APES: Articulated Part Extraction from Sprite Sheets

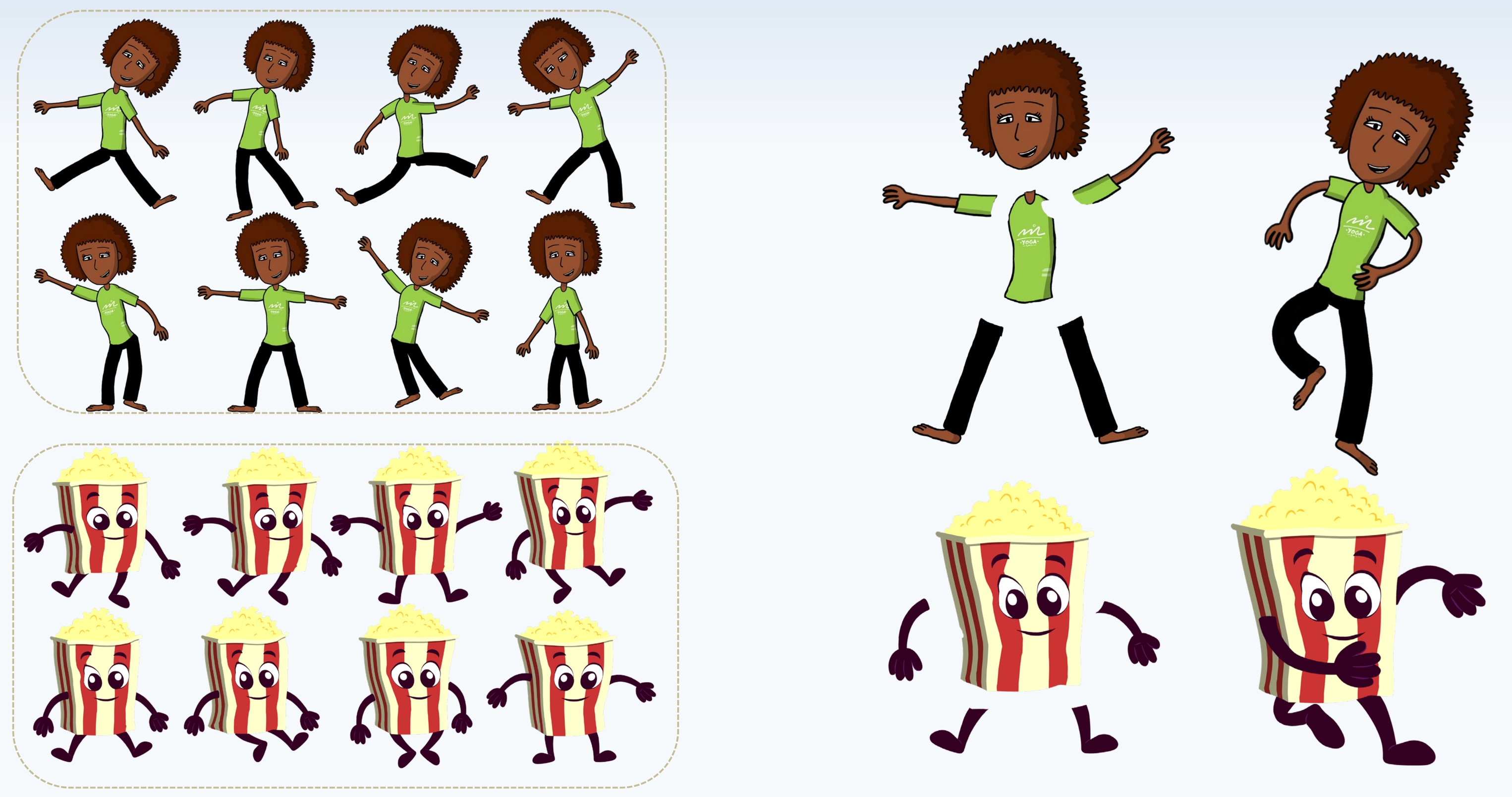
Zhan Xu[1 2] Matthew Fisher[2] Yang Zhou[2] Deepali Aneja[2] Rushikesh Dudhat[1] Li Yi[3] Evangelos Kalogerakis[1]

[1]University of Massachusetts Amherst [2]Adobe Research [3]Tsinghua University

CVPR JUNE 19-24 2022 NEW ORLEANS • LOUISIANA

Overview

Motivation: Discover articulated parts from input 2D characters drawn in various poses. Resulting parts can be used for animation and puppet creation.



Input Sprite sheet Output Parts & New poses

Challenges: Characters can have a wide range of different structure, which prevents a single template from working across all characters; limited available data of rigged, animated characters; poses shown in sprite sheets have both articulated variation and non-rigid deformation.

Earlier work: Prior image co-part segmentation methods pretrain networks with natural images, fail to capture large pose variation, and assume fixed articulation structure.

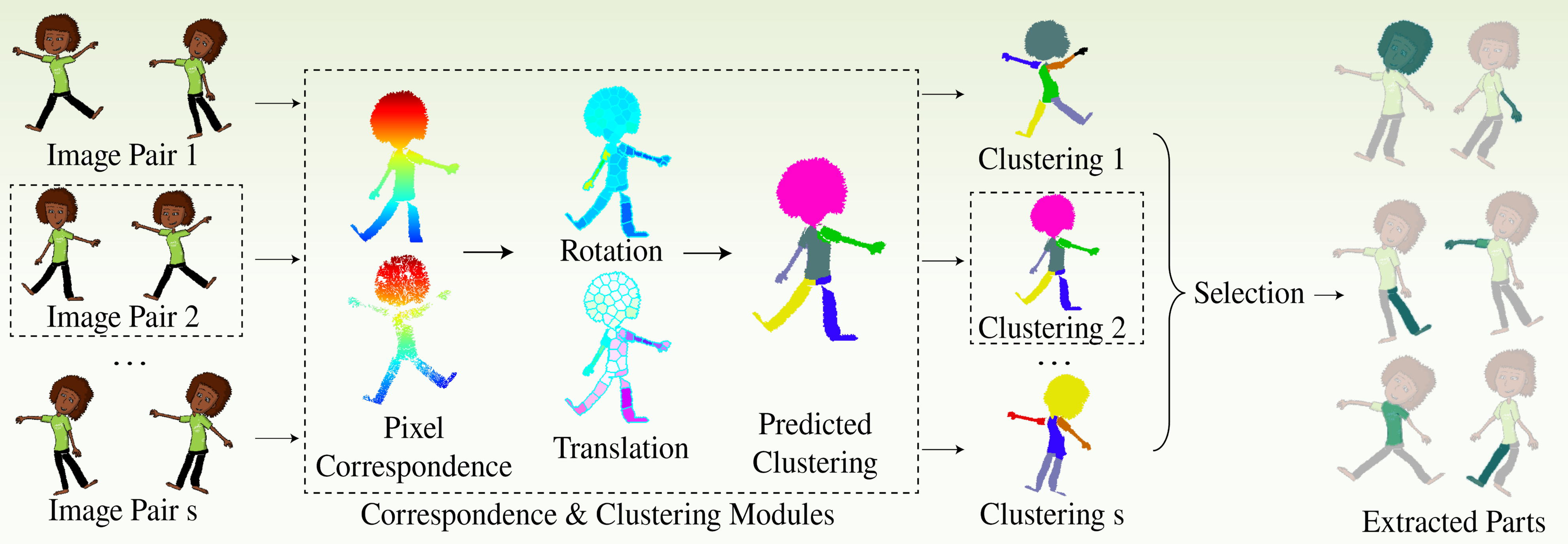
SCOPS [Hung et al. 2017] MoCoSeg [Siarohin et al. 2021]

Our approach: automatically extracts articulated parts from 2D characters by combining deep learning and linear programming optimization.

Key ideas of our method:

- **A correspondence module** predicts pixel-level correspondence (motion mapping) between a pair of different poses.
- **A clustering module** clusters pixels into articulated moving parts without relying on a known character template.
- **An optimization procedure** based on integer linear programming relaxation for finding parts that best reconstruct the given sprite poses.

Method



Architecture: an overview of our pipeline to extract parts from sprite sheet.

Correspondence module: predicts motion mapping of foreground pixels between a pair of images. Given each foreground pixel in the source image, its corresponding pixel in the target image is found as the pixel with the most similar feature vector.

$$x' = \underset{u \in \boldsymbol{I}_t, M_t(u)=1}{\operatorname{argmax}} (\boldsymbol{F}_s(x) \cdot \boldsymbol{F}_t(u))$$

I_s, I_t: source and target images
F_s, F_t: feature maps of I_s, I_t
M_t : mask indicating foreground
x, x': corr. pixels

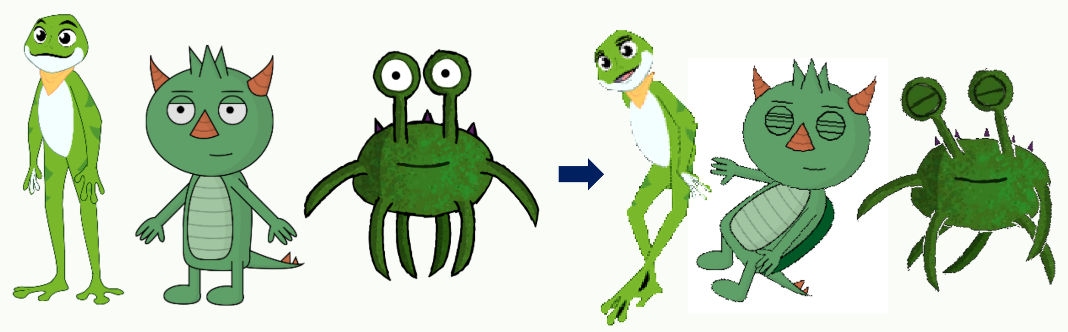
Clustering module: discovers articulated parts by grouping pixels with similar motion transformations. We predict rotation and translation for each super-pixel, and construct motion difference matrix and affinity matrix, followed by differentiable spectral clustering.

Part selection: selects a compact set of parts from a "soup" of candidate parts as a set cover problem. We measure the quality of possible solutions by deforming the selected parts to reconstruct all the poses via ARAP deformation and choose the one with the minimal reconstruction error.

$$min \sum_{q_c \in Q} z_c \quad s.t. \sum_{q_c : p_i \in q_c} z_c \geq 1 \ for\ all\ p_i \in \boldsymbol{P}$$

$P = \{p_i\}$: all the superpixels across all poses
$Q = \{q_i\}$: a "soup" of candidate parts
z_c: binary variable indicating whether part q_c is selected.

Training: Correspondence and clustering modules are trained in a supervised manner on our synthetic dataset. Correspondence module is pretrained on Creative Flow+ dataset. Part selection is done with a parameter-free optimization.

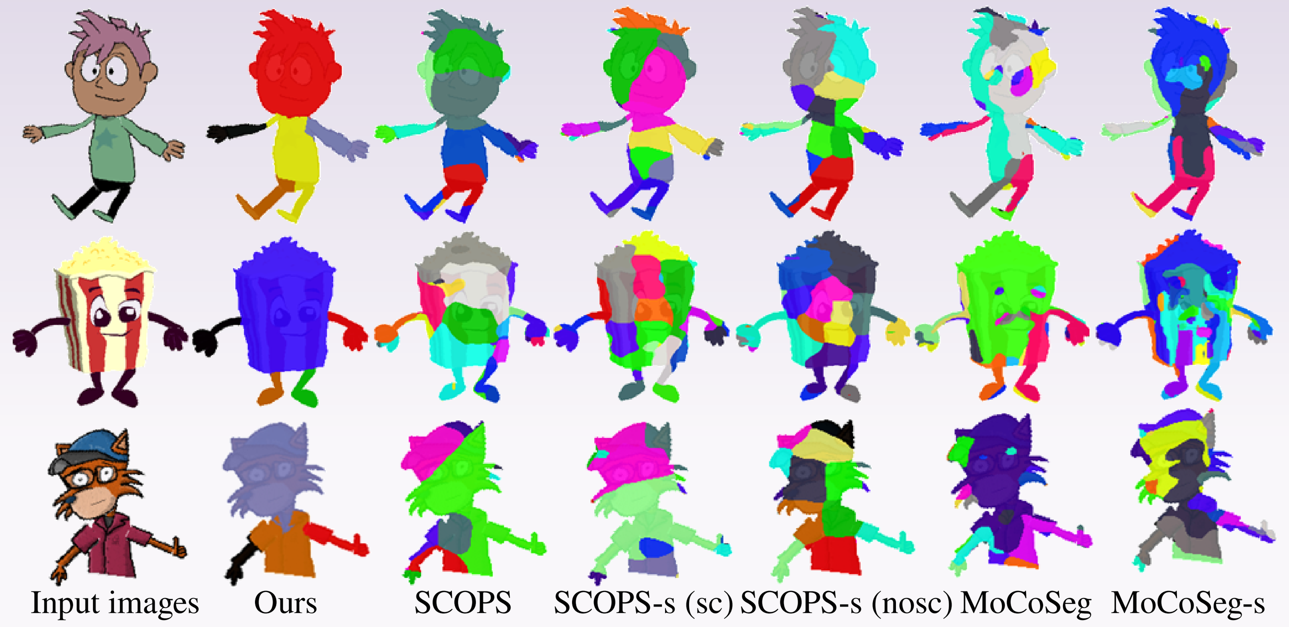


Synthetic poses of rigged puppets for training

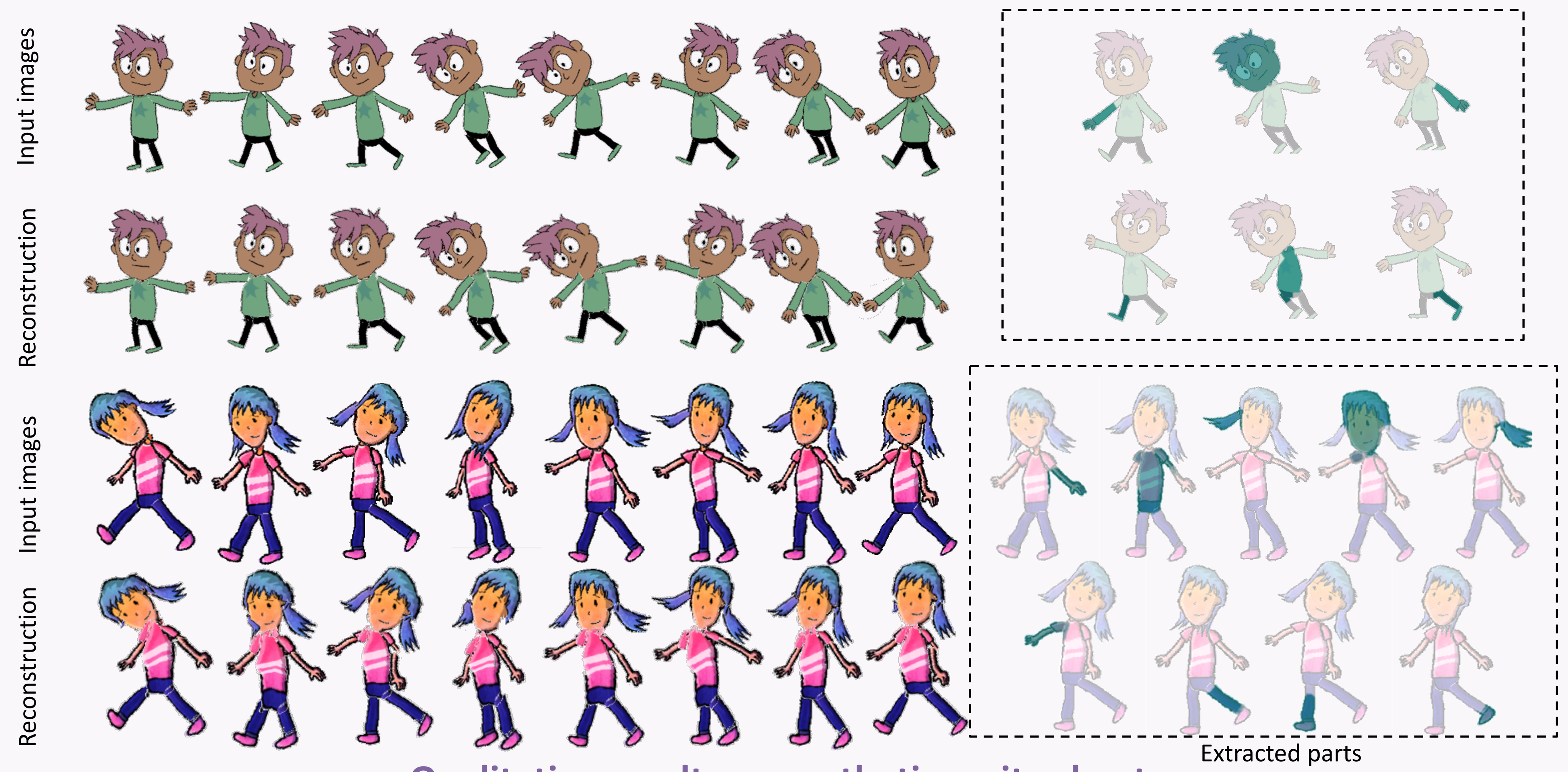
Results

Method	Part IoU
SCOPS	27.4%
SCOPS-s (sc)	33.1%
SCOPS-s (nosc)	35.8%
MoCoSeg	26.0%
MoCoSeg-s	32.3%
APES	**71.0%**

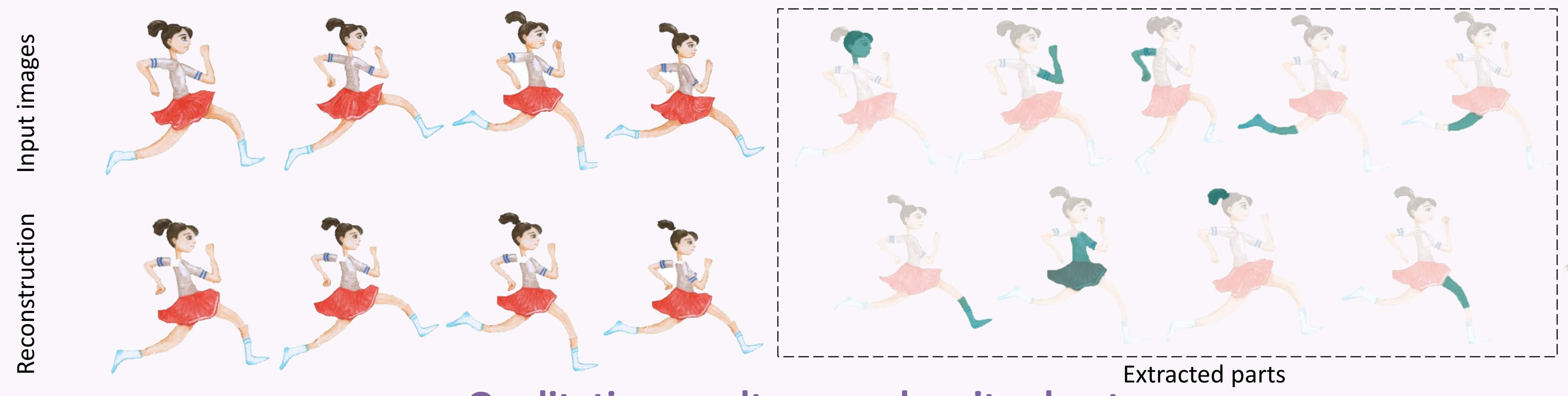
Quantitative comparison with alternatives



Qualitative comparison with alternatives



Qualitative results on synthetic sprite sheet



Qualitative results on real sprite sheet

For paper, code and dataset, please visit our project page:
https://zhan-xu.github.io/parts/