Monochrome Egocentric Articulated Hand-Tracking for Virtual Reality

НИС МОП Потапов Юрий 202

О чем будет доклад?

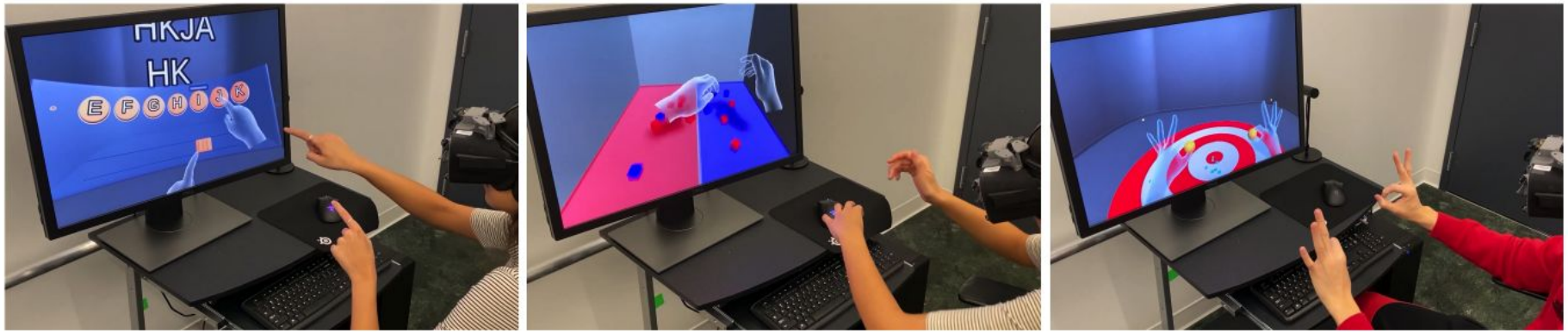
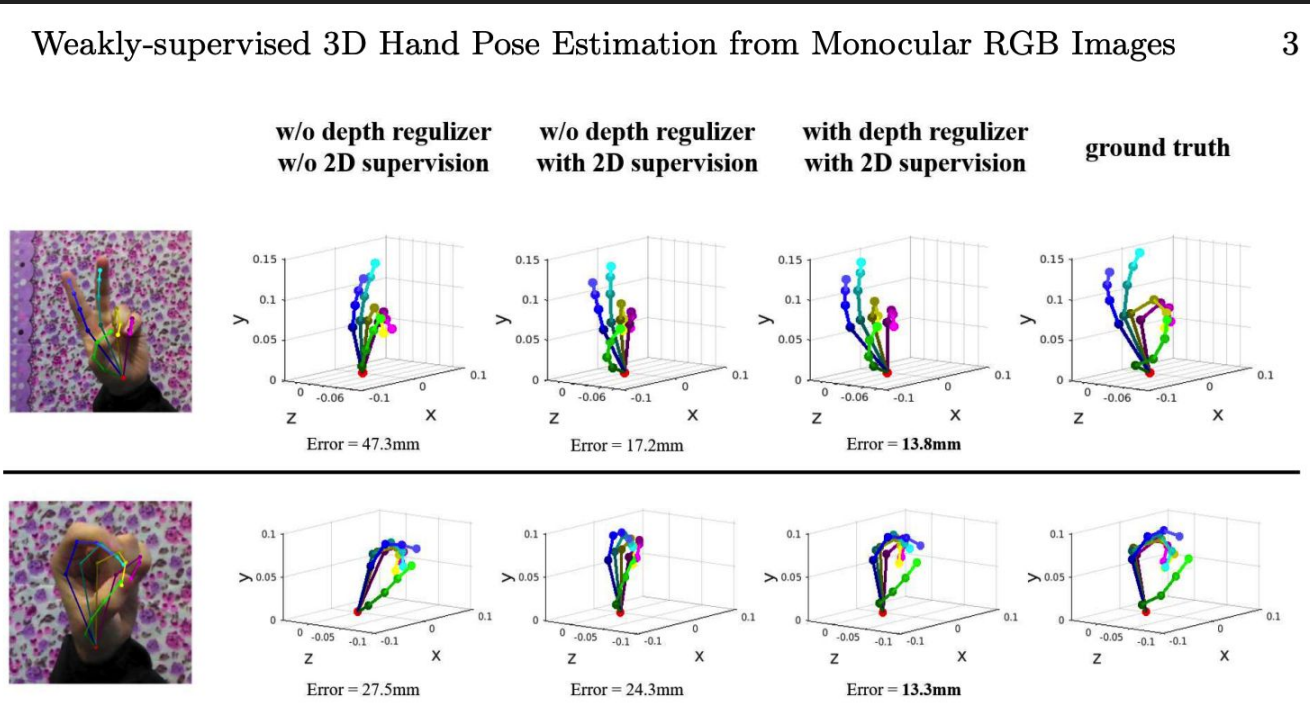


Fig. 1. We present a real-time hand-tracking system using four monochrome cameras mounted on a VR headset. We output the user's skeletal poses and rigged hand model meshes. Here we show some snapshots of users using our system to drive interactive VR experiences.

Что было раньше?

В предыдущих SOTA статьях время преобладают подходы, основанные на глубоком обучении, которые напрямую регрессируют координаты ключевых точек скелета руки по снимкам RGB камеры, или 2.5D камеры.



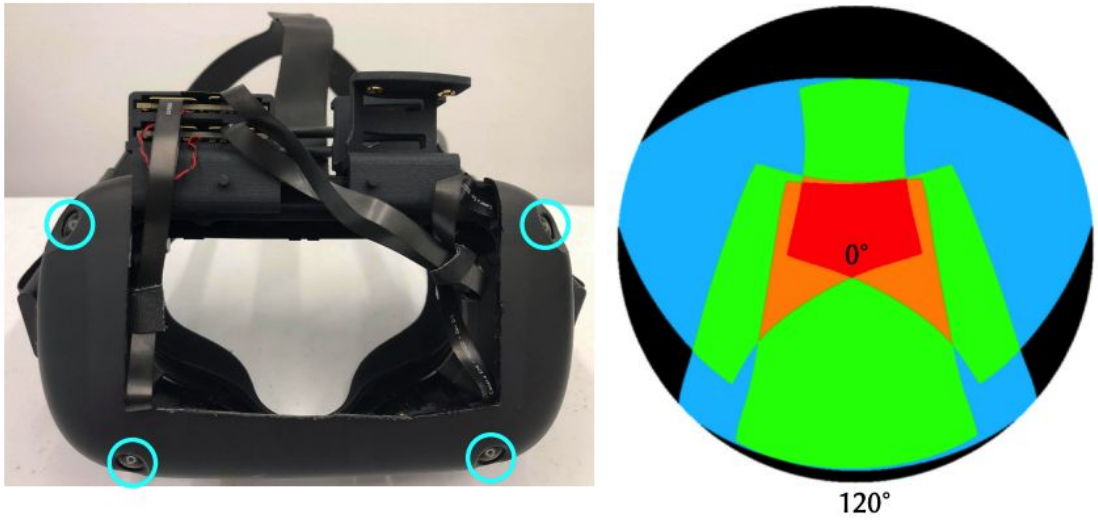


Fig. 3. **Camera configuration.** On the left is a frame holding the 4 monochrome VGA fisheye cameras (circled in blue) that we use for hand-tracking. The frame has a hollow front plate so users can see their hands for data collection purposes. On the right we plot the combined field of view at 50cm distance from the center of the 4 cameras. The angle increases linearly as we move out from the center of the plot. 0° corresponds to the forward-facing (imagine looking forward and extending a ray from the bridge of your nose). We color code the areas by how many cameras can see them: 4(red), 3(orange), 2(green), 1(blue), 0(black).

Почему это отличается от текущей задачи?

Поэтому нужно делать новый датасет, т.к. старые данные неактуальны

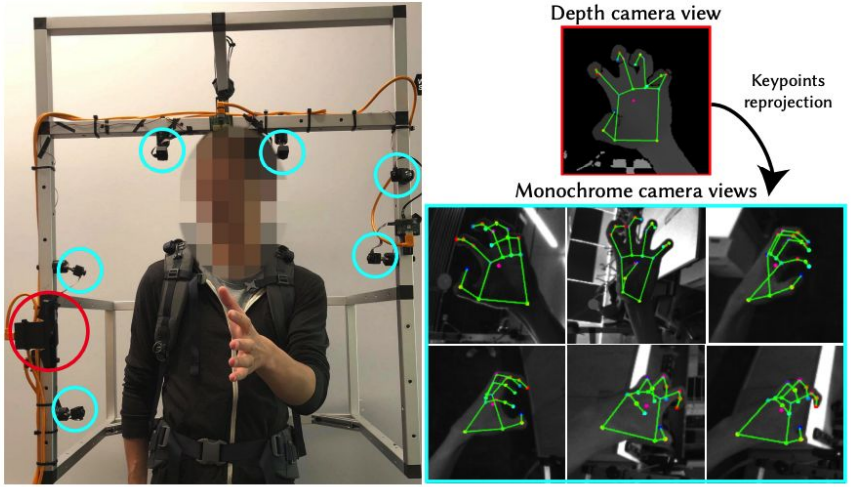


Fig. 5. **Multi-camera system for keypoint annotation.** Left image shows the multi-camera rig with a single depth camera (in red circle) and 6 monochrome cameras (in blue circles). The rig is attached to a backstrap so that user can put it on and walk out to environments with various lightings and backgrounds. Right images show captured frames and generated ground truth. We intentionally place left hand in front of depth camera with minimum occlusion. To this end, the ground truth is generated based on depth camera and projected to other monochrome views.

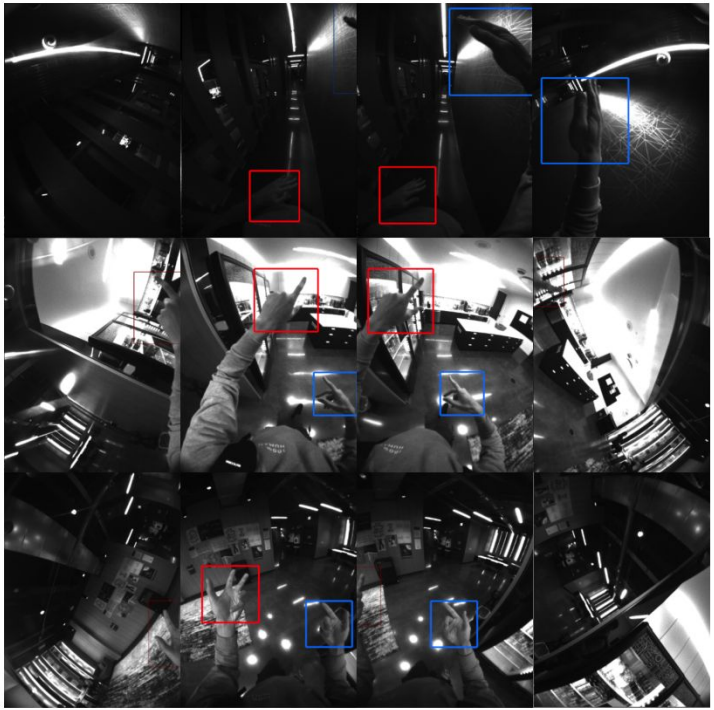
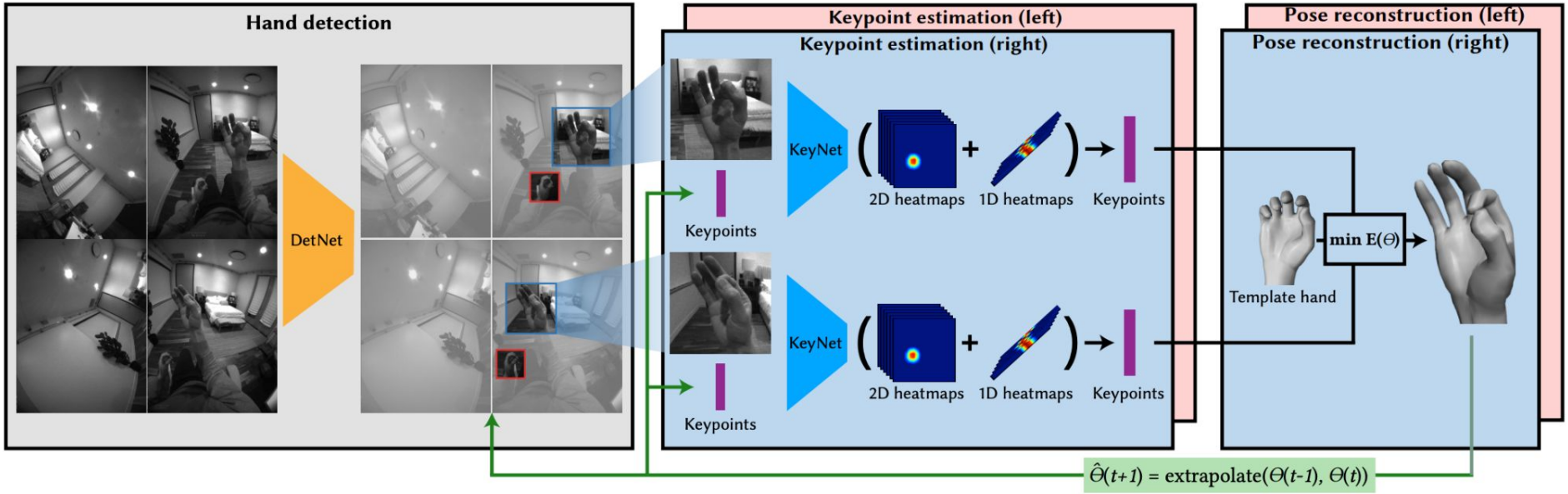
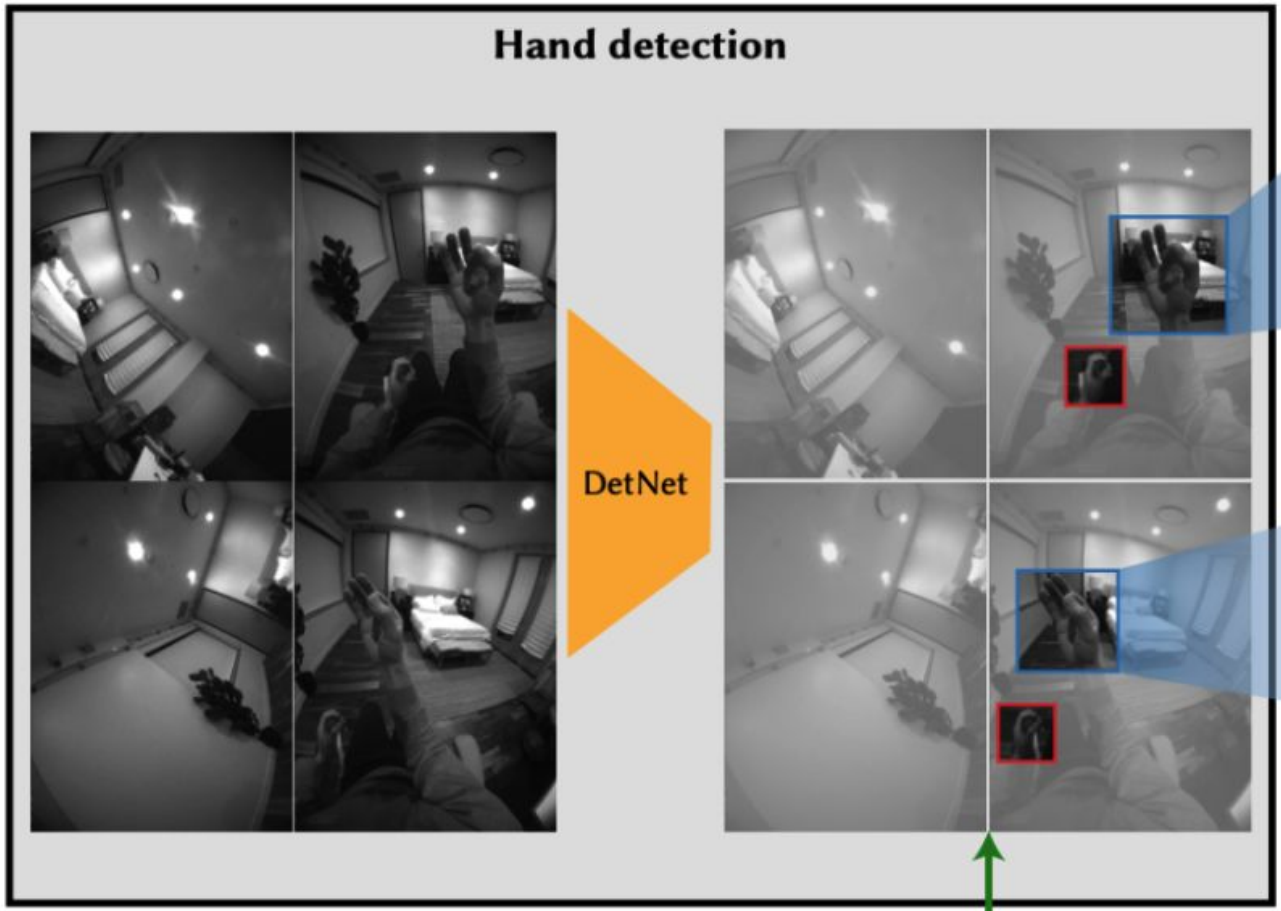


Fig. 6. **Examples of semi-automatically labelled bounding boxes.** Each row corresponds to images from different views captured at the same time. Accurately annotating hand region in images at low light (see Row 1), partially observed hands (see Row 2) and disambiguating left versus right hand is a challenging task for human annotator. On the contrary, our system only needs manual annotation at few frames. Annotation will then propagate to other views and subsequent frames. The examples here are sampled from a sequence of 10k frames where the annotation was done at about 100 bounding boxes per second. Given the capture system is mobile, the sequence features large background and lighting variations.

Пайплайн


Hand detection
DetNet
Keypoint estimation (left)
Keypoint estimation (right)
Keypoints
KeyNet
2D heatmaps
1D heatmaps
Keypoints
Keypoints
KeyNet
2D heatmaps
1D heatmaps
Keypoints
Pose reconstruction (left)
Pose reconstruction (right)
Template hand
min E(Θ)
$\hat{\Theta}(t+1) = \text{extrapolate}(\Theta(t-1), \Theta(t))$

DetNet



Как разметить в полуавтоматическом режиме BBOX-ы для DetNet?

1) аннотируем первые несколько кадров,
2) далее используем обученную KeyNet и наш систему трекинга для обозначения BBOX для следующих кадров, используя KeyPoints предыдущих двух кадров, получая новые KeyPoints, которые можно получить с помощью равенства: $\hat{\Theta}(t+1) = \text{extrapolate}(\Theta(t-1), \Theta(t))$, $\hat{\theta}_t = 2\theta_{t-1} - \theta_{t-2}$
3) если толокер увидел несоответствие, то аннотируем кадр, на котором мы ошиблись и заново запускаем пайплайн

DetNet

$$L = \sum_{i \in \{\text{left}, \text{right}\}} L_{\text{loc},i} + \lambda L_{\text{conf},i} \ ,$$

where $L_{\text{loc},i}$ supervises the bounding circles using an MSE loss, $L_{\text{conf},i}$ supervises the hand confidence loss using the standard binary cross entropy loss, and the coefficient λ balances the contribution of the two terms ($\lambda = 100$ when we train DetNet).

Table 4. Architecture for DetNet-F

Stage/Output	Input	Operator	Exp size	Out Channels	Stride	Repeat
Backbone	$1 \times 640 \times 480$	AvgPool2d 4×4	-	1	4	1
	$1 \times 160 \times 120$	Conv2d 3×3, BN, Relu	-	32	2	1
	$32 \times 80 \times 60$	IRB 3×3	96	32	2	1
	$32 \times 40 \times 30$	IRB 3×3	96	32	1	1
	$32 \times 40 \times 30$	IRB 3×3	192	64	2	1
	$64 \times 20 \times 15$	IRB 3×3	384	64	1	2
	$64 \times 20 \times 15$	IRB 3×3	384	64	2	1
	$64 \times 10 \times 8$	IRB 3×3	384	64	1	3
	$64 \times 10 \times 8$	IRB 3×3	384	96	1	1
	$96 \times 10 \times 8$	IRB 3×3	576	96	1	2
	$96 \times 10 \times 8$	IRB 3×3	576	128	2	1
	$128 \times 5 \times 4$	IRB 3×3	768	128	1	2
$160 \times 5 \times 4$	$128 \times 5 \times 4$	IRB 3×3	768	160	1	1
Hand center	$160 \times 5 \times 4$	Conv2d 1×1, BN	-	4	1	1
	$4 \times 5 \times 4$	AvgPool2d 5×4	-	4	1	1
2×2	$4 \times 1 \times 1$	Reshape 2×2	-	1	-	1
Hand radius	$160 \times 5 \times 4$	Conv2d 1×1, BN	-	2	1	1
$2 \times 1 \times 1$	$2 \times 5 \times 4$	AvgPool2d 5×4	-	2	1	1
Hand cls	$160 \times 5 \times 4$	Conv2d 1×1, BN	-	2	1	1
	$2 \times 5 \times 4$	AvgPool2d 5×4	-	2	1	1
$2 \times 1 \times 1$	$2 \times 1 \times 1$	Sigmoid	-	2	-	1

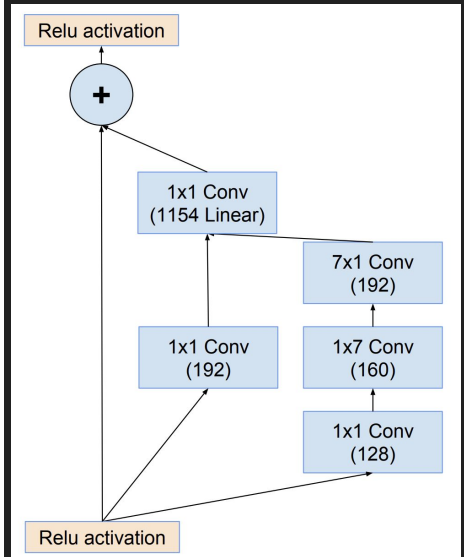
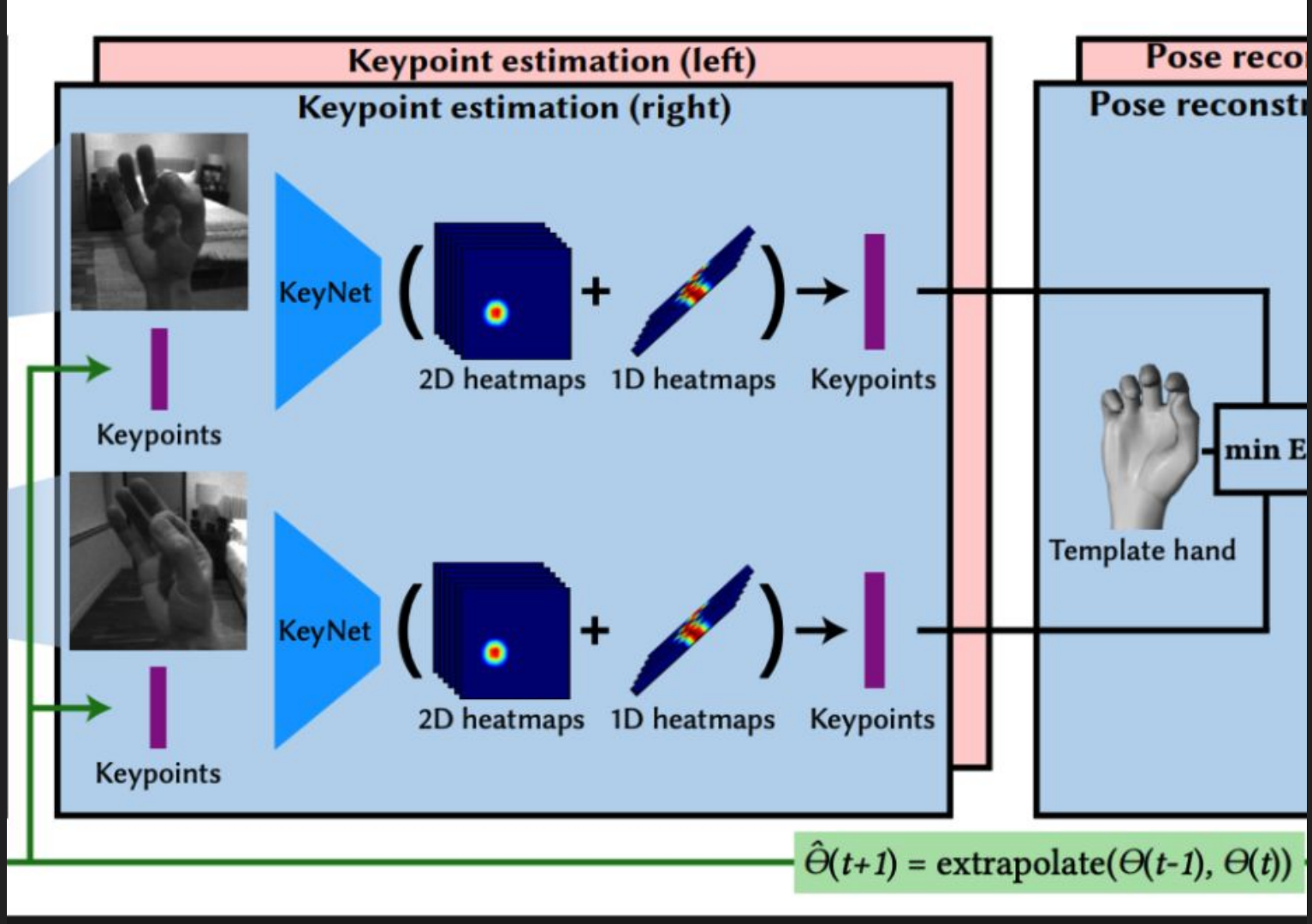
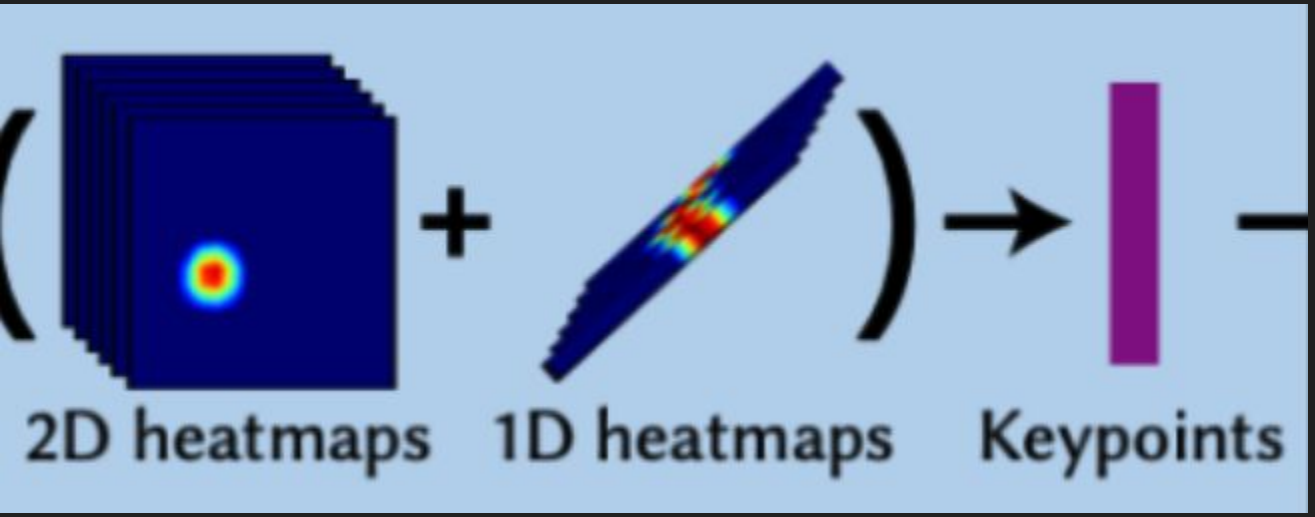


Figure 17. The schema for 17×17 grid (Inception-ResNet-B) module of the Inception-ResNet-v2 network.

KeyNet


Keypoint estimation (left)
Keypoint estimation (right)
Pose reco
Pose reconstr
KeyNet
2D heatmaps
1D heatmaps
Keypoints
Keypoints
KeyNet
2D heatmaps
1D heatmaps
Keypoints
Keypoints
Template hand
min E
$\hat{\Theta}(t+1) = \text{extrapolate}(\Theta(t-1), \Theta(t))$

2D heatmap & 1D heatmap



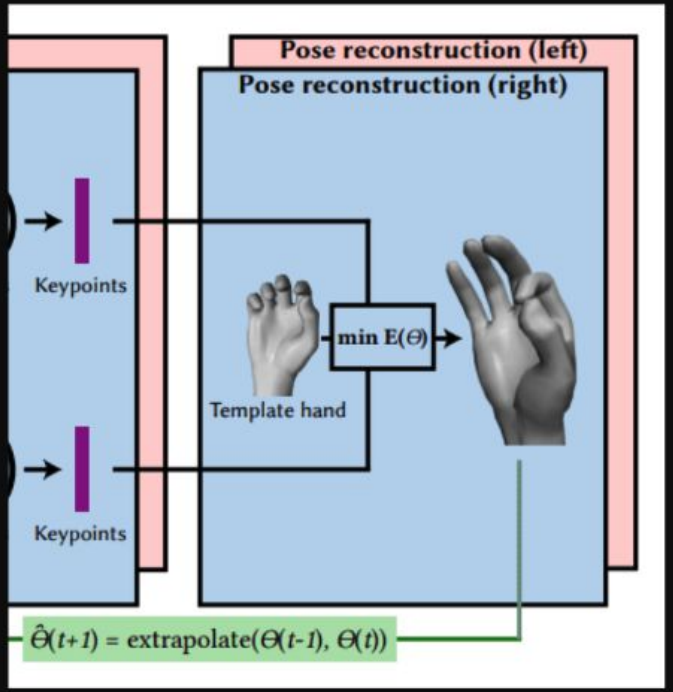
Нейросеть выдает 21 2D heatmap’у – уверенности в том, что каждый пиксель является соответствующей ключевой точкой

И также 21 1D heatmap-ы – показывает наиболее вероятное расстояние каждой точки до камеры

Table 5. Architecture for KeyNet-F

Stage/Output	Input	Operator	Exp size	Out Channels	Stride	n
Backbone image	1×96^2	Conv2d 3×3, BN, Relu	-	32	2	1
	32×48^2	IRB 3×3	1	32	1	1
	32×48^2	IRB 3×3	96	32	2	1
	32×24^2	IRB 3×3	192	32	1	1
	32×24^2	IRB 3×3	192	64	2	1
64×12^2	64×12^2	IRB 3×3	384	64	1	2
Backbone keypoints	63	Linear, Relu	-	4608	-	1
32×12^2	4608	Reshape 12×12	-	32	-	1
Backbone fused	$64 \times 12^2, 32 \times 12^2$	Concat	-	96	-	1
	96×12^2	IRB 3×3	384	64	1	2
	64×12^2	IRB 3×3	384	64	1	3
	64×12^2	IRB 3×3	384	96	1	1
	96×12^2	IRB 3×3	480	96	1	2
	96×12^2	IRB 3×3	576	128	2	1
	128×6^2	IRB 3×3	768	128	1	2
128×6^2	128×6^2	IRB 3×3	768	160	1	1
Keypoint heatmap	160×6^2	Conv2d 3×3 pad2, BN, Relu	-	63	1	1
	63×8^2	ConvTranspose2d 2×2	-	42	2	1
21×18^2	42×16^2	Conv2d 3×3 pad2, BN, Relu	-	21	1	1
Keypoint distance	160×6^2	AvgPool2d 6×6	-	160	6	1
	160×1^2	Conv2d 1×1, Relu	-	378	1	1
21×18	378×1^2	Reshape 18	-	21	-	1

Pose Reconstruction



3.5 Model based pose estimation

Once we have obtained the 21 3D keypoints of the hand, we solve for the pose of the hand,

$$\theta = \min_{\theta}(E_{2D} + w_1 E_{dist} + w_2 E_{temporal}).$$

The 2D error term E_{2D} enforces agreement with the detected 2D keypoints,

$$E_{2D} = \sum_{i,j} \|\Pi_j(p_i(\theta)) - \hat{p}_{i,j}\|_2^2,$$

where Π_j is the function that projects a point in 3D space to the jth camera's image space and $\hat{p}_{i,j}$ is the ith predicted keypoint in the jth camera's image space.

The 1D error term for the relative distance is,

$$E_{dist} = \sum_{i,j} ||(dist_j(p_i(\theta)) - dist_j(p_0(\theta)) - \phi \cdot (\hat{d}_{i,j}^{rel} - \hat{d}_{0,j}^{rel})||_2^2,$$

where $dist_j$ is a function that computes the distance between a point to the jth camera and $\hat{d}_{i,j}^{rel}$ is the predicted relative distance coordinate for the ith keypoint in the jth camera.

The temporal term is to ensure smoothness of the tracked hand poses

$$E_{temporal} = ||\theta - \theta_{t-1}||_2^2$$

θ_{t-1} is the hand pose from the previous frame when available. Note

Детали обучения

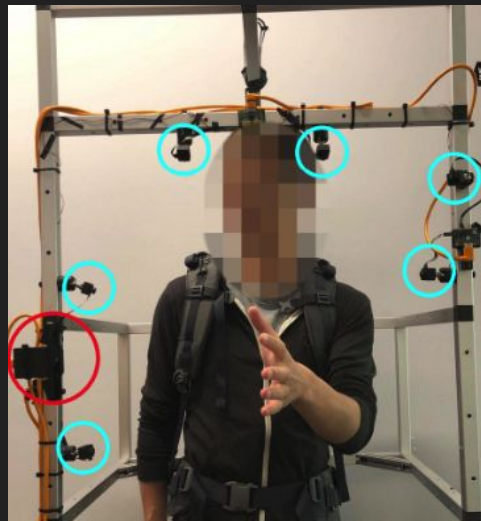
2.6 миллионов картинок для DetNet

пайплайн построен для одной руки, для другой достаточно его отразить

работает 60hz на 1080ti, и на видеоядре Snapdragon 835 в 30hz

для итоговой нейросети не понадобился сетап с картинки – данные брались прямо с 4-х камер

т.к. руки у всех разного размера, необходима калибровка в начале использования, она занимает 5с и используется в конструировании E_2D loss-а



Результаты

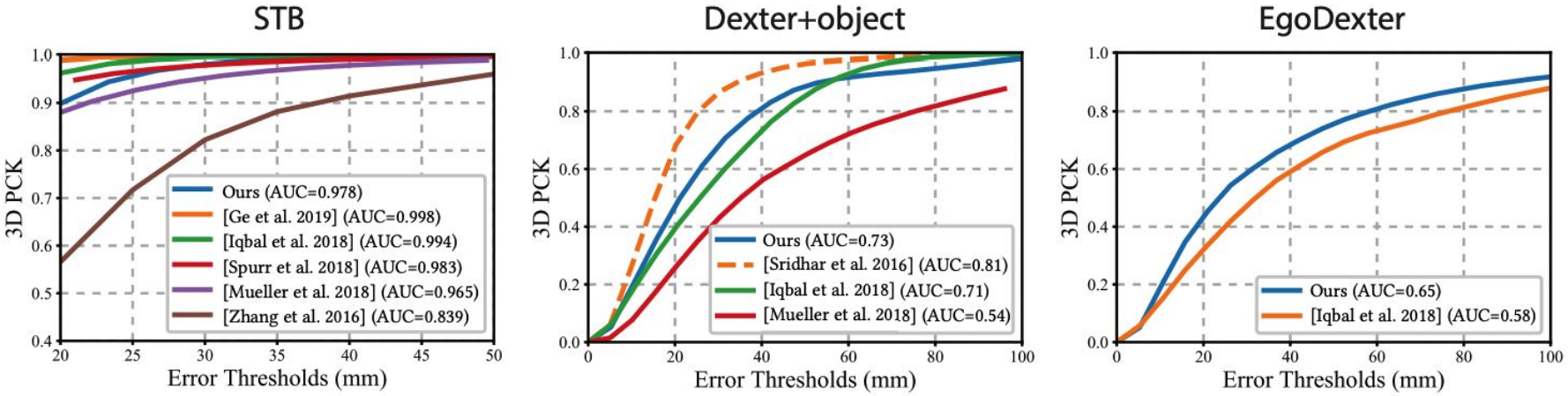


Fig. 11. **PCK plots on public benchmarks.** From left to right, we show the percentage of correct keypoints with respect to different error thresholds across all frames on STB[Zhang et al. 2016], Dexter+object[Mueller et al. 2017] and EgoDexter[Sridhar et al. 2016] benchmarks. The AUC is shown in the legend. The depth-based method [Sridhar et al. 2016] is drawn with dashed line in the middle plot.

(датасеты только из цветных картинок, поэтому картинки сделали серыми и мануально обучили DetNet на поиск BBox для такого формата данных)

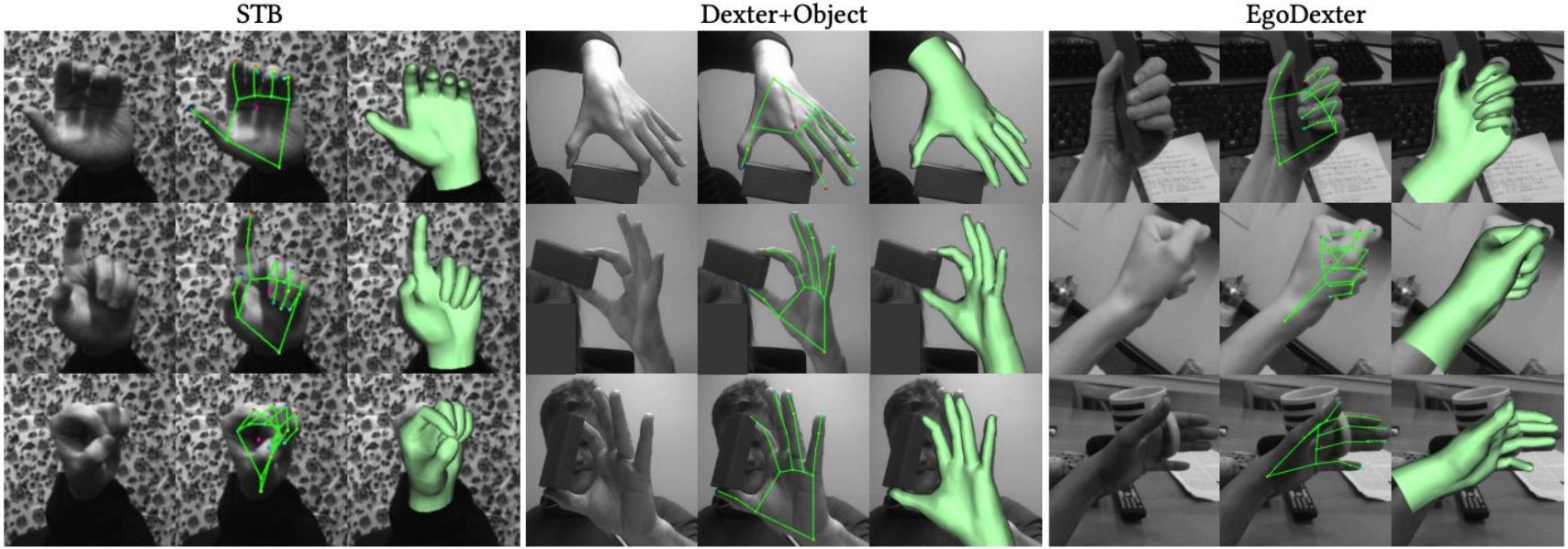


Fig. 12. Tracking results using our system on examples from STB, Dexter+Object and EgoDexter datasets.

Спасибо за внимание!

вопросы?