3D Animal Reconstruction with Deformable Template Models

Benjamin Biggs

Supervisor: Dr. Andrew Fitzgibbon
Prof. Roberto Cipolla

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Jesus College | January 2021

"Our perfect companions never have fewer than four feet."
Colette (1873 – 1954)

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Benjamin Biggs
January 2021

Acknowledgements

This work would not have been possible without the dedicated support from my PhD supervisors, Andrew Fitzgibbon and Roberto Cipolla. I would also like to thank the dedicated staff at GlaxoSmithKline, in particular the Pharma Supply Chain Tech Digital Innovation group, and particularly my line manager Patrick Hyett and project sponsor Julie Huxley-Jones.

Abstract

TODO. We present a system to recover the 3D shape and motion of a wide variety of quadrupeds from video. The system comprises a machine learning front-end which predicts candidate 2D joint positions, a discrete optimization which finds kinematically plausible joint correspondences, and an energy minimization stage which fits a detailed 3D model to the image. In order to overcome the limited availability of motion capture training data from animals, and the difficulty of generating realistic synthetic training images, the system is designed to work on silhouette data. The joint candidate predictor is trained on synthetically generated silhouette images, and at test time, deep learning methods or standard video segmentation tools are used to extract silhouettes from real data. The system is tested on animal videos from several species, and shows accurate reconstructions of 3D shape and pose.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

Animal welfare is an important concern for business and society, with an estimated 70 billion animals currently living under human care. Monitoring and assessment of animal health can be assisted by obtaining accurate measurements of an individual's shape, volume and movement. These measurements should be taken without interfering with the animal's normal activity, and are needed around the clock, under a variety of lighting and weather conditions, perhaps at long range (e.g. in farm fields or wildlife parks). Therefore a very wide range of cameras and imaging modalities must be handled. For small animals in captivity, a depth camera might be possible, but techniques which can operate solely from intensity data will have a much wider range of applicability

1.2 Approach

We address this problem using techniques from the recent human body and hand tracking literature, combining machine learning and 3D model fitting. A discriminative front-end uses a deep hourglass network to identify candidate 2D joint positions. These joint positions are then linked into coherent skeletons by solving an optimal joint assignment problem, and the resulting skeletons create an initial estimate for a generative model-fitting back-end to yield detailed shape and pose for each frame of the video. Although superficially similar to human tracking, animal tracking (AT) has some interesting differences that make it worthy of study:

Variability. In one sense, AT is simpler than human tracking as animals generally do not wear clothing. However, variations in surface texture are still considerable between individuals, and the variety of shape across and within species is considerably greater. If tracking is specialized to a particular species, then shape variation is smaller, but training data is even harder to obtain. Training data. For human tracking, hand labelled sequences of 2D segmentations and joint positions have been collected from a wide variety of sources [3–5]. Of these two classes of labelling, animal segmentation data is available in datasets such as MSCOCO [4], PASCAL VOC [6] and DAVIS [7]. However this data is considerably sparser than human data, and must be "shared" across species, meaning the number of examples for a given animal shape class is considerably fewer than is available for an equivalent variation in human shape. While segmentation data can be supplied by non-specialist human labellers, it is more difficult to obtain joint position data. Some joints are easy to label, such as "tip of snout", but others such as the analogue of "right elbow" require training of the operator to correctly identify across species. Of more concern however, is 3D skeleton data. For humans, motion capture (mocap) can be used to obtain long sequences of skeleton parameters (joint positions and angles) from a wide variety of motions and activities. For animal tracking, this is considerably harder: animals behave differently on treadmills than in their quotidian environments, and although some animals such as horses Creatures great and SMAL 3 and dogs have been coaxed into motion capture studios [8], it remains impractical to consider mocap for a family of tigers at play. These concerns are of course alleviated if we have access to synthetic training data. Here, humans and animals share an advantage in the availability of parameterized 3D models of shape and pose. The recent publication of the Skinned Multi-Animal Linear (SMAL) model [9] can generate a wide range of quadruped species, although without surface texture maps. However, as with humans, it remains difficult to generate RGB images which are sufficiently realistic to train modern machine learning models. In the case of humans, this has been overcome by generating depth maps, but this then requires a depth camera at test time [10]. The alternative, used in this work, is to generate 2D silhouette images so that machine learning will predict joint heatmaps from silhouettes only

1.3 Contributions

In summary, the contributions of this thesis are as follows:

1. We demonstrate a robust framework for 3D animal reconstruction using deformable template models

1.4 Co-Authored Papers

Extracts from this thesis appear in the following co-authored publications and preprints. Chapter 2 contains work from:

1. Biggs Benjamin, Roddick Thomas

1.5 Thesis Structure

The following four thesis chapters discuss important

Chapter 2

Related Work

2.1 Introduction

In this section, I will introduce the components required for the rest of this thesis.

2.2 Representing 3D Objects

Talk about meshes, radiance fields etc.

2.3 Reconstructing 3D Objects from Images

2.3.1 Classical theory

Give an overview towards classical NRSFM methods. Introduce camera geometry, rendering, etc. include lots of maths.

2.3.2 Articulated Subjects

Articulated subjects pose more challenges etc. Some methods maintain a model-free approach as before, talk about various works for humans (e.g. PiFu).

However, model-based approaches offer certain advantages (semantically meaningful parts, etc.). Explain SMPL and SMAL models.

2.4 3D Reconstruction with Deformable Template Meshes

Introduce energy based methods and include loss functions etc., incl. SMAL optimizer, SMALR and finally SMAL-ST (deep learning techniques). All the SMPL approaches: HMR, GraphCMR, SPIN etc.

Explain the various types of loss that are used. Good place for a Andrew's table.

2.5 Data-Limited Training

What do people do when there is no data available?

2.6 Dealing with Ambiguous Input

How to people cope with ambiguities?

Chapter 3

Learning from Synthetic Data

3.1 Introduction

And now I begin my third chapter here . . .

1. We want precise shape reconstructions in real-time
2. Refer back to SMAL-ST, requires 3D synthetic training data from video
3. A whole chapter on ’training in-the-loop’ methods
4. Talk about expectation maximization, SMBLD, EM-in-the-loop
5. Optional: Talk about how this can be extended to a multi-view GSK setup, similar to MeshRCNN

Chapter 4

Precise Shape Reconstructions

4.1 First section of the third chapter

And now I begin my third chapter here . . .

And now to cite some more people Read [2], Ancey et al. [1]

Chapter 5

Handling Ambiguities

5.1 First section of the third chapter

In this section, blah blah.

References

[1] Ancey, C., Coussot, P., and Evesque, P. (1996). Examination of the possibility of a fluid-mechanics treatment of dense granular flows. *Mechanics of Cohesive-frictional Materials*, 1(4):385–403.

[2] Read, C. J. (1985). A solution to the invariant subspace problem on the space l_1. *Bull. London Math. Soc.*, 17:305–317.

Appendix A

How to install $\LaTeX$

Windows OS

TeXLive package - full version

1. Download the TeXLive ISO (2.2GB) from https://www.tug.org/texlive/
2. Download WinCDEmu (if you don't have a virtual drive) from http://wincdemu.sysprogs.org/download/
3. To install Windows CD Emulator follow the instructions at http://wincdemu.sysprogs.org/tutorials/install/
4. Right click the iso and mount it using the WinCDEmu as shown in http://wincdemu.sysprogs.org/tutorials/mount/
5. Open your virtual drive and run setup.pl

or

Basic MikTeX - $\TeX$ distribution

1. Download Basic-MiK$\TeX$(32bit or 64bit) from http://miktex.org/download
2. Run the installer
3. To add a new package go to Start » All Programs » MikTex » Maintenance (Admin) and choose Package Manager

4. Select or search for packages to install

TexStudio - TeX editor

1. Download TexStudio from http://texstudio.sourceforge.net/#downloads

2. Run the installer

Mac OS X

MacTeX - TeX distribution

1. Download the file from https://www.tug.org/mactex/

2. Extract and double click to run the installer. It does the entire configuration, sit back and relax.

TexStudio - TeX editor

1. Download TexStudio from http://texstudio.sourceforge.net/#downloads

2. Extract and Start

Unix/Linux

TeXLive - TeX distribution

Getting the distribution:

1. TexLive can be downloaded from http://www.tug.org/texlive/acquire-netinstall.html.

2. TexLive is provided by most operating system you can use (rpm,apt-get or yum) to get TexLive distributions

Installation

1. Mount the ISO file in the mnt directory

   ```
   mount -t iso9660 -o ro,loop,noauto /your/texlive####.iso /mnt
   ```

2. Install wget on your OS (use rpm, apt-get or yum install)

3. Run the installer script install-tl.

   ```
   cd /your/download/directory
   ./install-tl
   ```

4. Enter command ‘i’ for installation

5. Post-Installation configuration:
   http://www.tug.org/texlive/doc/texlive-en/texlive-en.html#x1-320003.4.1

6. Set the path for the directory of TexLive binaries in your .bashrc file

For 32bit OS

For Bourne-compatible shells such as bash, and using Intel x86 GNU/Linux and a default directory setup as an example, the file to edit might be

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/i386-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

For 64bit OS

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/x86_64-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
```

```
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

Fedora/RedHat/CentOS:

```
sudo yum install texlive
sudo yum install psutils
```

SUSE:

```
sudo zypper install texlive
```

Debian/Ubuntu:

```
sudo apt-get install texlive texlive-latex-extra
sudo apt-get install psutils
```

Appendix B

Installing the CUED class file

LaTeX.cls files can be accessed system-wide when they are placed in the <texmf>/tex/latex directory, where <texmf> is the root directory of the user's TeXinstallation. On systems that have a local texmf tree (<texmflocal>), which may be named "texmf-local" or "localtexmf", it may be advisable to install packages in <texmflocal>, rather than <texmf> as the contents of the former, unlike that of the latter, are preserved after the LaTeXsystem is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory <texmf>/tex/latex/CUED for all CUED related LaTeXclass and package files. On some LaTeXsystems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For TeXLive systems this is accomplished via executing "texhash" as root. MIKTeXusers can run "initexmf -u" to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in LaTeX.