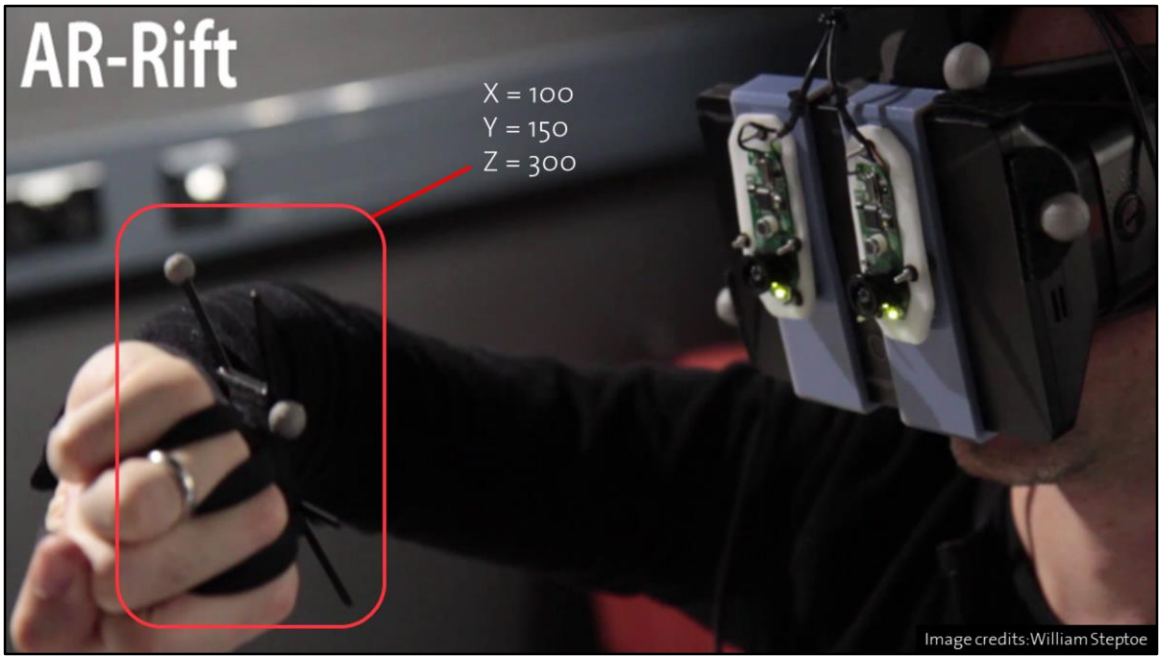
Joint Estimation of 3D Hand Position and Gestures from Monocular Video for Mobile Interaction

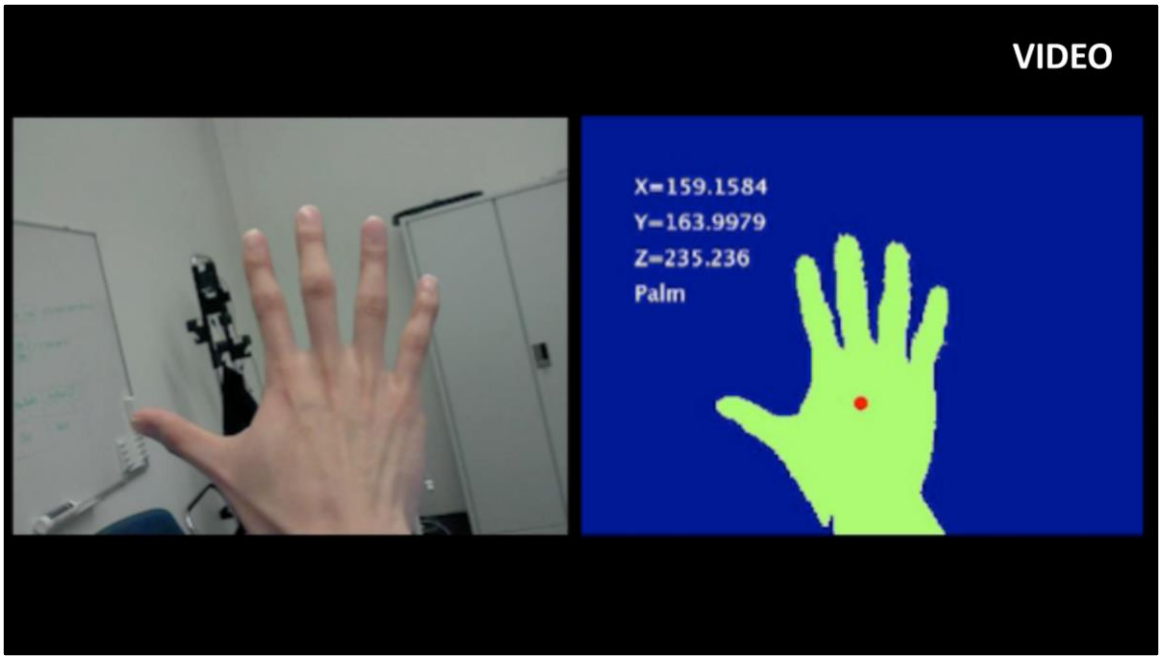
Jie Song[1], Fabrizio Pece[1], Gabor Sörös[1], Marion Koelle[2], Otmar Hilliges[1]

[1]ETH Zurich, [2]University of Passau

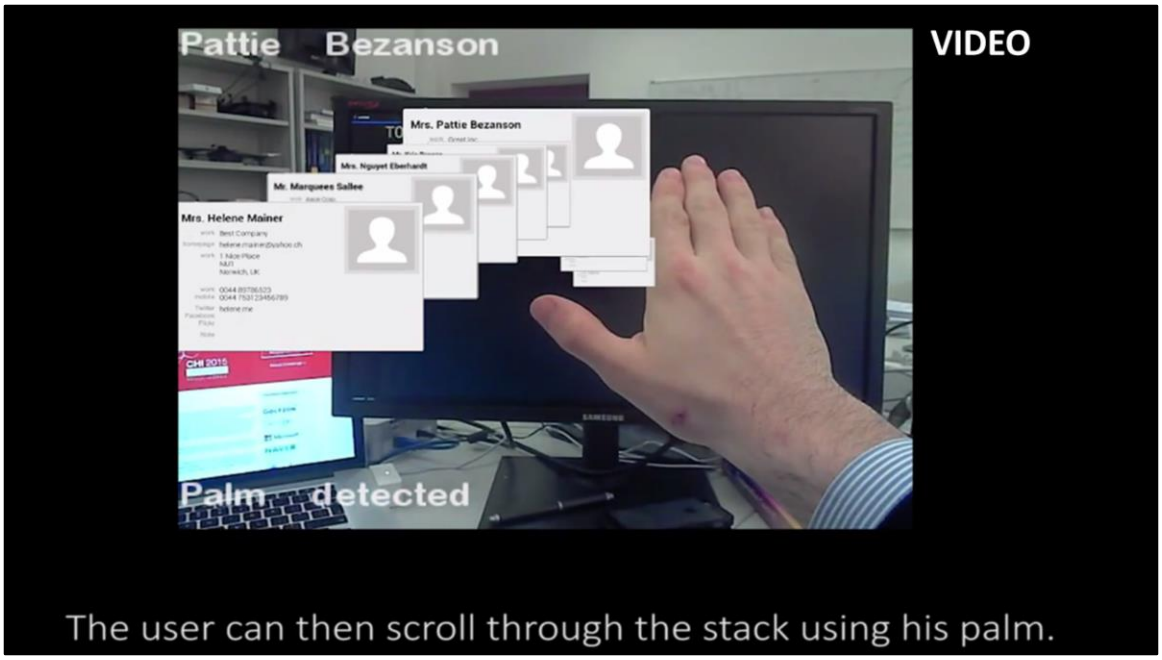
Hello, everyone, in this talk we present a method for **jointly** recognizing **hand gestures** and estimating **metric depth** for 3D interaction, based only on **RGB** input.



This topic is about **user input** for 3D user interfaces and AR/VR. Actually what we want here is mainly the **3D hand position.**
This has been done traditionally using **external tracking** which gives you **<x,y,z> position of your hand** <A>



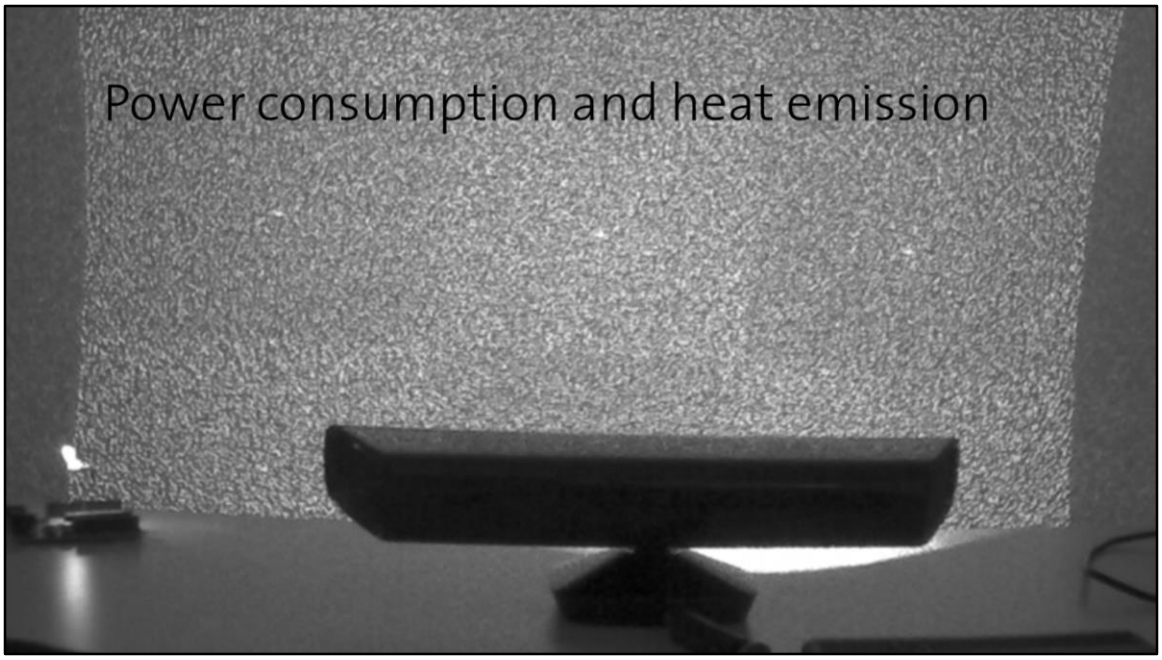
Our solution provides this x y z hand position by just **using monocular RGB input.**
<PLAY VIDEO>
In this video, on the left-hand side is the RGB raw input.
On the right-hand side is the (x,y,z) hand position estimated by our system.
The estimation is **robust under gesture variation**.
Here you see the **smooth transition** between palm and pinch gesture.



Here's an example that demonstrates our input functionality.
<PLAY NOW>
A PINCH gesture **brings up** the contact cards.
We can go through the cards by estimating the **depth of palm gesture**.
Finally, a FIST **gesture triggers** a call.

[Sharp et al. *Accurate, Robust, and Flexible Real-time Hand Tracking.* CHI '15.]

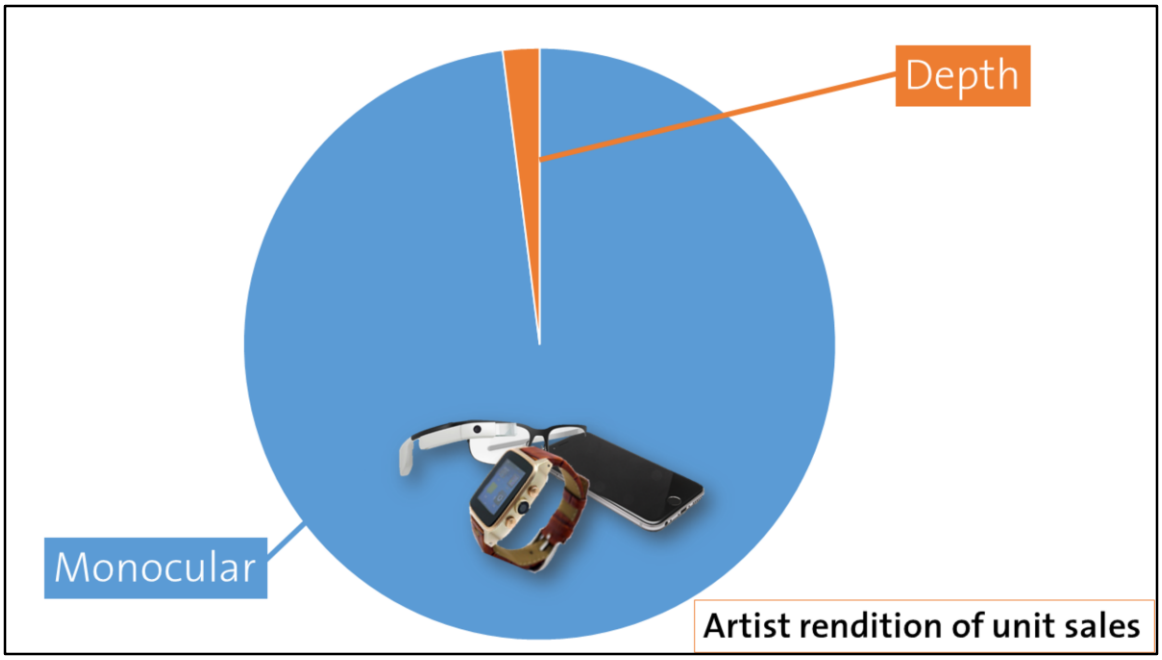
You might be wondering why **not directly using** a depth camera.
as you saw in **Jonathan's** previous talk. **Where he presented a hybrid approach for hand tracking.** The **fidelity** we can get from a depth is obviously very high. But there are still several issues with depth sensors **for body worn devices**. For example:
<A>



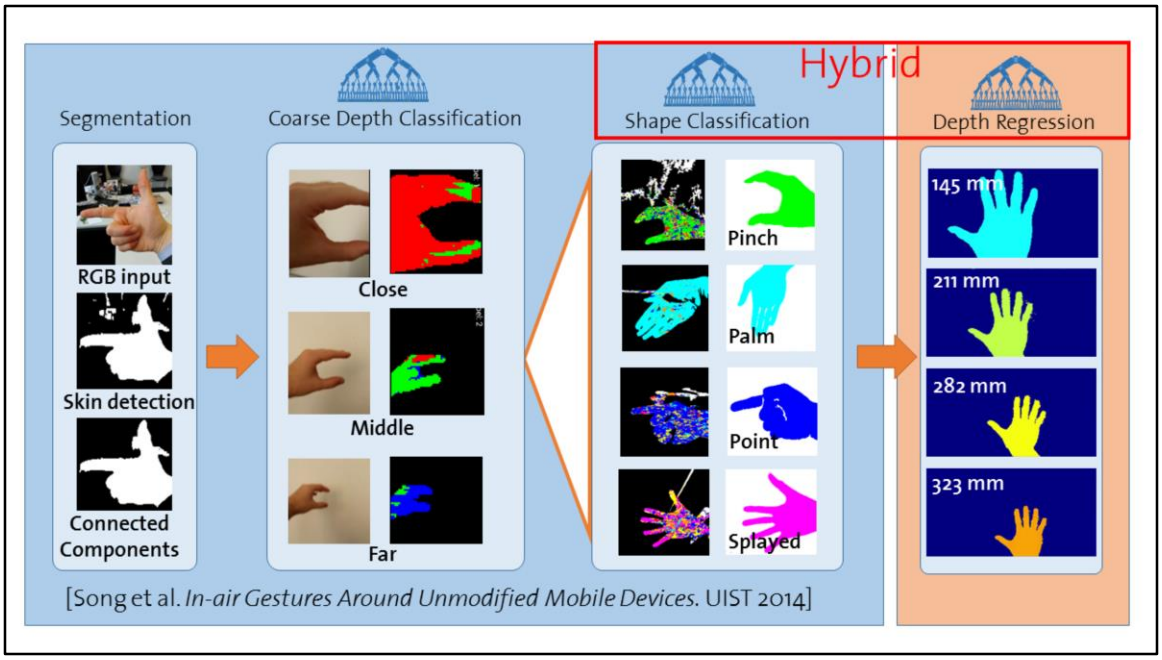
Active illumination based depth sensors has issues with power consumption and heat emission. **especially** when we wear them on our heads.

Physical baseline limit for stereo cameras

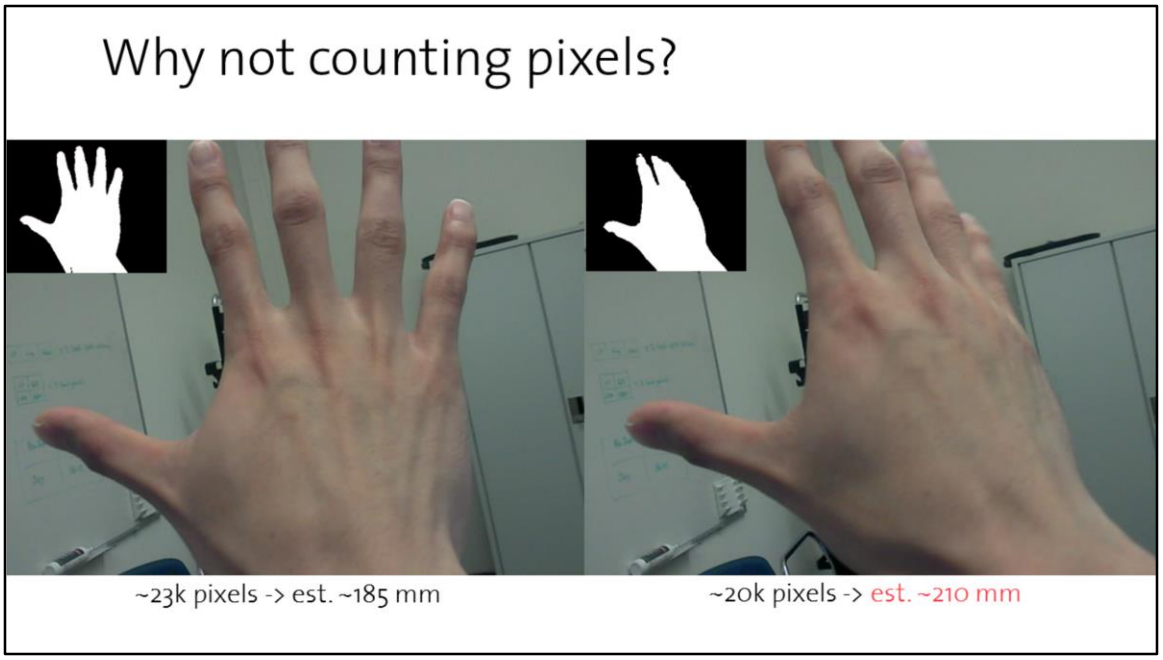
Passive stereo requires physical baseline, **impacting** overall size of the device.



Finally:
Even if there will be small depth cameras, **currently** there are certainly 6 billion devices with rgb cameras being sold out.
Tapping into this large install base is the main motivation for us to use just RGB cameras for 3D interaction.

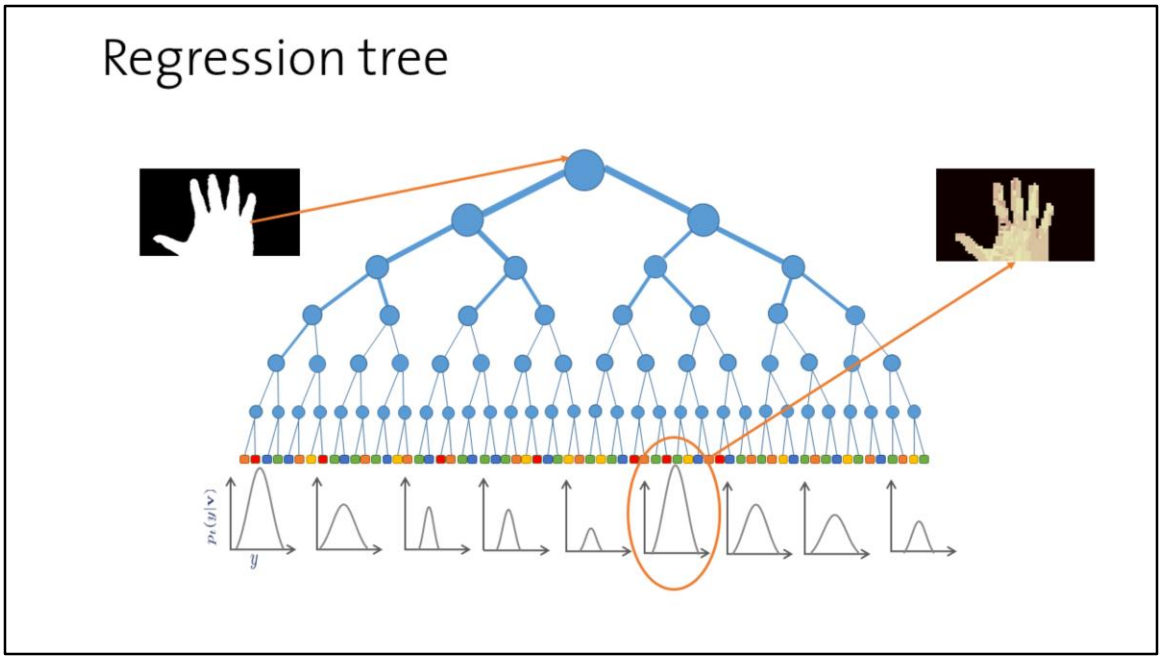


Here is a **overview** of our technical pipeline.
It is a **per pixel labelling process** based on multilayer random forests.
<A>
The **first part** of the flow does gesture recognition. It starts from a **RGB input**, after **segmenting** the hand out, the binary mask will be put into a series of trees to recognize hand gesture. This part is mainly based on our previous work. If you are interested in the details, you can refer to our last year's UIST paper.
<A>
The main contribution of this tech note is to **extend** our gesture recognizer to a **hybrid process** that can jointly classify hand gestures and estimate hand average depth.
We achieve this by a combination of classification and regression forests.
You may ask **why not just counting the pixels belonging to the segmented hand** and then **estimate the depth by interpolating the count**. There are several reasons why we chose to use more complicated machine learning method rather than naïve count for depth estimation.



For example, these are images of **same gesture at the same average depth**. But slight changes in appearance drastically change the depth estimated by counting foreground pixels even though the hand is perfectly segmented.

However, the most severe limitations happens under change in appearance of the hand.
For instance, in this figure you can see the same hand with same depth value and gesture, but with slight difference change in appearance.
<A>
A simple pixel count will show a 3K pixel difference, resulting in roughly 30mm variation in depth estimation!

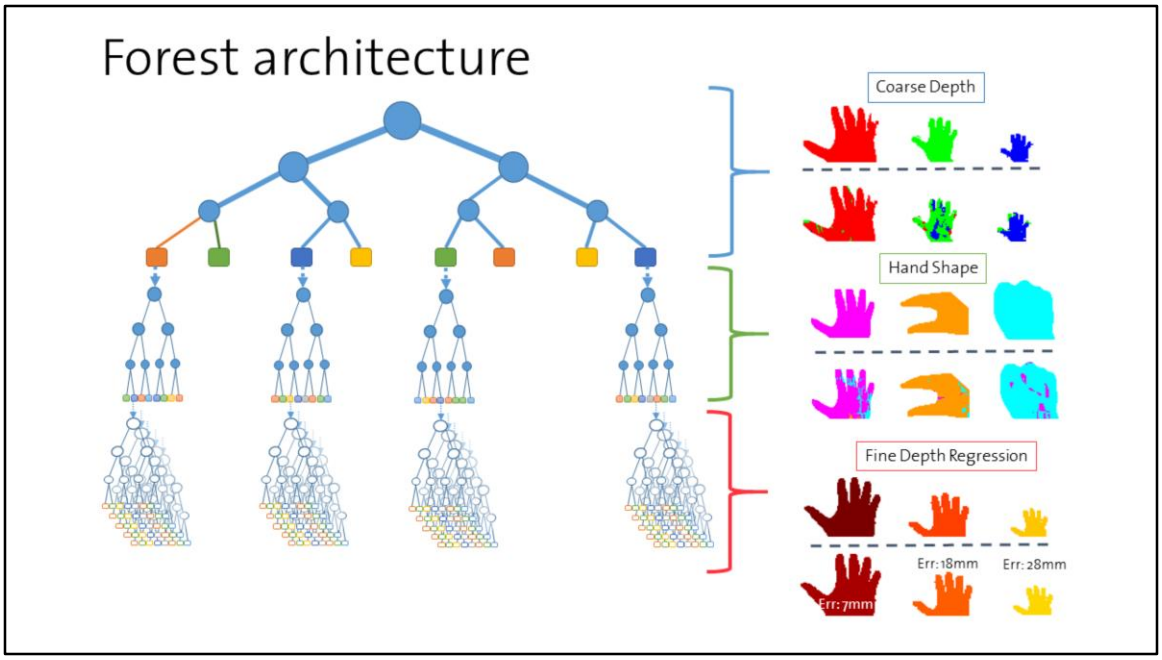


Hence we decided to use learning based depth regression forest to **cover more variations** in terms of appearance change and background noise. For regression tree, each foreground pixel will **traverse down** the tree using a **simple evaluation function at each layer**.

<A>

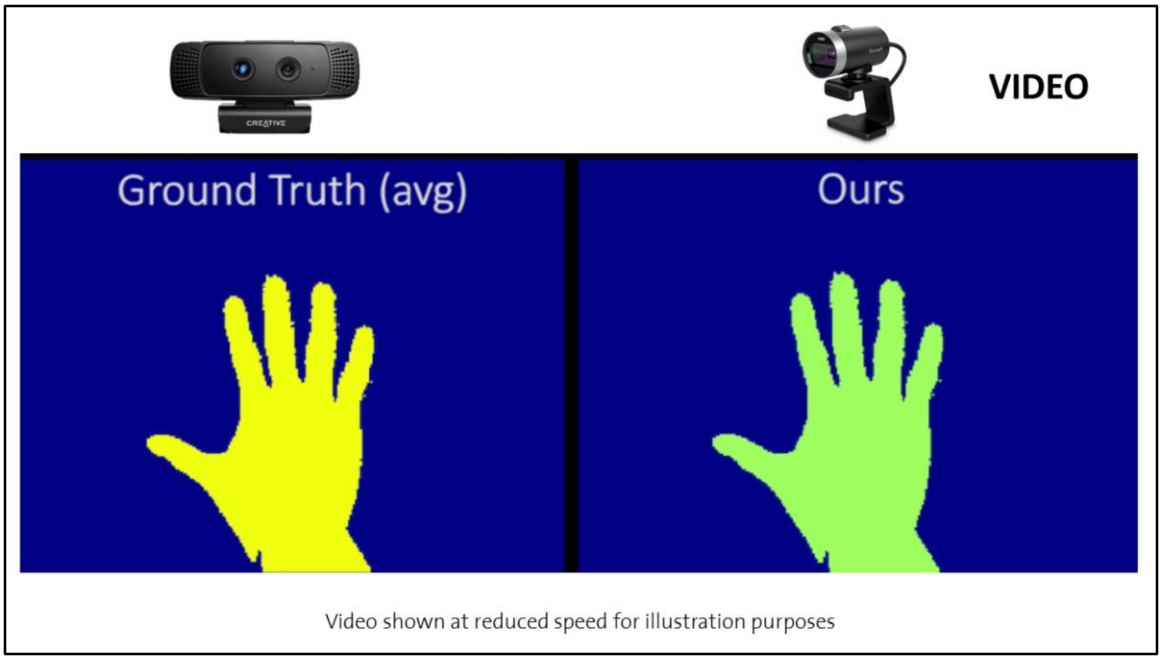
Once the pixel reaches one of the leaf nodes, a **pre learned continuous distribution** will be assigned to that pixel, In our case, **the continuous variable is the metric depth**.

Then, the final depth estimation of the hand is **determined by averaging the estimations** of all the hand pixels.

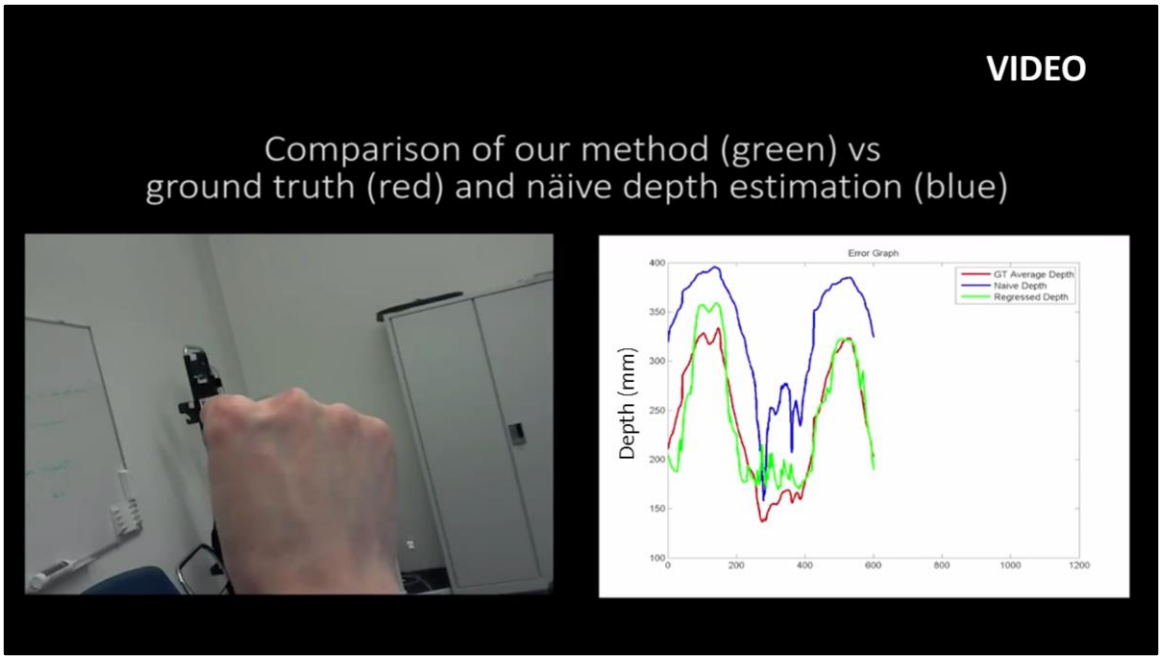


Putting all the pieces together: our whole forest structure works like this:
<A>
Initially, a coarse depth forest splits the hand depth into several discrete intervals.
<A>
Then, the hand shape forest **corresponding to estimated** depth interval will be run to classify the gestures.
<A>
Finally, the depth regression forest **associated with** the classified gesture is executed to estimate the hand depth.
By **splitting** the whole task into multiple easier subtasks, we can achieve good performance using several shallower forests rather than a very deep forest, which saves memory footprint.
This is an important aspect to allow our method to run on **memory constrained** devices.

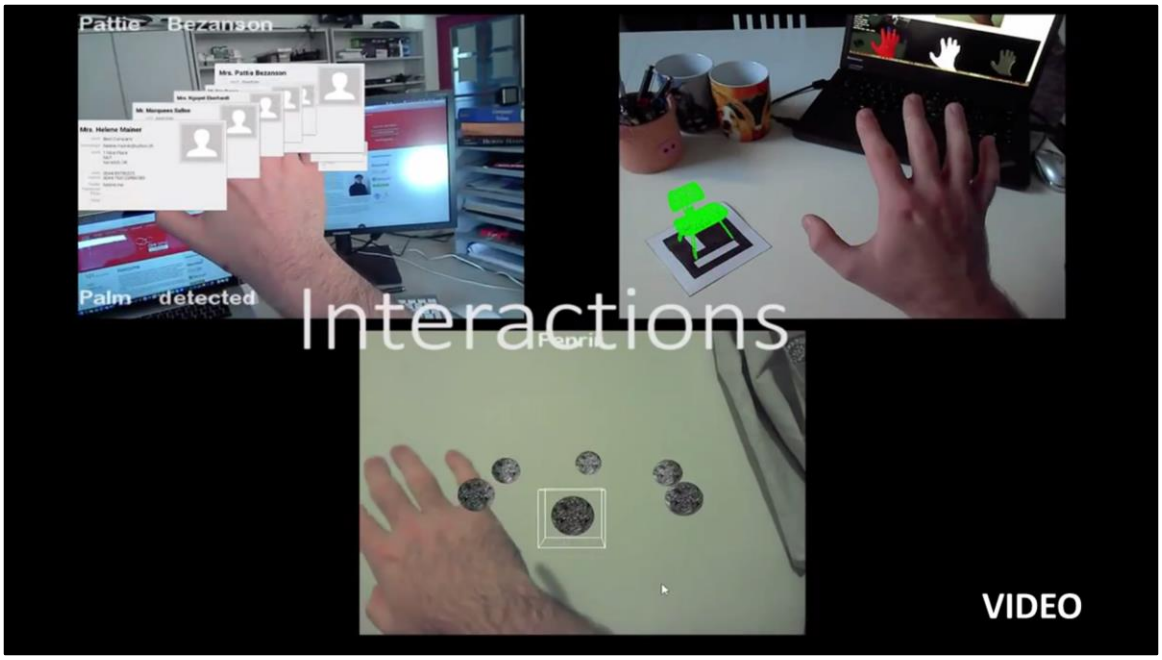
In our implementation we use depth N for the coarse depth forest, depth M for the gesture classification forest, and depth D for the depth regression forest.



Now, moving to the results.
Here is a **qualitative** comparison of our method to ground truth capture.
On the left hand side of the video you can see the average depth obtained from Creative Senz3D depth camera.
On the right hand side of the video is our estimation.
Both maps are shown in **false color-coding**.
Our method **gets relatively similar depth values** to the ground truth.



And here is a **quantitative** comparison.
Ground truth is shown in red, our method in green and then pixel counting method in blue.
our method **closely tracks** the ground truth, also under gesture variation.
Our average estimation error is around 2.8 cm while the naïve way is around 9cm.



Finally, here‘s few other applications to demonstrate our 3D input.
<PLAY>
The first is an AR application.
We can use palm gesture and its depth value to scroll a list of virtual furnitures. And use pinch gesture and its depth value to control the size.

The second is a 3D menu system,
The palm gesture and its depth allows to scroll menu elements. Fist gesture moves through levels.

Limitations

- Depth regression for discrete hand gestures
- Single, average depth value for the whole hand
- Low/no light is problematic

Clearly, there are still some **limitations** in our approach.
First: we can only detect discrete gestures and regress the depth for those gestures.
Second: depth regression provides only **a single, average depth value** for the whole hand. While this is enough for many 3D interaction scenarios, there are cases in which per-pixel depth is more desirable.
Finally, as with other vision based approaches, low or no light **poses a major problem**.

Contributions

- **Joint** hand gesture recognition and 3D hand positions estimation
- Quantitative and qualitative evaluation
- Demonstration by applications

To conclude, we presented a machine learning based approach to jointly recognize gestures and estimate its 3D position.
We haven shown both quantitative and qualitative evaluation of our method, and demonstrated it with application scenarios.

Thanks for you listening.
Do you have any questions?