

Menpo: A Comprehensive Platform for Parametric Image Alignment and Visual Deformable Models

Joan Alabert-i-Medina^{*}, Epameinondas Antonakos^{*}, James Booth^{*}, Patrick Snape^{*}
and Stefanos Zafeiriou
Department of Computing, Imperial College London
180 Queens Gate, London SW7 2AZ, UK
{ja310, e.antonakos, jab08, p.snape, szafeiri}@imperial.ac.uk

ABSTRACT

The Menpo Project, hosted at <http://www.menpo.io>, is a BSD-licensed software platform providing a complete and comprehensive solution for annotating, building, fitting and evaluating deformable visual models from image data. Menpo is a powerful and flexible cross-platform framework written in Python that works on Linux, OS X and Windows. Menpo has been designed to allow for easy adaptation of Lucas-Kanade (LK) parametric image alignment techniques, and goes a step further in providing all the necessary tools for building and fitting state-of-the-art deformable models such as Active Appearance Models (AAMs), Constrained Local Models (CLMs) and regression-based methods (such as the Supervised Descent Method (SDM)). These methods are extensively used for facial point localisation although they can be applied to many other deformable objects. Menpo makes it easy to understand and evaluate these complex algorithms, providing tools for visualisation, analysis, and performance assessment. A key challenge in building deformable models is data annotation; Menpo expedites this process by providing a simple web-based annotation tool hosted at <http://www.landmarker.io>. The Menpo Project is thoroughly documented and provides extensive examples for all of its features. We believe the project is ideal for researchers, practitioners and students alike.

Categories and Subject Descriptors

I.5.4 [Pattern Recognition]: Applications—*computer vision*; D.0 [Software]: General

General Terms

Algorithms; Design; Experimentation

*Alphabetical order due to equal contribution

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MM '14 November 03 - 07 2014, Orlando, FL, USA
Copyright 2014 ACM 978-1-4503-3063-3/14/11 ...\$15.00
<http://dx.doi.org/10.1145/2647868.2654890>



Figure 1: The Menpo Project provides all the tools necessary to perform facial point localisation.

Keywords

Active Appearance Models; Constrained Local Models; Supervised Descent Method; Regression; Python

1. INTRODUCTION

Rigid and non-rigid image alignment and the closely related field of visual deformable models (i.e., models of non-rigid visual objects, such as human faces) are all well-studied areas of computer vision, which have created a wealth of scientific research and countless applications. Despite this, there is still a noteworthy lack of high quality open source software in this area. Most existing packages are encrypted, compiled, non-maintained, badly structured or difficult to modify, all characteristics which make them unsuitable for adoption in cutting edge scientific research. Furthermore, the lack of an open source package that implements the most important techniques of the field in a unified manner obstructs the possibility to fairly assess the various methods against one another. We believe that the qualities and characteristics of the Menpo Project make it a suitable candidate to fill this existing gap.

Among the previously mentioned parametric image alignment algorithms, Lucas-Kanade (LK) [1] is arguably the most popular. Active Appearance Models (AAMs) [2] are an extension of LK to the problem of generative deformable model fitting that have recently been shown to achieve state-of-the-art results [3]. Similar performance has been achieved

with discriminative part-based model techniques such as Constrained Local Models (CLMs) [4] and the recently proposed Supervised Descent Method (SDM) [5].

Some implementations of the above methods are publicly available. Various implementations of LK are provided in [1] and an implementation of AAM building and fitting is given in [6]. To the best of our knowledge, for the recent state-of-the-art LK [7], AAMs [8,9], CLMs [4] and SDMs [5], only compiled, encrypted, or pre-trained implementations exist (some are provided by our group¹). In particular, it should be noted that whilst many packages provide fitting algorithms, few provide training code.

Menpo provides robust open source implementations of all the above algorithms and is designed and developed to be flexible and simple to use. One of Menpo's most powerful features is the unification of the algorithms within a single framework, which enables easier investigation of new research areas. Furthermore, it provides implementations of many powerful features and descriptors used in computer vision, such as Local Binary Patterns [10], Histogram of Oriented Gradients (HOGs) [11] and Image Gradient Orientations (IGOs) [7], all of which can be employed in the deformable model algorithms.

2. ARCHITECTURE

Choice of programming language. Menpo is primarily written in Python. The use of Python was motivated by its free availability on all platforms, unlike its major competitor in computer vision, Matlab. We believe this is hugely important for reproducible open science. Python provides a very flexible environment for performing research, and recent innovations such as the IPython notebook [13] have made it incredibly simple to provide documentation via examples. The vast majority of the execution time in Menpo is actually spent in highly efficient numerical libraries and bespoke C++ code, allowing us to achieve sufficient performance for real time facial point tracking whilst not compromising on the huge flexibility that Menpo offers.

Importing assets. Menpo can ingest a large range of 2D and 3D asset and annotation formats, and normalises differences between formats to provide consistent data to users. All major image types are supported, along with a number of common annotation formats used within the computer vision community. Because Menpo has been built with extensibility in mind, adding support for new asset formats is very simple. Since deformable modelling commonly requires pre-annotated data, Menpo will automatically import all annotation formats that have the same name as an importable file. For example, importing an entire facial database of images along with their annotations is as simple as:

```
import menpo.io as mio
images = list(mio.import_images("./trainset/*"))
```

High-level types. At Menpo's core are powerful `Image` and `PointCloud` classes. All types support rich introspection and visualisation, which is of huge benefit when working interactively in an IPython Notebook. The `Transform`

¹We have recorded more than 10K downloads for our implementations of some of the above techniques in the past couple of years.

²<http://www.visionopen.com/downloads/open-source-software/vosm/>

subclasses in Menpo provide a simple interface for manipulating data inside container types. All Menpo container types are transformable, and when a transform is applied on a container, all attached data is transformed too. For example, when rotating an image any attached landmarks are also rotated.

Vector representation. Computer vision algorithms are frequently formulated as linear algebra problems in high dimensional spaces, where each image is stripped into a vector form. Unfortunately, it is rare for the complete problem to be formulated in such a manner, as some operations still need to be applied in the image domain (e.g. calculating the gradient). Menpo bridges this gap by supporting bi-directional vectorisation of data through its `Vectorizable` interface. Through this, any type can be safely and efficiently converted to a vector form and back again, allowing for algorithms to be efficiently marshaled into vector and matrix forms.

3. ALGORITHMS

Menpo contains at its core two major families of algorithms: generative LK techniques, including AAMs, and discriminative part-based models, which consist of CLMs and SDMs. For all algorithms training and fitting implementations are provided, something which uniquely differentiates Menpo from other packages.

Lucas-Kanade. The LK family of algorithms seek to minimise the discrepancies between a given input image and either a template image (LK) or an appearance model (AAM). This is achieved in a generative manner, whereby parameters are computed in order to correctly warp one of the images into the reference frame of the other. Optimisation of these parameters is usually achieved via a Gauss-Newton approximation and a number of formulations of the cost function have been proposed. Within Menpo, we have implementations of all major LK algorithms [1] (forward additive/compositional, inverse additive/compositional) and all major AAM algorithms (project-out [1] and alternating [9]). Building and fitting an AAM is extremely simple, requiring only 4 lines of code:

```
from menpo.fitmultilevel.aam import *
aam = AAMBuilder().build(images) # Build
aam_fit = LucasKanadeAAMFitter(aam) # Fit
result = aam_fit.fit(test_image, initial_shape)
```

CLMs are part-based deformable models that fit global parametric models of shape to images by maximising the joint probability of their landmarks being correctly aligned. To this end, a set of individual discriminative classifiers that model the probability of each landmark being correctly aligned are learnt from the local appearance region around each landmark. Fitting consists of maximising the joint probability of each classifier whilst ensuring that the final shape is highly plausible under the global parametric shape model. Menpo has an implementation of the well-known Regularized-Landmark Mean-Shift algorithm proposed by Saragih et. al in [4]. The obvious advantage of Menpo's implementation with respect to other existing implementations is the ease with which new appearance representations and different discriminative classifiers can be incorporated in to the existing framework.

Regression-based techniques aim at solving the same type of problem solved by deformable models such as AAMs

Table 1: Comparison of existing implementations. No other package provides implementations of training and fitting for the AAM, CLM and regression-based methods.

Tool	Environment	License	LK Image Alignment	AAM		CLM		Regression	
				Train	Fit	Train	Fit	Train	Fit
Menpo	Python	BSD	+	+	+	+	+	+	+
DeMoLib [6]	C++	Academic	-	+	+	-	-	-	-
Intraface [5]	C++/Matlab	Academic	-	-	-	-	-	-	+
FaceTracker [4]	C++	Academic	-	-	-	-	+	-	-
DRMF [12]	Matlab	Academic	-	-	-	-	+	-	-
VOSM ² [1]	C++	LGPL	-	+	+	-	-	-	-
	Matlab	-	+	-	-	-	-	-	-

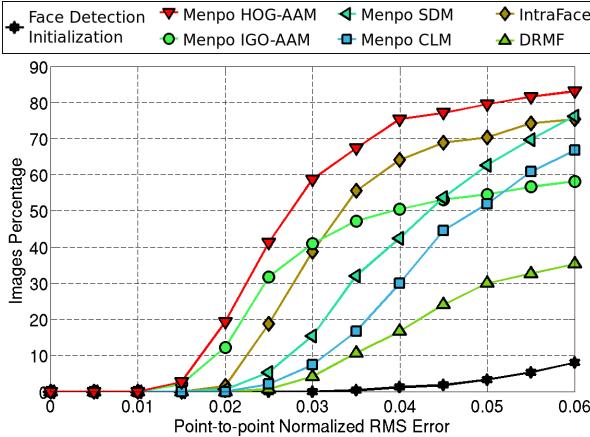


Figure 2: Performance comparison between some of Menpo’s basic methods (SDM, CLMs, HOG and IGO AAMs) and two other available packages (Intraface [5] and DRMF with CLMs [12]) on AFW database.

or CLMs; i.e. finding the optimal position of the set of landmarks describing the shape of an object of interest in an image. These techniques attempt to solve the problem by generally learning a cascade of functional mappings (commonly referred to as a cascade of regressors) between the object appearance and the optimal landmark configuration. Due to the availability of high quality annotated data in some areas of computer vision (such as non-rigid face alignment), these techniques have become effective and extremely attractive due to their extremely low computational fitting complexity. Menpo provides a complete framework for developing cascade-regression methods for object landmark localisation. More specifically, Menpo provides a flexible implementation of the popular Supervised Descent Method (SDM) [5], and cascade-regression implementations for fitting Deformable Models that are the direct application of the Supervised Descent approach to CLMs (similar to [12]) and AAMs.

4. FITTING PERFORMANCE

Due to the rich functionality of Menpo, several experiments could be run to assess performance. However, due to lack of space, we restrict the performance evaluation to the problem of non-rigid face alignment “in-the-wild”.

The experiment of Figure 2 compares the fitting accuracy of four representative algorithms implemented in Menpo against two publicly available but closed-source implementations of state-of-the-art methods using two popular “in-the-wild” face

databases. The four Menpo algorithms under test are as follows: (1) the Alternating Inverse Compositional algorithm for fitting AAMs using IGO features (Menpo IGO-AAMs) [8] and (2) HOG features (Menpo HOG-AAMs) [3], (3) the Regularised Landmark Mean-Shift [4] algorithm for CLMs fitting using HOG features (Menpo CLMs) and (4) the Supervised Descent Method [5] using HOG features (Menpo SDMs). All algorithms were trained using the same 811 annotated training images of the publicly available LFPW database [14] provided by the 300W Challenge³ [15].

We test against: (1) the DRMF Matlab implementation provided by [12]⁴ and (2) the SIFT-SDMs Matlab implementation of [5]⁵. Due to the closed source nature of these implementations it is impossible to know what data was used to train the models, however through personal communication with the authors we know several thousand images were employed.

Results are reported for the 337 annotated images from the Annotated Faces in the Wild (AFW) database³ [16]. All methods were initialised using the publicly available bounding box face detections³. Note that these initialisations are particularly challenging, and so the experiment emphasises the robustness of the techniques. For each method, the Cumulative Error Distribution (CED) curves over 49 landmark points is reported in Fig. 2. The results indicate that on the whole Menpo is comparable to state-of-the-art, even given the deficiencies in training data used. Menpo HOG-AAM is by far the most accurate technique, although it is admittedly much slower than the SDM family of methods.

It should be noted that due to the fact that the third-party techniques tested do not provide training implementations it is impossible to guarantee a fair test. The aim of the experiment is rather to demonstrate that Menpo’s open-source implementations can compete against the state-of-the-art. We choose to train our techniques with a fairly small cohort of publicly available images due to the considerable computational requirements of the training process. It is hoped that this decision ensures that the results reported here are readily reproducible by the interested reader.

One of the aims of Menpo is to promote open, reproducible science. To that end, the full code for reproducing the results reported here will be made available before publication on the Menpo website. Going forwards we intend to maintain a standardised suite of tests based around the experiments performed in this paper for the community to have a single

³<http://ibug.doc.ic.ac.uk/resources/300-W/>

⁴<https://sites.google.com/site/akshayasthana/clm-wild-code>

⁵<http://www.humansensing.cs.cmu.edu/intraface/>

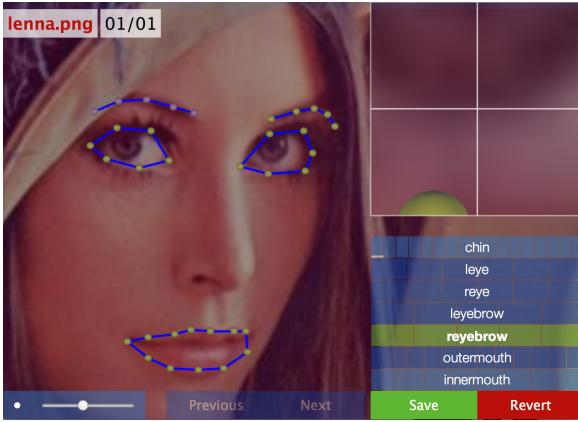


Figure 3: An example screenshot of the web based landmarking software.

point of reference for comparing cutting edge techniques.

5. APPLICATIONS

Building Menpo on top of Python makes it extremely easy to integrate into different applications. Currently the Menpo Project hosts two applications built on top of Menpo.

Web based landmarker <http://www.landmarker.io> is a web application for annotating 2D and 3D data. It has no dependencies beyond a modern web browser such as Google Chrome and is designed to be trivial to use. Most importantly, the decentralisation of the landmarking software means researchers can recruit annotators by simply directing them to the website. The annotations and assets themselves are served to the client from a separate server component which is run by the user. This allows researches to benefit from the web-based nature of the tool without having to compromise privacy or security. The server utilises Menpo to import assets and save out annotations. An example screenshot is given in Figure 3. In the future, the tool will use Menpo to provide initial estimations of the correct points for the images and meshes of interest.

Real-time tracker Menpo-Tracker is an experimental application for detecting and tracking object landmarks in real-time on a standard laptop. It defines a general framework for tracking deformable objects (it is not specifically tailored to faces) using SDM objects built using Menpo. This tracking application is implemented in Python and utilises OpenCV for video acquisition. We have high hopes that due to the accessibility of Python and its widespread user base, Menpo-Tracker will attract users from communities outside of computer vision. We have plans for Menpo-Tracker to be able to track object landmarks in real-time using GPU-based implementations of AAMs and CLMs built using future iterations of Menpo.

6. CONCLUSION

We present Menpo, the first, to the best of our knowledge, complete platform for parametric image alignment and deformable model construction, fitting and performance evaluation. Menpo is flexible and there are many interesting areas where it can be extended to provide further state-of-the-art implementations. Currently, we are working on implementing construction and fitting of 3D Morphable Models

(a lot of work has already been done towards this direction - Menpo includes an off-screen OpenGL rasterizer for the purpose, and all the tools necessary to visualise and work with 3D triangular meshes). Most importantly, we wish for Menpo to become an important set of reference implementations for researchers interested in the parametric image alignment and deformable modelling fields.

7. ACKNOWLEDGMENT

The work of Joan Alabert-i-Medina and Patrick Snape was funded by the Qualcomm Innovation Fellowship and by an EPSRC DTA from Imperial College London. The work of James Booth was funded by an EPSRC DTA from Imperial College London. The work of Epameinondas Antonakos and Stefanos Zafeiriou was partially supported by the EPSRC project EP/J017787/1 (4DFAB).

8. REFERENCES

- [1] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *IJCV*, vol. 56, no. 3, pp. 221–255, 2004.
- [2] T. Cootes *et al.*, “Active appearance models,” *IEEE T-PAMI*, vol. 23, no. 6, pp. 681–685, 2001.
- [3] E. Antonakos *et al.*, “Hog active appearance models,” in *ICIP*, 2014.
- [4] J. Saragih *et al.*, “Deformable model fitting by regularized landmark mean-shift,” *IJCV*, vol. 91, no. 2, pp. 200–215, 2011.
- [5] X. Xiong and F. De la Torre, “Supervised descent method and its applications to face alignment,” in *CVPR*, pp. 532–539, 2013.
- [6] J. Saragih and R. Goecke, “Learning aam fitting through simulation,” *Pat. Rec.*, vol. 42, no. 11, pp. 2628–2636, 2009.
- [7] G. Tzimiropoulos *et al.*, “Robust and efficient parametric face alignment,” in *ICCV*, 2011.
- [8] G. Tzimiropoulos *et al.*, “Generic active appearance models revisited,” in *ACCV*, pp. 650–663, 2012.
- [9] G. Papandreou and P. Maragos, “Adaptive and constrained algorithms for inverse compositional active appearance model fitting,” in *CVPR*, pp. 1–8, 2008.
- [10] T. Ojala *et al.*, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE T-PAMI*, vol. 24, no. 7, pp. 971–987, 2002.
- [11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, pp. 886–893, 2005.
- [12] A. Asthana *et al.*, “Robust discriminative response map fitting with constrained local models,” in *CVPR*, pp. 3444–3451, 2013.
- [13] F. Perez and B. Granger, “IPython: a system for interactive scientific computing,” *CiSE*, vol. 9, no. 3, pp. 21–29, 2007.
- [14] P. N. Belhumeur *et al.*, “Localizing parts of faces using a consensus of exemplars,” in *CVPR*, 2011.
- [15] C. Sagonas *et al.*, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *ICCV-W 2013*, pp. 397–403, 2013.
- [16] X. Zhu *et al.*, “Face detection, pose estimation, and landmark localization in the wild,” in *CVPR*, 2012.