Weekly Report

Philip Woolley

2024-12-06

Time Log Reponse:

- Finished final project report and pushed code to repository
- N/A
-

1 Abstract

Abstract Point cloud registration from laser scanning data is a technique to establish the mapping relationship between source and target point clouds, which has been widely used in automatic 3D reconstruction, pose estimation, localization, and navigation. While algorithms like Super4PCS and MSSF-4PCS can achieve registration without initial poses, they are relatively slow, less accurate, and require iterations. To address these issues, we propose a 3D point cloud registration algorithm based on interval segmentation and multi-dimensional feature. Firstly, the source and target point clouds are segmented internally and the point cloud curvature is designed to narrow down the search range for the registration between the segmented point clouds. Secondly, the corresponding four-point sets in the segmented areas of the source and target point clouds are determined using affine invariance constraints. Finally, a multi-dimensional feature vector based on curvature features and fast point feature histogram is established to determine the unique corresponding four-point set pairs, and the rigid body transformation matrix is solved accordingly. Our algorithm is tested on publicly available 3D point cloud data models Bunny, Dino, Dragon, and Horse from Stanford University. Results showed that our algorithm improved registration accuracy by 24.39% and registration efficiency by 46.21% compared to the MSSF-4PCS point cloud registration algorithm. Multiple sets of experimental results confirmed the effectiveness of our algorithm. The proposed 3D point cloud registration is proved to be fast with high accuracy, which can be utilized for automatic segmentation, reconstruction, and modelling from Laser Scanning Data.

Summary This paper describes 3D point cloud registration for models created using laser scanning. I chose to review this paper to see the data presentations they use, as I am looking for clear, concise ways to display the results of the registration portion of my 3D pipeline. I think the primary step-by-step model visualization the authors provide is very clear, and is similar to how I intend to present my process. The authors methods are meant for full model-to-model matching, and so are not particularly useful for my project. The authors present full algorithm pseudocode, which can be useful but would bog down my report. They also present a graph of RMSE vs hyperparameter values, which is an interesting idea I had not thought of.

Citation A. Xu, L. Rao, G. Fan and N. Chen, "Fast and High Accuracy 3D Point Cloud Registration for Automatic Reconstruction From Laser Scanning Data," in IEEE Access, vol. 11, pp. 42497-42509, 2023, doi: 10.1109/ACCESS.2023.3270502. keywords: Point cloud compression;Three-dimensional displays;Feature extraction;Real-time systems;Approximation algorithms;Matrix converters;Histograms;Point cloud registration;point cloud curvature;affine invariance;multi-dimensional features,

2 Scripts and Code Blocks

This week I worked only on documentation, so there are not code blocks.

3 Documentation

4 Script Validation (Optional)

5 Results Visualization

6 Proof of Work

Please see repotitory

7 Next Week's Proposal

- N/A

Week16 report

Ruiqing Wang | CiChild CV team

- What progress did you make in the last week?
  1. Meet with Bree and Finish report writing for DLC on lizard report
  2. Commit to github repo and update my scripts
  3. Finish final presentation on DeepLabCut
  4. Attend Cichild group meeting with current work
  5. Help assembling paper report submissions and address submission situation.
- What are you planning on working on next?
  1. Finish up project deliverables
  2. Meet with Cichild CV team to discuss current progress
- Is anything blocking you from getting work done?

N/A

Paper abstract

Paper: JRDB-Pose: A Large-scale Dataset for Multi-Person Pose Estimation and Tracking
https://openaccess.thecvf.com/content/CVPR2023/html/Vendrow_JRDB-Pose_A_Large-Scale_Dataset_for_Multi-Person_Pose_Estimation_and_Tracking_CVPR_2023_paper.html

Abstract: Autonomous robotic systems operating in human environments necessitate a sophisticated understanding of their surroundings to ensure accurate and safe decision-making, particularly in scenarios involving close human-robot interactions and navigation through crowded spaces. The JRDB-Pose dataset addresses the limitations of existing datasets by providing comprehensive pose annotations and tracking IDs across diverse, challenging scenes, thereby facilitating the development and evaluation of multi-person pose estimation and tracking algorithms. This dataset is particularly valuable as it captures the complexities of human motion dynamics and occlusions, which are critical for advancing robotic perception in social navigation contexts.

Methodology:
The JRDB-pose dataset includes 600,000 pose annotations and head box annotations, each linked to unique tracking IDs and detailed occlusion labels, facilitating the study of human motion in crowded settings. The dataset's integration with JRDB and JRDB-Act enhances its utility for robotic navigation tasks, while the introduction of new evaluation metrics, OSPA-Pose and OSPA(2)-Pose, allows for a more nuanced assessment of state-of-the-art methods in pose estimation and tracking. The JRDB-Pose dataset is constructed from 64 minutes of video data sourced from the JRDB dataset, encompassing 54 sequences recorded in diverse indoor and outdoor university campus settings. It features 57,687 annotated panoramic frames, with a total of 636,000 pose instances and 11 million labeled keypoints, including 17 specific anatomical keypoints per pose, all of which are assigned consistent tracking IDs that maintain integrity even during occlusions. The annotation process involved manual labeling based on predefined bounding boxes, ensuring high-quality and temporally consistent data, which is crucial for advancing multi-person detection, pose estimation, and activity recognition research.

Scripts and Code Blocks

This week I mainly working on writing up current report and so some finish-up analysis.
Uploaded the running scripts for DLC:

DLC_model/configuration_set_up/Network_training_export.py

```
@@ -0,0 +1,19 @@
+ import deeplabcut
+ import os
+
+ path_config_file = "/home/hice1/rwang753/scratch/week8/20_videos_0603-RW-2024-10-10/config.yaml"
+
+ deeplabcut.create_training_dataset(path_config_file, net_type='resnet_50', augmenter_type='imgaug')
+
+ deeplabcut.train_network(path_config_file, shuffle=2, displayiters=100,saveiters=100)
+
+ deeplabcut.evaluate_network(path_config_file,plotting=True)
+
+ new_video_path = "/home/hice1/rwang753/scratch/week9/trained_20videos_0603_RW_2024-10-10/20_videos_0603-RW-2024-10-10/videos/5_27_2024"
+ deeplabcut.analyze_videos(path_config_file,new_video_path, videotype='.MP4',save_as_csv=True)
+ video_files = [f for f in os.listdir(new_video_path) if f.endswith(('.MP4'))]
+ for video_file in video_files:
+     # Construct the full path to the video file
+     video_path = os.path.join(new_video_path, video_file)
+     deeplabcut.create_labeled_video(path_config_file,video_path)
+     deeplabcut.plot_trajectories(path_config_file,video_path)
```

Documentation

Regarding the final presentation, please check the website link.
For final lizard report, here is the link:
https://gtvault-my.sharepoint.com/:w:/g/personal/bshi42_gatech_edu/EYzZJ44yBrdOg1_6B6p4c-oBMG_zc3qVlx-UIuxUH3TWfw?e=5knKdp

Results Visualization and Code Validation

Assembled all current work regarding deeplabcut and SLEAP, discuss the potential work on next spring. For Github link, here is: https://github.com/Human-Augment-Analytics/Lizard-Pose-Estimation-and-Evaluation

Proof of Work and code validation
N/A

Next Week's Proposal

1. Finish up writing
2. Meet with Cichild CV team to discuss current progress

Week 16 Document Submission

Lizard X-RAY Landmark Group

Mercedes Quintana

What progress did you make in the last week?

- Finished website for semester
- Prepared app for demo with Ayush
- Annotated new GitHub
- Prepared slides for showcase

What are you planning on working on next?

- Create new toe dataset to better classify toes
- Continue to update the lizard cv website
- Add more functionality for app

Is anything blocking you from getting work done?

- Last week, between being sick and family obligations, I couldn't get any work done so I didn't submit a weekly report

Abstracts:

URL: https://www.mdpi.com/2076-3417/11/10/4416

Measurement of Fish Morphological Features through Image Processing and Deep Learning Techniques

Abstract: Noninvasive morphological feature monitoring is essential in fish culture, since these features are currently measured manually with a high cost. These morphological parameters can concern the size or mass of the fish, or its health as indicated, for example, by the color of the eyes or the gills. Several approaches have been proposed, based either on image processing or machine learning techniques. In this paper, both of these approaches have been combined in a unified environment with novel techniques (e.g., edge or corner detection and pattern stretching) to estimate the fish's relative length, height and the area it occupies in the image. The method can be extended to estimate the absolute dimensions if a pair of cameras is used for obscured or slanted fish. Moreover, important fish parts such as the caudal, spiny and soft dorsal, pelvic and anal fins are located. Four species popular in fish cultures have been studied: Dicentrarchus labrax (sea bass), Diplodus puntazzo, Merluccius merluccius (cod fish) and Sparus aurata (sea bream). Taking into consideration that there are no large public datasets for the specific species, the training and testing of the developed methods has been performed using 25 photographs per species. The fish length estimation error ranges between 1.9% and 13.2%, which is comparable to the referenced approaches that are trained with much larger datasets and do not offer the full functionality of the proposed method.

Summary: This paper describes creating a dataset and a method to compute noninvasive morphological features to check the health of farm fish using deep learning and image processing. Fish part locatlization had a success rate of 91% with an average length error estimation of 4.93%.

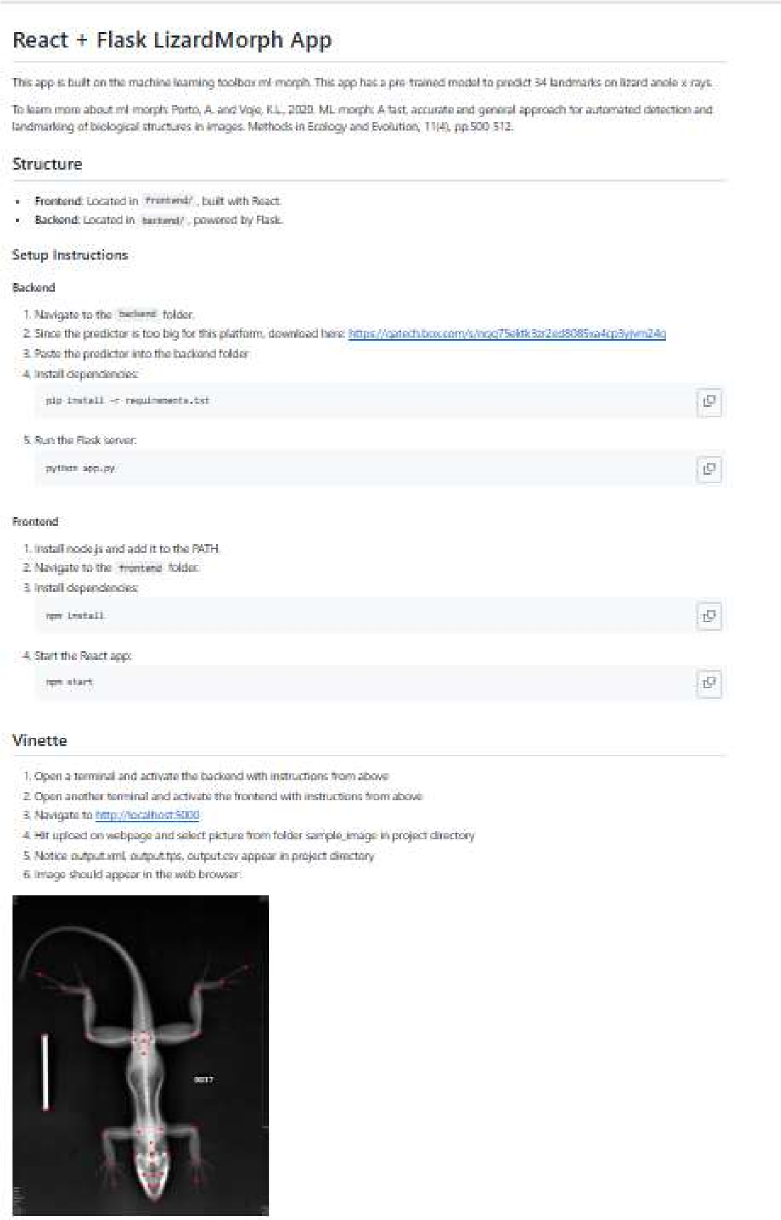
Scripts and Code Blocks:

Ayush and I put together the visuals from the base app that he created. We put everything in a new Github repository: https://github.com/Human-Augment-Analytics/Lizard-CV-Web-App/tree/main

This is the repository we are going to use from now on.

Documentation:

This is the tutorial I wrote for the new Github:



React + Flask LizardMorph App

This app is built on the machine learning toolbox ml-morph. This app has a pre-trained model to predict 34 landmarks on lizard anole x rays.

To learn more about ml-morph: Porto, A. and Voje, K.L., 2020. ML-morph: A fast, accurate and general approach for automated detection and landmarking of biological structures in images. Methods in Ecology and Evolution, 11(4), pp.500-512.

Structure

- **Frontend**: Located in `frontend/`, built with React.
- **Backend**: Located in `backend/`, powered by Flask.

Setup Instructions

Backend

1. Navigate to the `backend` folder.
2. Since the predictor is too big for this platform, download here: https://gatech.box.com/s/[illegible]
3. Paste the predictor into the backend folder
4. Install dependencies:

```
pip install -r requirements.txt
```

5. Run the Flask server:

```
python app.py
```

Frontend

1. Install node.js and add it to the PATH.
2. Navigate to the `frontend` folder.
3. Install dependencies:

```
npm install
```

4. Start the React app:

```
npm start
```

Vinette

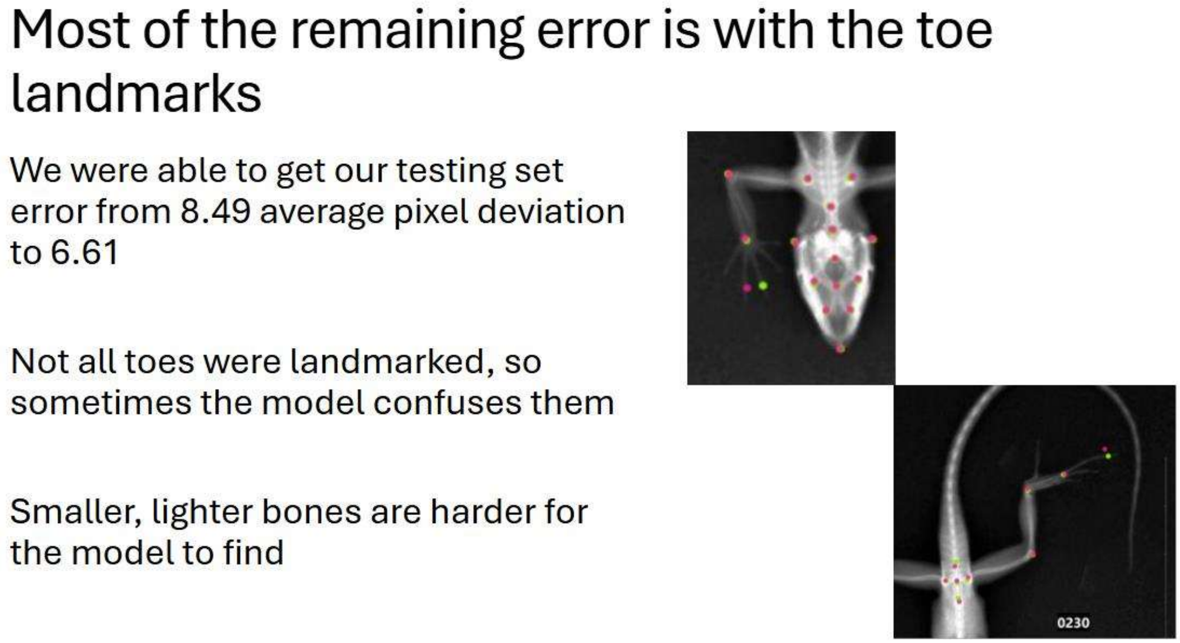
1. Open a terminal and activate the backend with instructions from above
2. Open another terminal and activate the frontend with instructions from above
3. Navigate to http://localhost:5000
4. Hit upload on webpage and select picture from folder sample_image in project directory
5. Notice output.xml, output.tps, output.csv appear in project directory
6. Image should appear in the web browser:

Script Validation:

I have no validation steps now.

Results Visualization / Proof of Work:

Here is a slide from the showcase, which describes our remaining error issues:



Most of the remaining error is with the toe landmarks

We were able to get our testing set error from 8.49 average pixel deviation to 6.61

Not all toes were landmarked, so sometimes the model confuses them

Smaller, lighter bones are harder for the model to find

Next Semester Proposal:

Next semester, I will create a pipeline to improve classification accuracy on the toes. I will also continue to update the website if I am able to continue as the webdev. I will continue to improve the app usability by making it possible to upload multiple images.