

# The Menpo Project

## Overview



The Menpo Project [1] is a BSD-licensed set of tools and software designed to provide an end-to-end pipeline for collection and annotation of image and 3D mesh data. In particular, the Menpo Project provides tools for annotating images and meshes with a sparse set of fiducial markers that we refer to as landmarks. For example, Figure 1 shows an example of a face image that has been annotated with 68 2D landmarks. These landmarks are useful in a variety of areas in Computer Vision and Machine Learning including object detection, deformable modelling and tracking. The Menpo Project aims to enable researchers, practitioners and students to easily annotate new data sources and to investigate existing datasets. Of most interest to the Computer Vision is the fact that The Menpo Project contains completely open source implementations of a number of state-of-the-art algorithms for **face detection** and **deformable model building**.

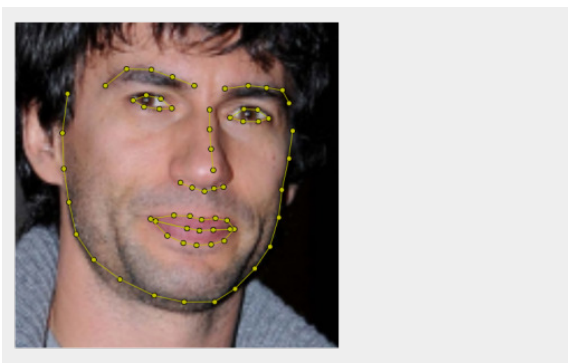


Figure 1. A facial image annotated with 68 sparse landmarks.

In the Menpo Project, we are actively developing and contributing to the state-of-the-art in **deformable modelling** [2], [3], [4], [5]. Characteristic examples of widely used state-of-the-art deformable model algorithms are **Active Appearance Models** [6],[7], **Constrained Local Models** [8], [9] and **Supervised Descent Method** [10]. However, there is still a noteworthy *lack of high quality open source software* in this area. Most existing packages are encrypted, compiled, non-maintained, partly documented, badly structured or difficult to modify. This makes them unsuitable for adoption in cutting edge scientific research. Consequently, research becomes even more difficult since performing a fair comparison between existing methods is, in most cases, infeasible. For this reason, we believe the Menpo Project represents an important contribution towards open science in the area of deformable modelling. We also believe it is important for deformable modelling to move beyond the established area of facial annotations and to extend to a wide variety of deformable object classes. We hope Menpo can accelerate this progress by providing all of our tools completely free and permissively licensed.

## Project Structure

The core functionality provided by the Menpo Project revolves around a powerful and flexible cross-platform framework written in **Python**. This framework has a number of subpackages, all of which rely on a core package called `menpo`. The specialised subpackages are all based on top of `menpo` and provide state-of-the-art Computer Vision algorithms in a variety of areas (`menpofit`, `menpodetect`, `menpo3d`, `menpowidgets`).

- **menpo** – This is a general purpose package that is designed from the ground up to make importing, manipulating and visualising image and mesh data as simple as possible. In particular, we focus on data that has been annotated with a set of sparse landmarks. This form of data is common within the fields of Machine Learning and Computer Vision and is a prerequisite for constructing deformable models. All `menpo` core types are `Landmarkable` and visualising these landmarks is a primary concern of the `menpo` library. Since landmarks are first class citizens within `menpo`, it makes tasks like masking images, cropping images within the bounds of a set of landmarks, spatially transforming landmarks, extracting patches around landmarks and aligning images simple. The `menpo` package has been downloaded more than 3000 times and we believe it is useful to a broad range of computer scientists.
- **menpofit** – This package provides all the necessary tools for training and fitting a large variety of state-of-

the-art deformable models under a unified framework. The methods can be roughly split in three categories:

1. *Generative Models*: This category includes implementations of all variants of the Lucas-Kanade alignment algorithm [6], [11], [2], Active Appearance Models [7], [12], [13], [2], [3] and other generative models [14], [4], [5].
2. *Discriminative Models*: The models of this category are Constrained Local Models [8] and other closely related techniques [9].
3. *Regression-based Techniques*: This category includes the commonly-used Supervised Descent Method [10] and other state-of-the-art techniques [15], [16], [17].

The `menpofit` package has been downloaded more than 1000 times.

- **menpodetect** – This package contains methodologies for performing generic object detection in terms of a bounding box. Herein, we do not attempt to implement novel techniques, but instead wrap existing projects so that they integrate natively with `menpo`. The current wrapped libraries are DLib, OpenCV, Pico and ffd2.
- **menpo3d** – Provides useful tools for importing, visualising and transforming 3D data. `menpo3d` also provides a simple OpenGL rasteriser for generating depth maps from mesh data.
- **menpowidgets** – Package that includes Jupyter widgets for ‘fancy’ visualization of `menpo` objects. It provides user friendly, aesthetically pleasing, interactive widgets for visualising images, pointclouds, landmarks, trained models and fitting results.

The Menpo Project 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 important for reproducible open science. Python provides a flexible environment for performing research, and recent innovations such as the Jupyter notebook 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 flexibility that the Menpo Project offers.

Note the Menpo Project has benefited enormously from the wealth of scientific software available with the Python ecosystem! The Menpo Project borrows from the best of the scientific software community wherever

possible (e.g. scikit-learn, matplotlib, scikit-image, PIL, VLFeat, Conda) and the Menpo team have contributed patches back to many of these projects.

## Getting Started

We, as the Menpo team, are firm believers in making installation as simple as possible. The Menpo Project is designed to provide a suite of tools to solve a complex problem and therefore has a complex set of 3rd party library dependencies. The default Python packing environment does not make this an easy task. Therefore, we evangelise the use of the Conda ecosystem. In our website, we provide detailed step-by-step instructions on how to install Conda and then Menpo on all platforms (Windows, OS X, Linux) (please see <http://www.menpo.org/installation/>). Once the conda environment has been set up, installing each of the various Menpo libraries can be done with a **single command**, as:

```
$ source activate menpo
(menpo) $ conda install -c menpo menpofit
(menpo) $ conda install -c menpo menpo3d
(menpo) $ conda install -c menpo menpodetect
```

As part of the project, we maintain a set of Jupyter notebooks that help illustrate how Menpo should be used. The notebooks for each of the core Menpo libraries are kept inside their own repositories on our Github page, i.e. `menpo/menpo-notebooks`, `menpo/menpofit-notebooks` and `menpo/menpo3d-notebooks`. If you wish to view the static output of the notebooks, feel free to browse them online following these links: `menpo`, `menpofit` and `menpo3d`. This gives a great way to passively read the notebooks without needing a full Python environment. Note that these copies of the notebook are tied to the latest development release of our packages and contain only static output and thus cannot be run directly – to execute them you need to download them, install Menpo, and open the notebook in Jupyter.

## Usage Example

Let us present a simple example that illustrates how easy it is to manipulate data and train deformable models using Menpo. In this example, we use annotated data to train an Active Appearance Model (AAM) for faces. This procedure involves four steps:

1. Loading annotated training images
2. Training a model
3. Selecting a fitting algorithm
4. Fitting the model to a test image

Firstly, we will load a set of images along with their annotations and visualize them using a widget. In order to save memory, we will crop the images and convert them to greyscale. For an example set of images, feel free to download the images and annotations provided by [18] from here. Assuming that all the image and PTS annotation files are located in `/path/to/images`, this can be easily done as:

```
import menpo.io as mio
from menpoviz import visualize_images

images = []
for i in mio.import_images('/path/to/images',
    verbose=True):
    i = i.crop_to_landmarks_proportion(0.1)
    if i.n_channels == 3:
        i = i.as_grayscale()
    images.append(i)

visualize_images(images) # widget for visualising
    the images and their landmarks
```

An example of the `visualize_images` widget is shown in Figure 2.

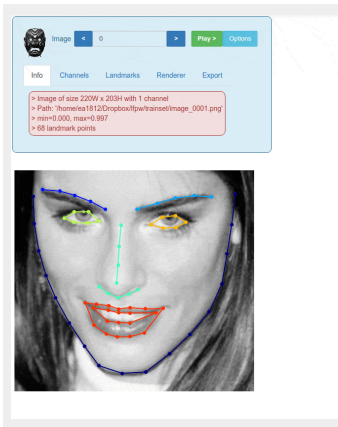


Figure 2. Visualising images inside Menpo is highly customizable (within a Jupyter notebook)

The second step involves training the Active Appearance Model (AAM) and visualising using an interactive widget. Note that we use Image Gradients Orientations [13], [11] features to help improve the performance of the generic AAM we are constructing. An example of the output of the widget is shown in Figure 3.

```
from menpofit.aam import HolisticAAM
from menpo.feature import igo

aam = HolisticAAM(images, holistic_features=igo,
    verbose=True)

print(aam) # print information regarding the model
aam.view_aam_widget() # visualize aam with an
    interactive widget
```

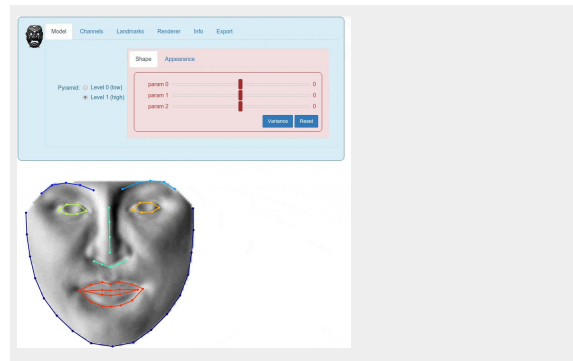


Figure 3. Many of the base Menpo classes provide visualisation widgets that allow simple data exploration of the created models. For example, this widget shows the joint texture and shape model of the previously created AAM.

Next, we need to create a `Fitter` object for which we specify the Lucas-Kanade algorithm to be used, as well as the number of shape and appearance PCA components.

```
from menpofit.aam import LucasKanadeAAMFitter

fitter = LucasKanadeAAMFitter(aam, n_shape=[5, 15],
    n_appearance=0.6)
```

Assuming that we have a `test_image` and an initial `bounding_box`, the fitting can be executed and visualized with a simple command as:

```
from menpoviz import visualize_fitting_result

fitting_result = fitter.fit_from_bb(test_image,
    bounding_box)
visualize_fitting_result(fitting_result) #
    interactive widget to inspect a fitting result
```

An example of the `visualize_fitting_result` widget is shown in Figure 4.

Now we are ready to fit the AAM to a set of `test_images`. The fitting process needs to be initialized with a bounding box, which we retrieve using the DLib face detector that is provided by `menpodetect`. Assuming that we have imported the `test_images` in the same way as shown in the first step, the fitting is as simple as:

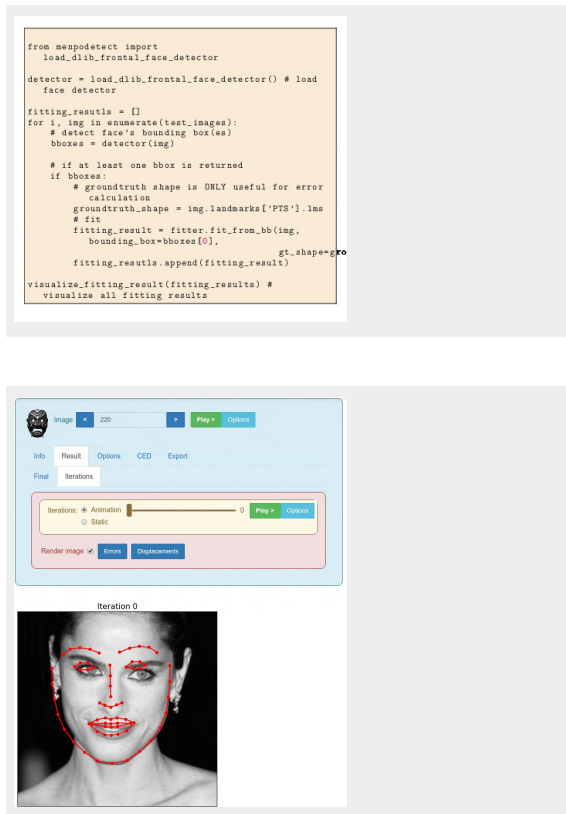


Figure 4. Once fitting is complete, Menpo provides a customizable widget that shows the progress of fitting a particular image.

#### Web Based Landmarker

URL: <https://www.landmarker.io/>

**landmarker.io** is a web application for annotating 2D and 3D data, initially developed by the Menpo Team and then heavily modernised by Charles Lirsac. It has no dependencies beyond a modern web browser and is designed to be simple and intuitive to use. It has several exciting features such as Dropbox support, snap mode (Figure 6) and easy integration with the core types provided by the Menpo Project. Apart from the Dropbox mode, it also supports a server mode, in which 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 5.

The application is designed in such a way to allow for efficient manual annotation. The user can also annotate any object class and define their own template of landmark labels. Most importantly, the

decentralisation of the landmarking software means that researchers can recruit annotators by simply directing them to the website. We strongly believe that this is a great advantage that can aid towards acquiring large databases of correctly annotated images for various object classes. In the near future, the tool will support a semi-assisted annotation procedure, for which Menpo will be used to provide initial estimations of the correct points for the images and meshes of interest.

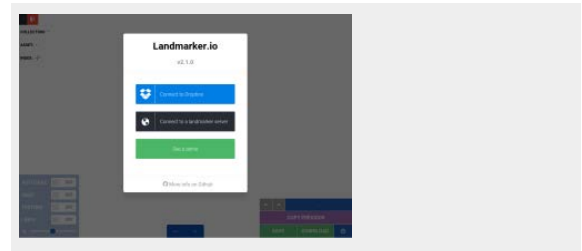


Figure 5. The landmarker provides a number of methods of importing assets, including from Dropbox and a custom Menpo server.

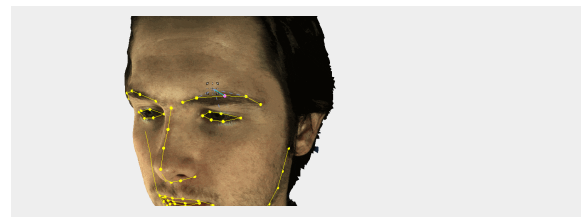


Figure 6. The landmarker provides an intuitive snap mode that enables the user to efficiently edit a set of existing landmarks.

[/caption]

## Conclusion and Future Work

The research field of rigid and non-rigid object alignment lacks of high-quality open source software packages. Most researchers release code that is not easily reusable, which further makes it difficult to compare existing techniques in a fair and unified way. Menpo aims to fill this gap and give solutions to these problems. We put a lot of effort on making Menpo a solid platform from which researchers of any level can benefit. Note that Menpo is a rapidly changing set of software packages that attempts to keep track of the recent advances in the field. In the future, we aim to add even more state-of-the-art techniques and increase our support for 3D deformable models [19]. Finally, we plan to develop a separate benchmark package that will standardize the way comparisons between various methods are performed.

Note that by the time this article was released, the versions of the Menpo packages were as follows:

- menpo 0.6.0
- menpofit 0.3.0
- menpo3d 0.2.0
- menpodetect 0.3.0
- menpowidgets 0.1.0
- landmarker.io 0.2.1

If you have any questions regarding Menpo, please let us know on the menpo-users mailing list.

## References

- [1] J. Alabort-i-Medina, E. Antonakos, J. Booth, P. Snape, and S. Zafeiriou, “Menpo: A comprehensive platform for parametric image alignment and visual deformable models,” in *Proceedings Of The ACM International Conference On Multimedia*, 2014, pp. 679–682. <http://doi.acm.org/10.1145/2647868.2654890>
- [2] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou, “Feature-based lucas-kanade and active appearance models,” *Image Processing, IEEE Transactions on*, 2015. <http://dx.doi.org/10.1109/TIP.2015.2431445>
- [3] J. Alabort-i-Medina and S. Zafeiriou, “Bayesian active appearance models,” in *Computer Vision And Pattern Recognition (CVPR), 2014 IEEE Conference On*, 2014, pp. 3438–3445. <http://dx.doi.org/10.1109/CVPR.2014.439>
- [4] J. Alabort-i-Medina and S. Zafeiriou, “Unifying holistic and parts-based deformable model fitting,” in *Computer Vision And Pattern Recognition (CVPR), 2015 IEEE Conference On*, 2015, pp. 3679–3688. [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Alabort-i-Medina\\_Unifying\\_Holistic\\_and\\_2015\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Alabort-i-Medina_Unifying_Holistic_and_2015_CVPR_paper.pdf)
- [5] E. Antonakos, J. Alabort-i-Medina, and S. Zafeiriou, “Active pictorial structures,” in *Computer Vision And Pattern Recognition (CVPR), 2015 IEEE Conference On*, 2015, pp. 5435–5444. [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Antonakos\\_Active\\_Pictorial\\_Structures\\_2015\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Antonakos_Active_Pictorial_Structures_2015_CVPR_paper.pdf)
- [6] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework,” *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004. <http://dx.doi.org/10.1023/B:VISI.0000011205.11775.f4>
- [7] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004. <http://dx.doi.org/10.1023/B:VISI.0000029666.37597.d3>
- [8] J. M. Saragih, S. Lucey, and J. F. Cohn, “Deformable model fitting by regularized landmark mean-shift,” *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011. <http://dx.doi.org/10.1007/s11263-010-0380-4>
- [9] A. Asthana, S. Zafeiriou, G. Tzimiropoulos, S. Cheng, and M. Pantic, “From pixels to response maps: Discriminative image filtering for face alignment in the wild,” 2015. <http://dx.doi.org/10.1109/TPAMI.2014.2362142>
- [10] X. Xiong and F. De la Torre, “Supervised descent method and its applications to face alignment,” in *Computer Vision And Pattern Recognition (CVPR), 2013 IEEE Conference On*, 2013, pp. 532–539. <http://dx.doi.org/10.1109/CVPR.2013.75>
- [11] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “Robust and efficient parametric face alignment,” in *Computer Vision (ICCV), 2011 IEEE International Conference On*, 2011, pp. 1847–1854. <http://dx.doi.org/10.1109/ICCV.2011.6126452>
- [12] G. Papandreou and P. Maragos, “Adaptive and constrained algorithms for inverse compositional active appearance model fitting,” in *Computer Vision And Pattern Recognition (CVPR), 2008 IEEE Conference On*, 2008, pp. 1–8. <http://dx.doi.org/10.1109/CVPR.2008.4587540>
- [13] G. Tzimiropoulos, J. Alabort-i-Medina, S. Zafeiriou, and M. Pantic, “Active orientation models for face alignment in-the-wild,” *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 12, pp. 2024–2034, 2014. <http://dx.doi.org/10.1109/TIFS.2014.2361018>
- [14] G. Tzimiropoulos and M. Pantic, “Gauss-newton deformable part models for face alignment in-the-wild,” in *Computer Vision And Pattern Recognition (CVPR), 2014 IEEE Conference On*, 2014, pp. 1851–1858. <http://dx.doi.org/10.1109/CVPR.2014.239>
- [15] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, “Incremental face alignment in the wild,” in *Computer Vision And Pattern Recognition (CVPR), 2014 IEEE Conference On*, 2014, pp. 1859–1866. <http://dx.doi.org/10.1109/CVPR.2014.240>
- [16] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *Computer Vision And Pattern Recognition (CVPR), 2014 IEEE Conference On*, 2014, pp. 1867–1874. <http://dx.doi.org/10.1109/CVPR.2014.241>



[17] G. Tzimiropoulos, “Project-out cascaded regression with an application to face alignment,” in *Computer Vision And Pattern Recognition (CVPR), 2015 IEEE Conference On*, 2015, pp. 3659–3667. [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Tzimiropoulos\\_Project-Out\\_Cascaded\\_Regression\\_2015\\_CVPR\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Tzimiropoulos_Project-Out_Cascaded_Regression_2015_CVPR_paper.pdf)

[18] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference On*, 2013, pp. 397–403. [http://www.cv-foundation.org/openaccess/content\\_iccv\\_workshops\\_2013/W11/papers/Sagonas\\_300\\_Faces\\_in-the-Wild\\_2013\\_ICCV\\_paper.pdf](http://www.cv-foundation.org/openaccess/content_iccv_workshops_2013/W11/papers/Sagonas_300_Faces_in-the-Wild_2013_ICCV_paper.pdf)

[19] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” in *Proceedings Of The 26th Annual Conference On Computer Graphics And Interactive Techniques*, 1999, pp. 187–194. <http://dx.doi.org/10.1145/311535.311556>

1. Alphabetical author order signifies equal contribution#
2. Currently unreleased – the next released versions of `menpo`, `menpofit` and `menpodetect` will reflect these version numbers. All samples were written using the current development versions.#

**Authors:** Joan Alabort-i-Medina<sup>1</sup>, Epameinondas Antonakos<sup>1</sup>, James Booth<sup>1</sup>, Patrick Snape<sup>1</sup>, and Stefanos Zafeiriou  
**Affiliation:** Department of Computing, Imperial College London, UK  
**URLs:** <http://www.menpo.org/>, <https://github.com/menpo/>

## SIGMM PhD Thesis Award — Call for nominations

**SIGMM Award for Outstanding PhD Thesis in Multimedia Computing, Communications and Applications**

### Award Description

This award will be presented at most once per year to a researcher whose PhD thesis has the potential of very high impact in multimedia computing, communication and applications, or gives direct evidence of such

impact. A selection committee will evaluate contributions towards advances in multimedia including multimedia processing, multimedia systems, multimedia network services, multimedia applications and interfaces. The award will recognize members of the SIGMM community and their research contributions in their PhD theses as well as the potential of impact of their PhD theses in multimedia area. The selection committee will focus on candidates' contributions as judged by innovative ideas and potential impact resulting from their PhD work. The award includes a US\$500 honorarium, an award certificate of recognition, and an invitation for the recipient to receive the award at a current year's SIGMM-sponsored conference, the ACM International Conference on Multimedia (ACM Multimedia). A public citation for the award will be placed on the SIGMM website, in the SIGMM Records e-newsletter as well as in the ACM e-newsletter.

### Funding

The award honorarium, the award plaque of recognition and travel expenses to the ACM International Conference on Multimedia will be fully sponsored by the SIGMM budget.

### Nomination Applications

Nominations will be solicited by the 31<sup>st</sup> May 2016 with an award decision to be made by August 30. This timing will allow a recipient to prepare for an award presentation at ACM Multimedia in that Fall (October/November). The initial nomination for a PhD thesis must relate to a dissertation deposited at the nominee's Academic Institution between January and December of the year previous to the nomination. As discussed below, /some dissertations may be held for up to three years by the selection committee for reconsideration/. If the original thesis is not in English, a full English translation must be provided with the submission. Nominations for the award must include:

1. PhD thesis (upload at: <https://cmt.research.microsoft.com/SIGMMA2016/>)
2. A statement summarizing the candidate's PhD thesis contributions and potential impact, and justification of the nomination (two pages maximum);
3. Curriculum Vitae of the nominee
4. Three endorsement letters supporting the nomination including the significant PhD thesis contributions of the candidate. Each endorsement should be no longer than 500 words with clear specification of nominee PhD thesis contributions and potential impact on the multimedia field.
5. A concise statement (one sentence) of the PhD thesis contribution for which the award is being given. This