PAPER PRESENTATION

蔡侑軒

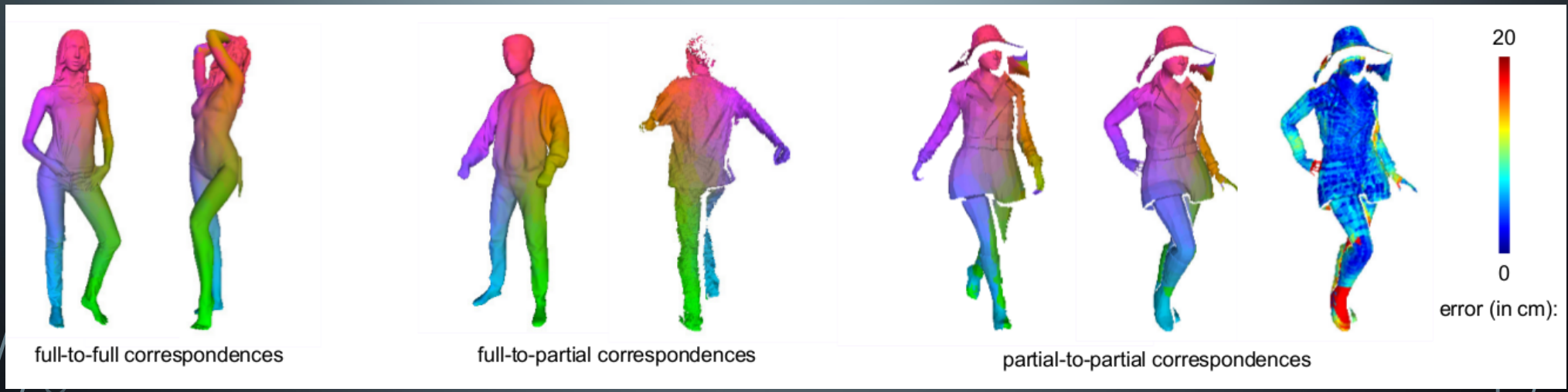
CARAMEL YO

PAPER

- Dense Human Body Correspondences Using Convolutional Networks
- CVPR 2016

PROBLEMS TO SOLVE

- Find the dense correspondences between 3D sc ans of people



CONTRIBUTIONS

- Require partial geometric information in the for m of two depth maps or partial reconstructed su rfaces only
- Work for humans in many poses and wearing m any clothing
- The method mentioned in this paper does not re quire the two people to be scanned from similar viewpoints, and runs in real time

BASIC IDEA

- Formulate the correspondence problem as a cla ssification problem

IMPLEMENTATION

- Train a feature descriptor on depth map pixels, a nd then train it to solve a body region classificati on problem

IMPLEMENTATION

- We desire the feature vector to saticfy two prope rties:
  - F depends only on the pixels location on the human body
  - ||f(p) – f(q)|| is small when p and q represent nearby points on the human body

IMPLEMENTATION

- Indirect methods optimize the network architect ure to perform classification (basic idea in this paper)

IMPLEMENTATION

- The network consists of a descriptor extraction tower and a classfication layer
- Peel off the classification layer after training

IMPLEMENTATION

- Classification networks tend to assign similar (dissimilar) descriptors belonging to the same (different) class
- Satisfy the above properties implicitly
- Computational efficiency

IMPLEMENTATION

- Between different human models, it is only possible to obtain a sparse set of key point correspondences, while for different poses of the same person, we may have dense pixel-wise correspondences
- Classification network treats all classes equally

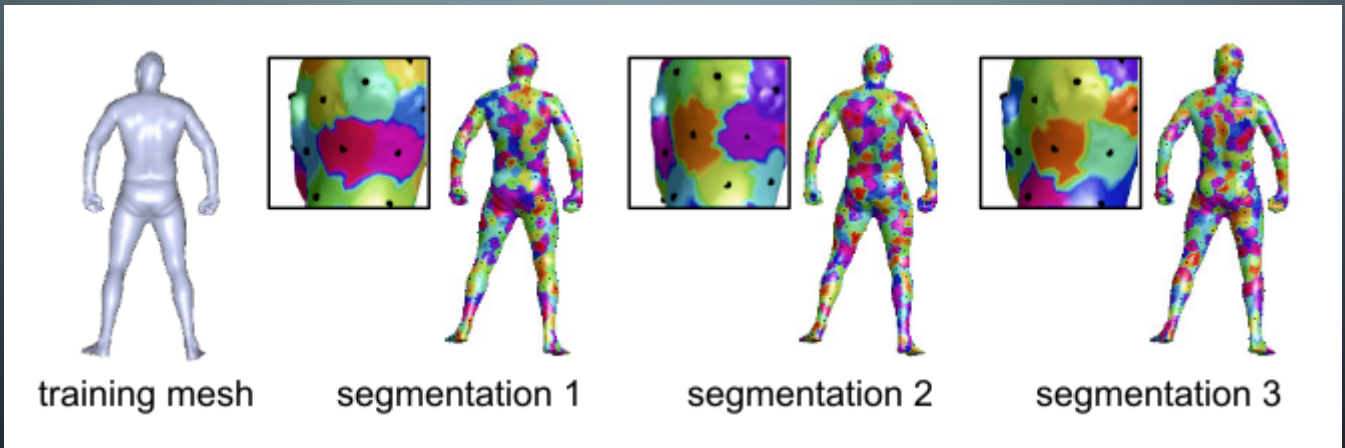
IMPLEMENTATION

- Learn per-pixel descriptors for depth images to solve a group of classification problems
- Use a single feature extraction tower shared by the dif ferent classification tasks

$$\{\mathbf{w}_i^\star\}, \mathbf{w}^\star = \arg\min_{\{\mathbf{w}_i\}, \mathbf{w}} \sum_{i=1}^{M} l(\mathbf{w}_i, \mathbf{w}).$$

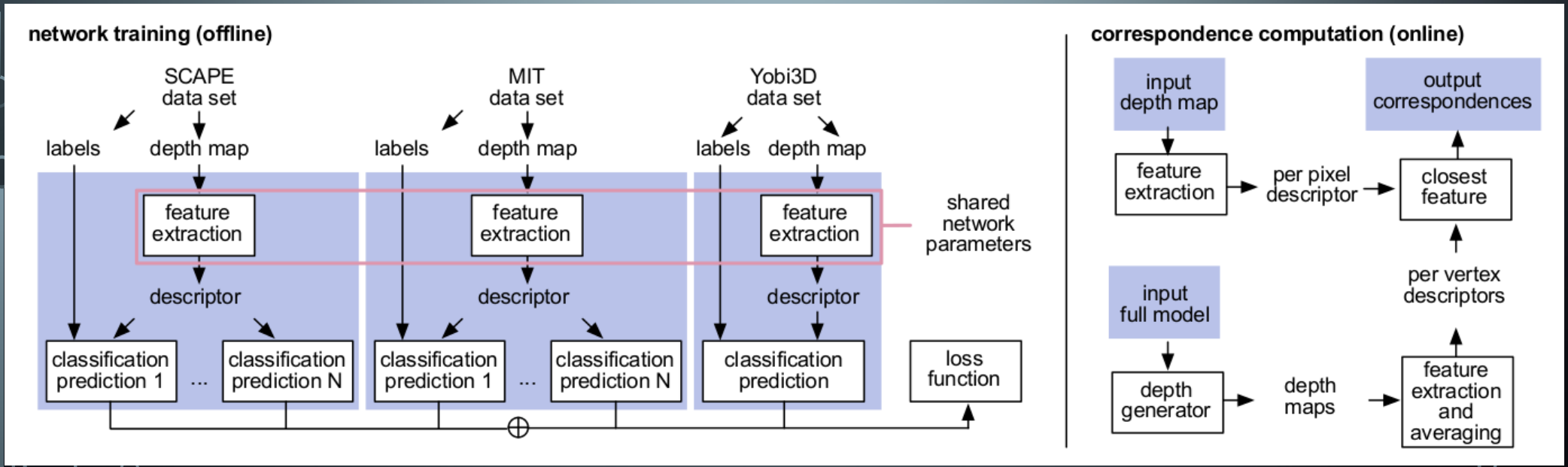
IMPLEMENTATION

- Two classification tasks
  - Classify key points, used for inter-subject training
  - Classify dense pixel-wise labels by segmenting models into p atches, used for intra-subject training



IMPLEMENTATION

- For full or partial 3D scans, we compute a per-vertex f eature descriptor by averaging the per-pixel descripto rs of the depth maps



DATASET

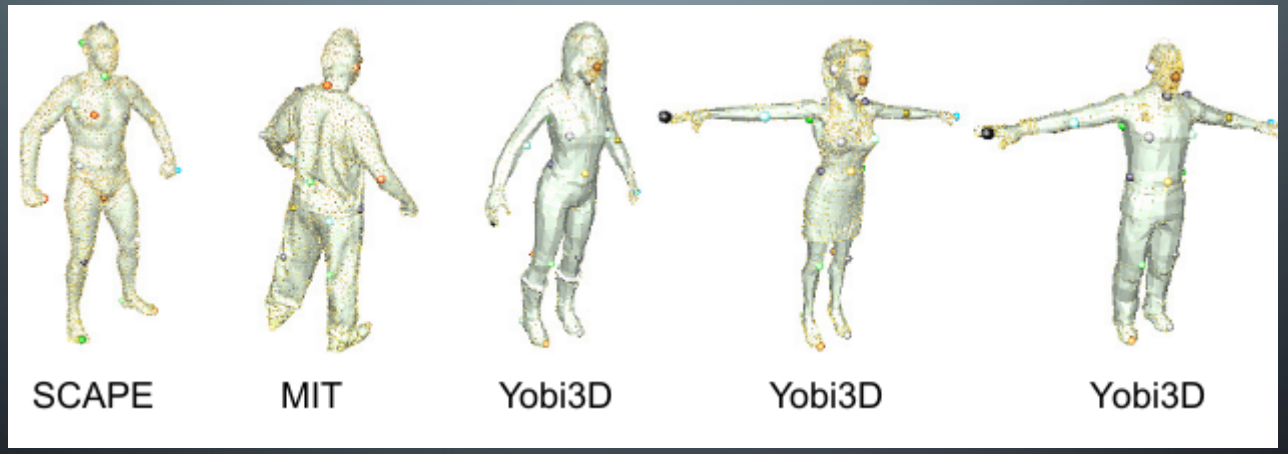
- SCAPE
  - 71 registered meshes of one person in different poses
- MIT
  - Animation sequences of three different characters
- Yobi3D
  - A diverse set of 2000 digital characters with varying clothing

DATASET

- Yobi3D dataset covers the shape variability in lo cal geometry, while the SCAPE and MIT datasets cover the variability in pose

DETAILS

- Key point annotations
- For shapes in the SCAPE and MIT datasets, we o nly annotate one rest-shape and use the ground-truth correspondences to propagate annotation s

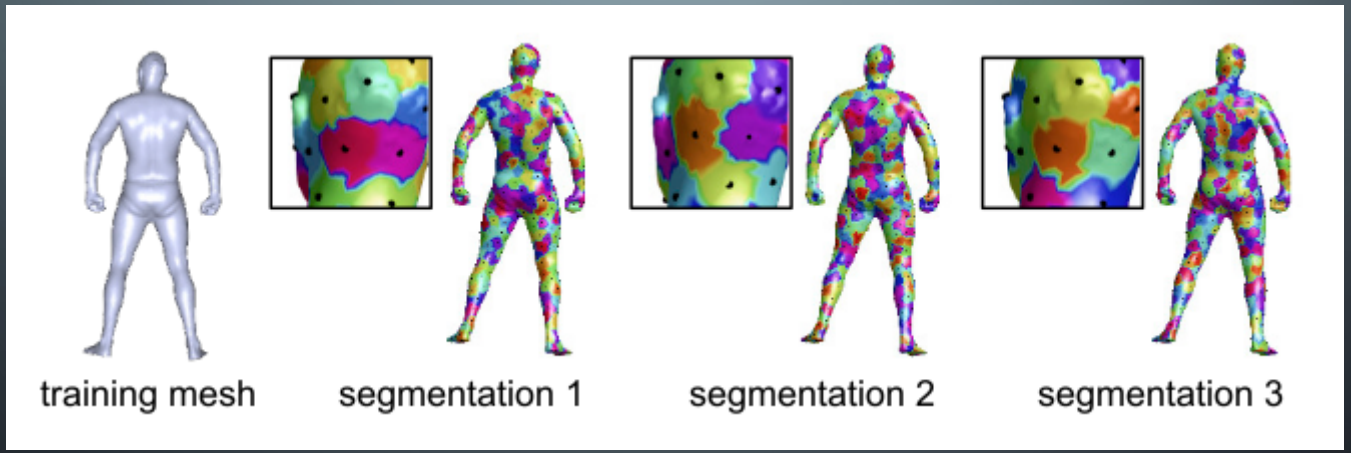


DETAILS

- 500-patch segmentation generation
- Each segmentation is generated by randomly pi cking 10 points on each model, and then adding the remaining points via furthest point-samplin g
- In total we use 100 pre-computed segmentation s

DETAILS

- 500-patch segmentation generation
- Each such segmentation provides 500 classes fo r depth scans of the same person(with different poses)



NETWORK

- The descriptor extraction tower takes a depth im age as input and extracts for each pixel a dimens ion d(d = 16 in this paper) descriptor vector

	0	1	2	3	4	5	6	7	8	9	10
layer	**image**	**conv**	**max**	**conv**	**max**	**2×conv**	**conv**	**max**	**2×conv**	**int**	**conv**
filter-stride	-	11-4	3-2	5-1	3-2	3-1	3-1	3-2	1-1	-	3-1
channel	1	96	96	256	256	384	256	256	4096	4096	16
activation	-	relu	lrn	relu	lrn	relu	relu	idn	relu	idn	relu
size	512	128	64	64	32	32	32	16	16	128	512
num	1	1	4	4	16	16	16	64	64	1	1

NETWORK

- The downsampling not only makes the computations faster and more memory efficient, but also removes salt-and-pepper noise

	0	1	2	3	4	5	6	7	8	9	10
layer	**image**	**conv**	**max**	**conv**	**max**	**2×conv**	**conv**	**max**	**2×conv**	**int**	**conv**
filter-stride	-	11-4	3-2	5-1	3-2	3-1	3-1	3-2	1-1	-	3-1
channel	1	96	96	256	256	384	256	256	4096	4096	16
activation	-	relu	lrn	relu	lrn	relu	relu	idn	relu	idn	relu
size	512	128	64	64	32	32	32	16	16	128	512
num	1	1	4	4	16	16	16	64	64	1	1

NETWORK

- The upsampling implicitly performs linear smoo thing between the descriptors of neighboring pi xels

	0	1	2	3	4	5	6	7	8	9	10
layer	**image**	**conv**	**max**	**conv**	**max**	**2×conv**	**conv**	**max**	**2×conv**	**int**	**conv**
filter-stride	-	11-4	3-2	5-1	3-2	3-1	3-1	3-2	1-1	-	3-1
channel	1	96	96	256	256	384	256	256	4096	4096	16
activation	-	relu	lrn	relu	lrn	relu	relu	idn	relu	idn	relu
size	512	128	64	64	32	32	32	16	16	128	512
num	1	1	4	4	16	16	16	64	64	1	1

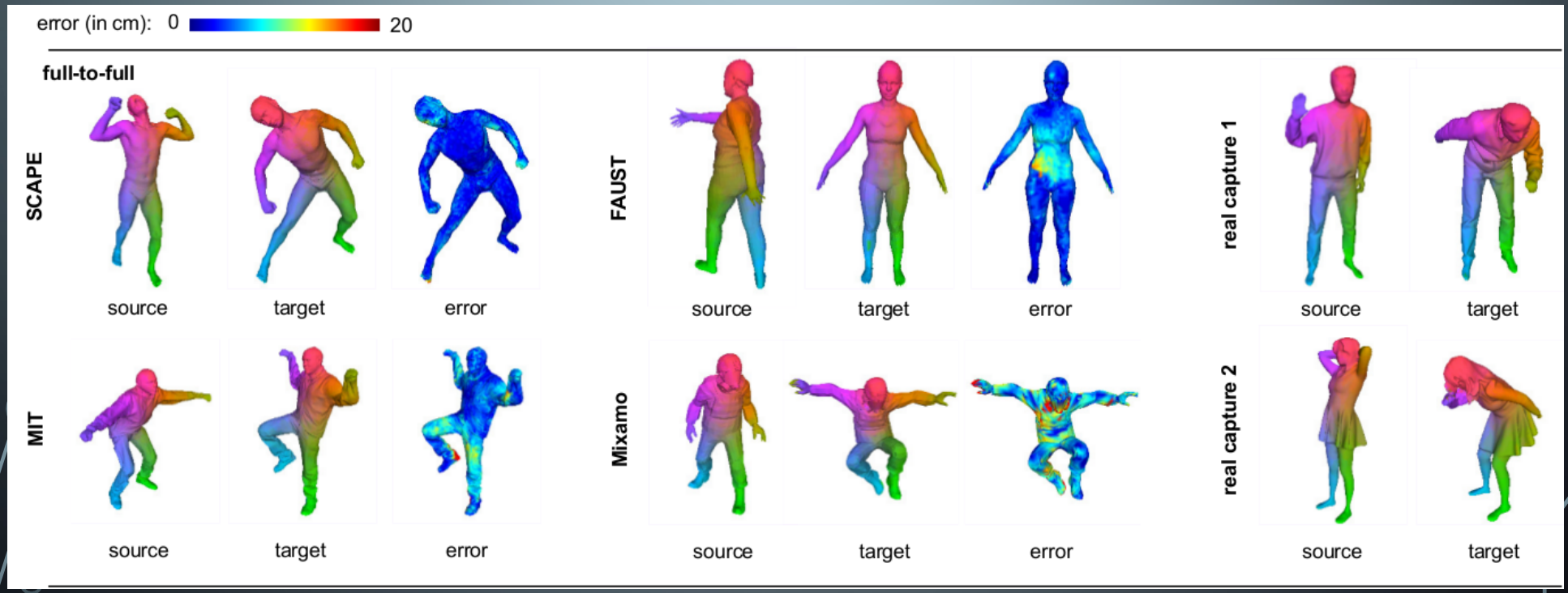
NETWORK

- We introduce one layer for each segmentation of each person in the SCAPE and MIT datasets and one shared layer for all the key points
- Employ softmax as loss function

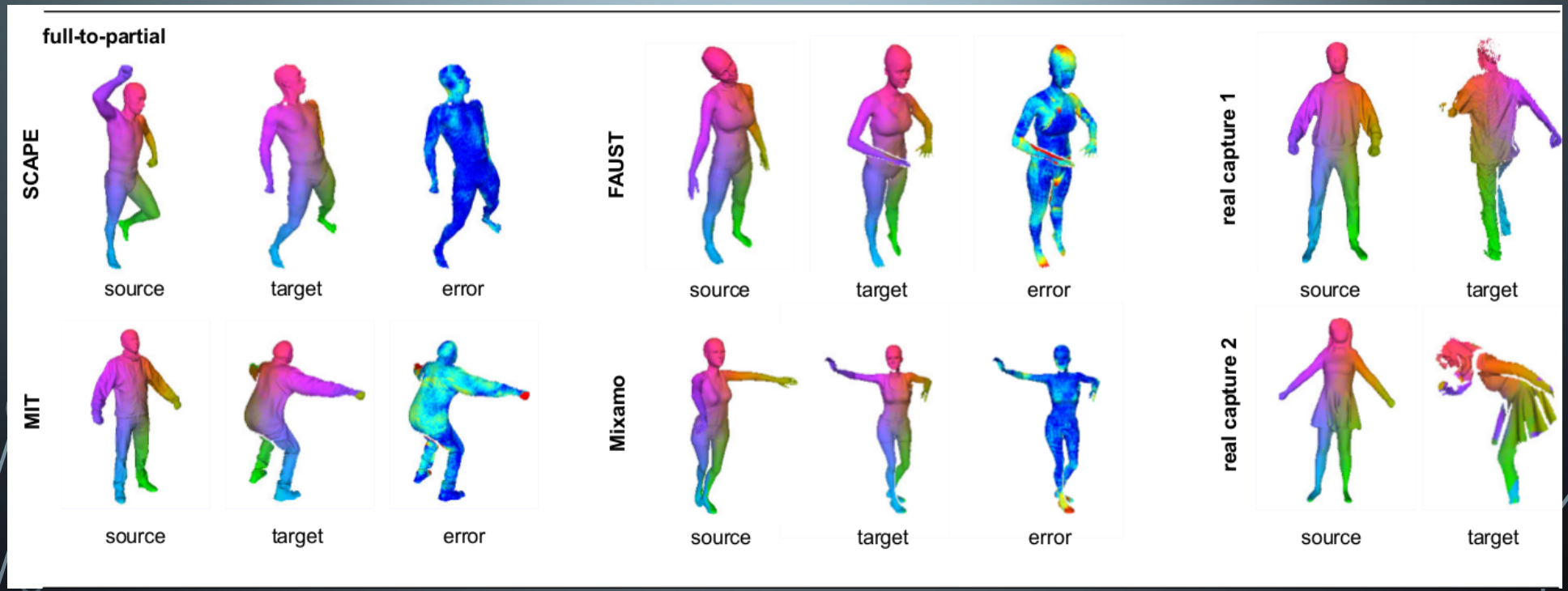
TRAINING

- Randomly pick a task(key points or dense label s) for a random partial scan and feed it into the network for training
- If the task is dense labels, we also randomly pick a segmentation among all possible segmentatio ns

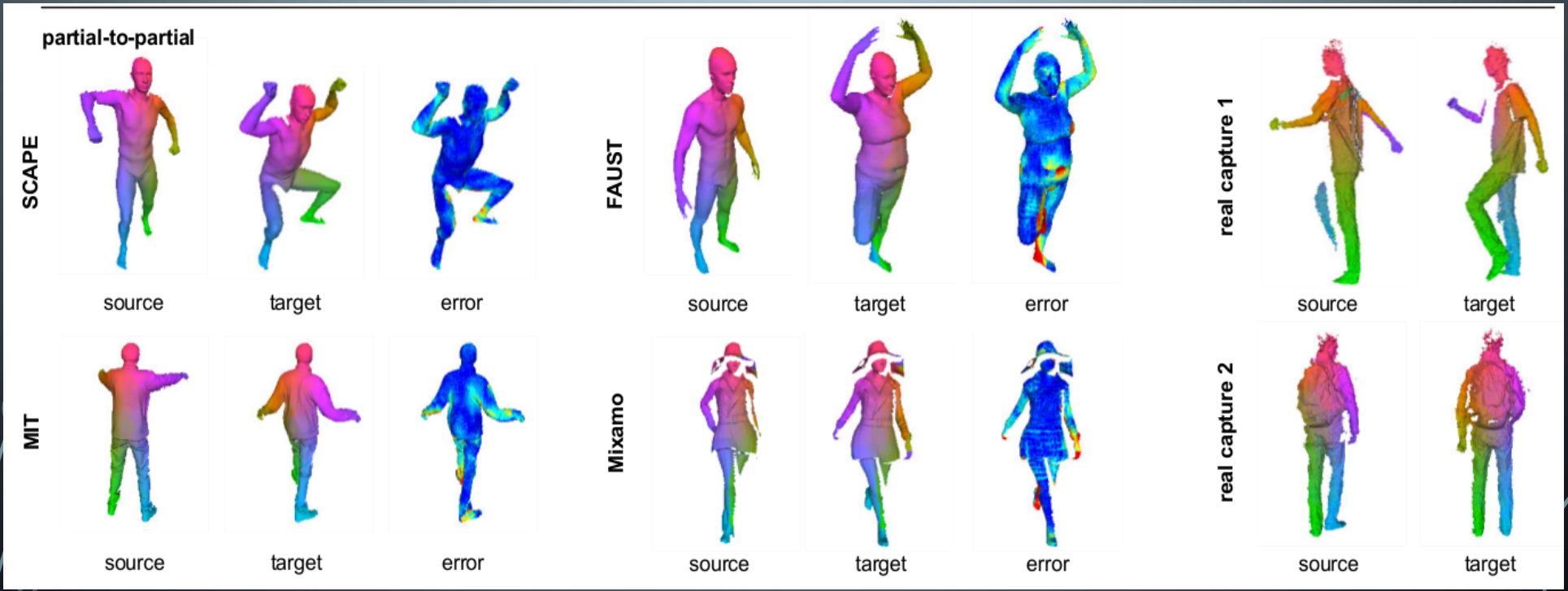
RESULT


error (in cm): 0 20
full-to-full
SCAPE
source
target
error
FAUST
source
target
error
real capture 1
source
target
MIT
source
target
error
Mixamo
source
target
error
real capture 2
source
target

RESULT


full-to-partial
SCAPE
source
target
error
FAUST
source
target
error
real capture 1
source
target
MIT
source
target
error
Mixamo
source
target
error
real capture 2
source
target

RESULT


partial-to-partial
SCAPE
source
target
error
FAUST
source
target
error
real capture 1
source
target
MIT
source
target
error
Mixamo
source
target
error
real capture 2
source
target

LIMITATION

- Supervised learning problems

Any questions?

Good Luck!