institution. Can be difficult to get commercial licenses.
- No good, tweakable artificial dataset generator with CC-BY or better licenses
- Lots of unexplored research areas due to a lack of foundational research that's fully open and easy-to-use
- It sucks to be doing commercial R&D in this space. Huge companies can sometimes do this profitably by vertically integrating, but it's risky.
- We want to help make it possible to try new things on a budget, using our software as a base to start from.
- Push the SoTA forward while also lowering the barrier to entry.



Explicit goals at 30,000 feet

- Publish a completely FOSS optical hand-tracking software that competes with or exceeds SoTA, runs in realtime, and has a C api. ✓
- Publish all real datasets we've collected, with open licenses that allow anybody to use and contribute to them. → SOON
- Publish all of the data and code used to generate our synthetic dataset, with the same set of licenses and contribution pathways. → SOON
- Keep improving what we've got, and make it easy for other people (individuals, research institutions, companies; literally anybody) to help! ✓
- Contract work for people who want a specific R&D area explored, or for people who want tight integration with a product they're shipping. → SOON





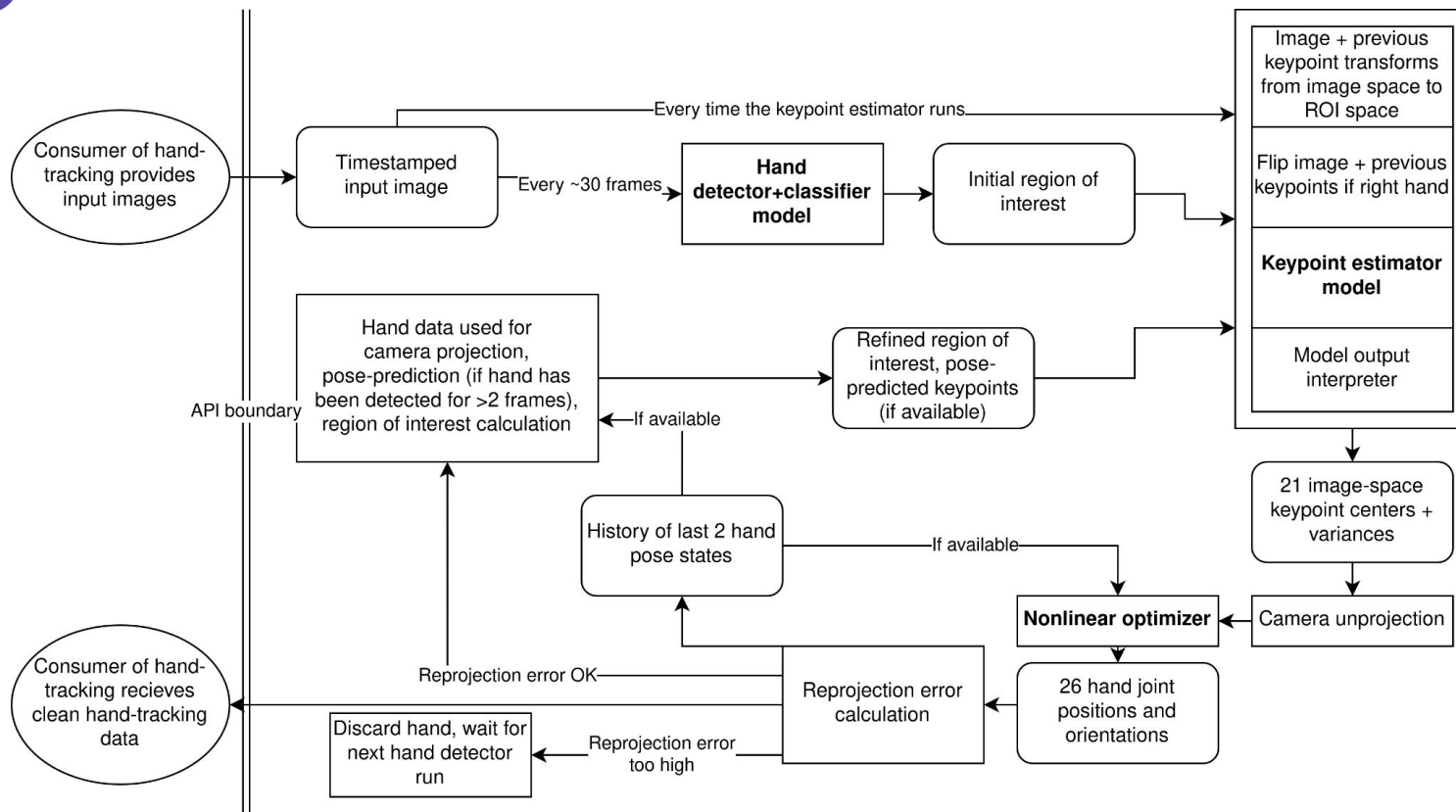
- Why do we care so much about optical hand tracking?
- **Current status**
- What's next?
- Wrapping up

Boom!

(Prerecorded video just in case)



(About a week ago)
Same optimizer; better
keypoint estimator trained
with artificial data! Jitter
is mostly gone now.



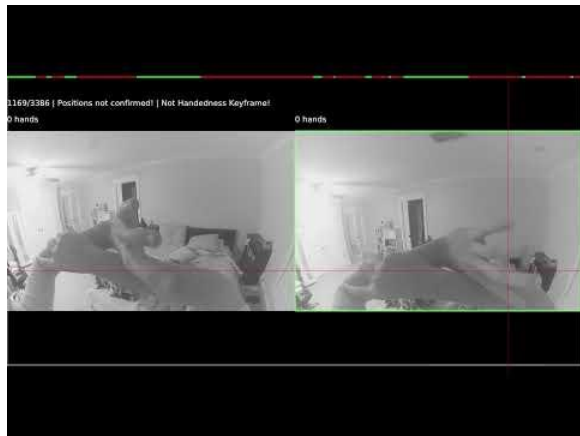
Mercury's hand detector + classifier

- Sees in grayscale, 320x240.
- [SimDR](#) model architecture (Deviation from MEGATrack: Their model architecture is much more efficient, and it's on my backlog to train a new one.)
- Outputs heatmaps for hand centers and radii
- Datasets:
 - [EPIC-KITCHENS](#)
 - [EgoHands](#)
 - [TV-Hand and COCO-Hand](#)
 - An in-house dataset collected with a North Star and annotated semi-automatically
- Training data pipeline uses a bunch of PyTorch Datasets and a ConcatDataset



Training the hand detector

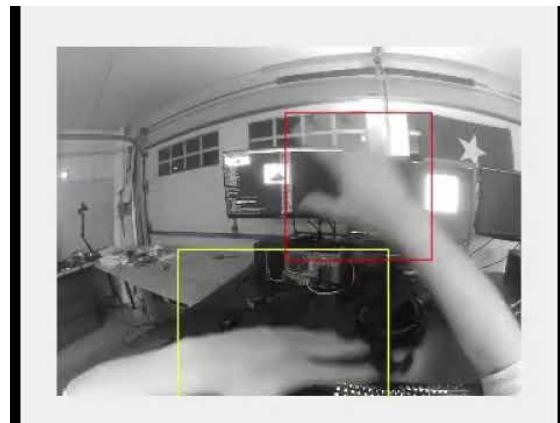
Annotation

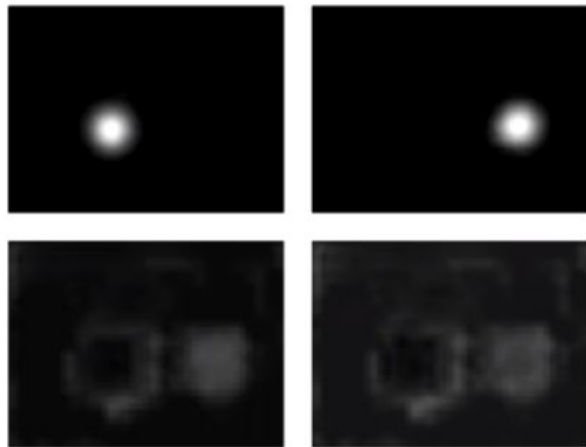


Training



Inference





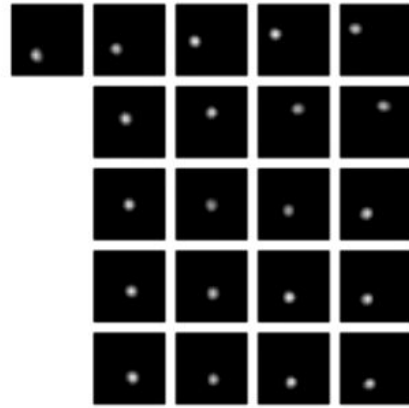
COLLABORA

Open First

Mercury's hand keypoint estimator

- Trained a custom hand keypoint estimator
- Sees in grayscale, 128x128. (Deviation from MEgATrack: MEgATrack sees in 96x96)
- Outputs are
 - 21 2D heatmaps predicting the most likely keypoint position in pixel coords
 - 21 1D heatmaps predicting the most likely keypoint depth relative to the middle-proximal joint
- [Same model architecture as Quest hand tracking](#)
- Datasets:
 - Small greenscreen dataset I collected and annotated using Mediapipe
 - CMU panoptic dataset
 - [FreiHand](#)
 - Artificial dataset I generated (talk about this later!)





COLLABORA

Open First

Review: Problems with existing public hand landmark datasets

- Many don't have strictly sequential annotations: instead, you get a bunch of unrelated individual images. (Notably, InterHand2.6M does this exactly right)
- You need to calibrate your cameras and store joint locations in 3D relative to the hand! Pretty much all datasets just annotate in 2D image-space, making them unhelpful for more complicated model architectures. (Notably, InterHand2.6M does this exactly right)
- Basically all real-world datasets have some percentage of incorrect annotations that are hard to filter out. (InterHand was the worst here, so bad that I decided to give up on using it. Last I checked was April 2022; it may have been fixed since then.)
- Privacy and liability issues. You might be screwed if even one image has unwanted PII.
- Feels like almost every dataset has a non-commercial license; you can't ship a commercial product out of what we have now because of dataset licenses
- If you collect a new real-world dataset, you have to worry about the above and:
 - Am I going to accidentally lose data?
 - How long will it take my annotators to annotate everything? How often are they making mistakes?
 - Am I using the right type of cameras for my application?
 - Panoptic studios are expensive



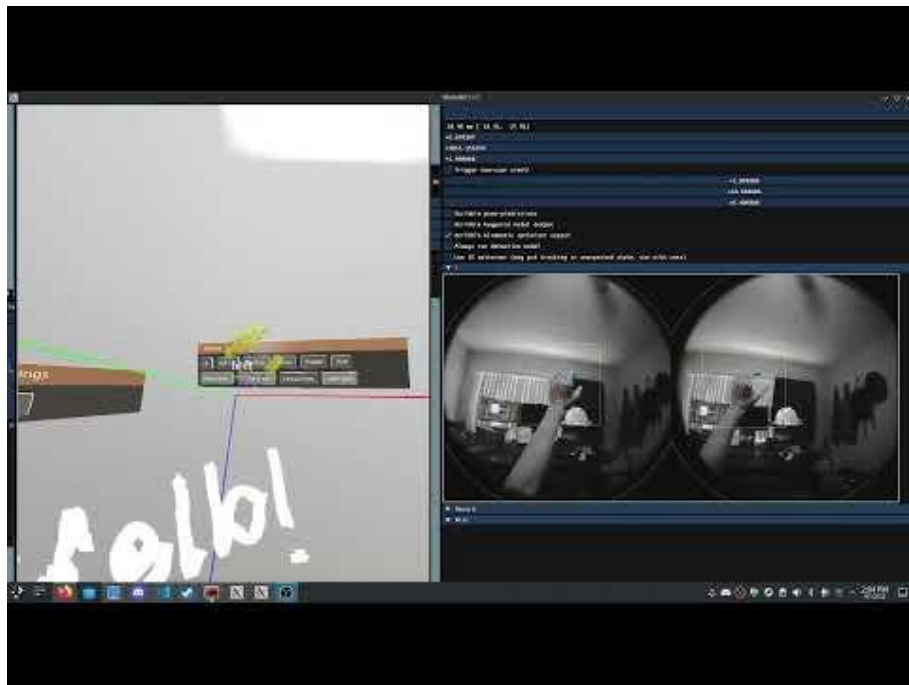
Solution: Synthetic data!



COLLABORA

Open First

Solution: Synthetic data!



What I showed you before was trained on a lot of synthetic data and a little bit of real data, but we can also track hands using `_only_` synthetic data!

The quality isn't quite as good yet, but it's totally a fixable data problem: we need to add more diverse poses, clothing, rings, etc. to the dataset generator.

Solution: Synthetic data!

- ML mesh generation isn't quite good enough yet, so we're using "classical" techniques.
- Started with some very high-quality hand scans from an asset store
 - Having a small licensing issue here - we will publish existing scans or create new ones hopefully by Q1 2023
- Rigged them in Blender
- Wrote a C++ pipeline that
 - Generates pose data by permuting a wrist-pose dataset I collected using Lighthouse tracking with a finger-pose dataset I collected with our hand tracking under good lighting conditions
 - Runs blender with a custom Python script controlled by environment variables. Blender sets up lighting, HDR background, animates the hand and renders a bunch of cubemaps
 - Takes the cubemaps and a random camera calibration, and distorts them into final images that look like they were captured on a HMD's onboard camera
 - Writes out the finger poses as 3D coordinates relative to the left camera
 - Is published [here](#) (code quality has room for improvement)



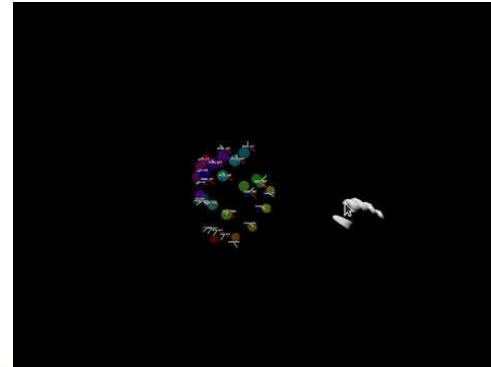
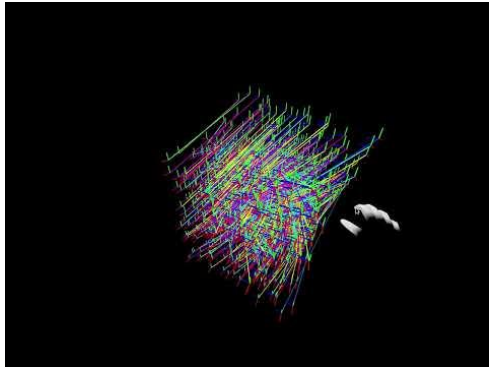
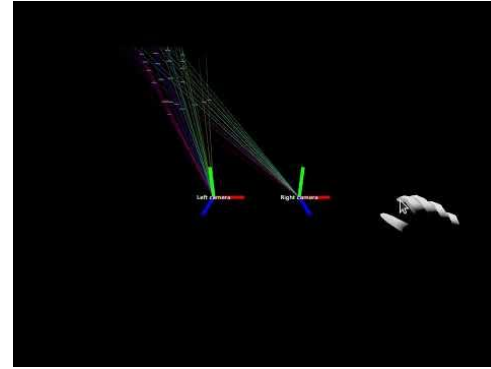
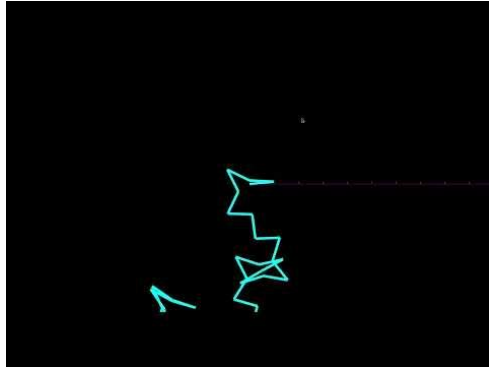
Training the keypoint estimator

- Basically same idea as the hand detector. Many PyTorch Datasets, rolled into a ConcatDataset
- [NoneChucks](#) is really nice for when you've found some mistakes in the dataset you spent 30 hours generating
- Fancy loss function that only looks at model's depth prediction if we have ground truth depth
- Training code is published [here](#) (code quality has room for improvement)

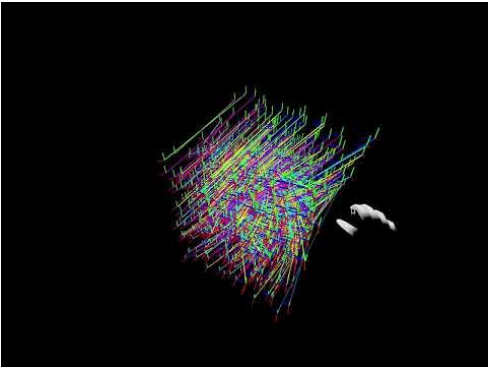
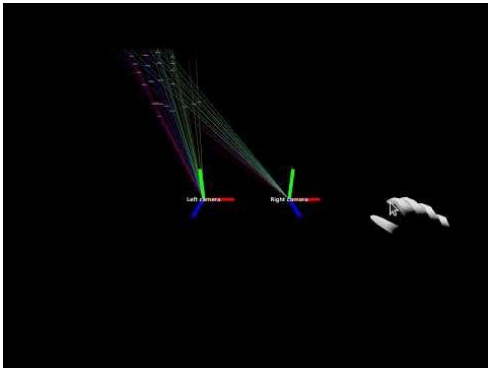
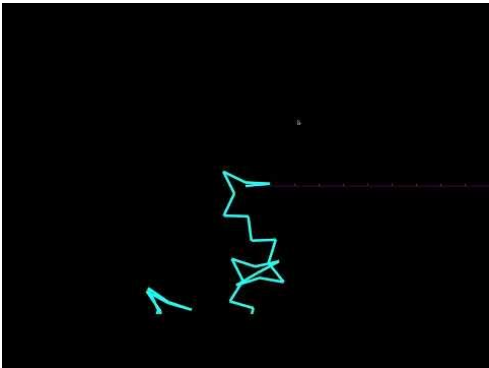
Mercury's realtime nonlinear optimization!

- We start with predictions of the keypoint locations in camera space, and want to end up with 6dof poses for each joint. You can very painfully and inaccurately do this by hand using camera triangulation and Inverse Kinematics, but it's way, way easier to...
- Write a function that takes a low-dimensional vector encoding hand pose (hand size, 6dof wrist pose, joint curl values), evaluates this to 6dof joint poses, and outputs a metric for "How close are these poses to the model predictions?"
 - Then, ✨ask your computer ✨ to evaluate this function many times, working towards a good solution. This is easily accomplished using gradient descent - you could easily implement this in PyTorch, for example. Levenberg-Marquardt is typically faster, and it's really just a mashup of gradient descent and Gauss-Newton.
- The exact same technique is used to solve for
 - HMD/controller pose for lighthouse tracking (Vive/Index)
 - HMD/Controller pose for constellation tracking (Oculus/WMR)
 - Head pose for SLAM (Basically everyone)
- Very simple to add extra terms to system: temporal consistency, extra sensors, myoelectric tracking, wrist-mounted IMU, etc.
- Works for underconstrained, exactly-constrained and over-constrained systems.
- More flexible than, and often more efficient than Kalman filters. Typically does the same thing given Gaussian priors.
- We're using [tinyceres](#), our fork of [Ceres](#). Ceres and Ceres's docs are extremely good, huge shout out to them!

Nonlinear optimization is awesome!



Ceres is awesome!





- Why do we care so much about optical hand tracking?
- Current status
- **What's next?**
- Wrapping up

What's next?

- Improving our artificial data to the point where real data has no marginal value
 - Long sleeves, rings, watches, tattoos
 - Better pose remapping from mocap datasets to the specific model
 - Soft-body flesh/tendons/skin simulation? (Do I know anybody who knows anything here? Code-first approach pls)
 - More mocap data
 - Random walks over plausible hand pose space
 - Intertwining fingers
 - Try simulating depth cameras and event cameras
- Train very slow but very accurate models to explore the limit of how accurate optical egocentric hand tracking can get
 - Diffusion in pose-space?
- Fix issues in NLO; try to use gaussian priors everywhere.
- Try a bunch of ideas for improving accuracy in our real-time models
 - Train a keypoint estimator that sees in stereo?
 - “Refinement” model that fixes up the region of interest, so that we can track quick movements with less jitter?
 - Lots of stuff to try here. “idk try it i guess” is by far the most effective way to do novel computer vision research.
 - Create better techniques to assess accuracy/performance
- Elbow tracking!
- Non-egocentric tracking, models that see in RGB
- Publish every last bit of our dataset, dataset generation and tracking code.



Crazier ideas, if you're bored

- Modular non-linear optimizer that takes all tracking data (egocentric hand tracking, external optical hand tracking, SLAM, EMG sensors, etc) and optimizes for the full upper body pose (or more) with gaussian priors and physics priors.
- Neural full-body-pose prediction. Take semantic data describing the past ~30 seconds of tracking data (head, left hand, right hand, etc.) and predict where they'll be for anywhere between 0-1000ms from now
- Reinforcement learning agent that tries to beat Levenberg-Marquardt for real-time speed
 - Nonlinear optimization is amazing, and here we're trying to nonlinearly optimize a nonlinear optimizer to make it really efficient for a known problem. Make sense?
 - Should handily beat LM if you fine-tune it on real tracking data - should be able to learn about the shape of the global manifold as well as know how to operate on the local manifold. Also means you (probably?) don't have to do a search for the best LM starting parameters
 - Yay, now you can safely anthropomorphize your XR tracking pipeline!
- More research into event cameras!!!






Careers - Open Source Consulting | Collabora - Chromium

careers.html

COLLABORA About Services Industries News & Blog Careers Contact



Open Source all day, every day

Since 2005, we've helped clients navigate the ever-evolving world of Open Source, enabling them to develop the best solutions – whether writing a line of code or shaping a longer-term strategic software development plan.




Our team of engineers and developers are among the most motivated and active Open Source contributors and maintainers around the world. They have a passion for technology and strive to accelerate the adoption of Open Source technologies, methodologies and philosophy.

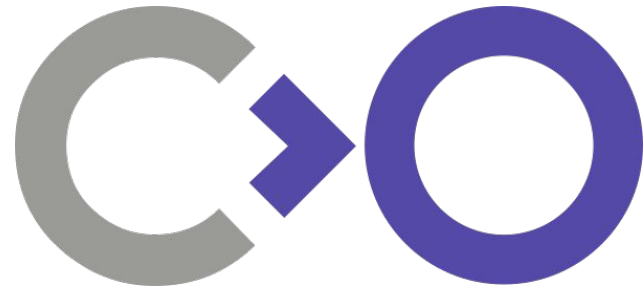
If you share this passion, and want to be part of a growing, globally distributed team, we want to hear from you!

Current Opportunities

Below is a list of our current job openings. If you see a position that interests you, click on the title to learn more and apply!

More on Careers @ Collabora

-  **Empathy first:**
Driving growth through people leadership
-  **Engaging in an "Open First" remote internship at Collabora**
-  **Why remote working can be good for people, business and environment**

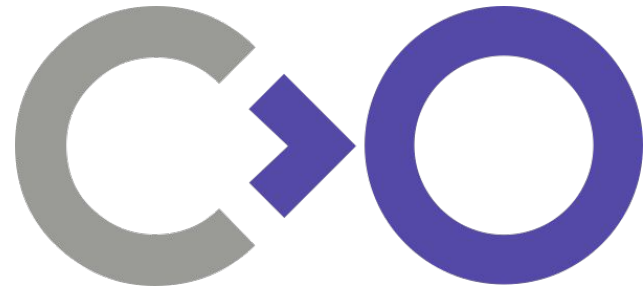


Questions?



COLLABORA

Open First



Thank you!



COLLABORA

Open First