AAM – What to do?

1. Build new model (see section: building new models).
2. Find some objects in the set of test images (3-5 pictures). If the suggested position of the object is not proper for object matching, move the view of the model (use command: Move all)
3. Improve your model, using test images.
4. Find objects in the rest of test images. (Attention: if you cannot view move model to the accepted position, restart the program am_markup.)
5. In report (written in MS Word, or Wordpad):
    a. Include some images with landmarks, and found object on the "unseen" images.
    b. What means that the AAM is a generative model? Add illustrations.
    c. In you opinion: what is the biggest problem in using AAM to find skulls, faces, or cars?

AMM Modelling and Search Software[1]: Basic Tutorial

The data consists of three subdirectories:

- `images` - containing images,
- `points` - containing points annotated on each image,
- `models` - containing data files describing the models.

Building new models

Build a statistical shape/appearance model of the shape of the skulls / lips / cars (use proper object for your data).

To summarise:

Preliminaries

- Create a new directory for the points (eg skull_points[2]).

Annotating the first image

- Go into the models directory and run `am_markup start`.
- Load in the first face image.
- Click on "Add Curve" and then click points around the outline of the object (eg skull), starting at one corner.

[1] Software was developed by Tim Cootes from University of Manchester (http://www.isbe.man.ac.uk/~bim/)

[2] There are some different sets of files: skull, faces and cars. Use the appropriate name for your group.

- Complete the curve by clicking on the first point.
- Add curves defining the inner boundaries of the lips / skulls / cars by clicking on one corner point to start, and completing by clicking on the other corner point.
- When completed, save the points into the skull_points directory, with a name to match the image name.
- Save the parts to "skull.parts" in the models directory. (These define the connectivity).
- Close am_markup

Building the first model

- Edit the file “skull.smd", changing the model name, the parts file name, points directory as appropriate. Put the name of the labelled image and associated points file in the "training_set" section.
- Use `am_build_apm skull` to create an initial model (from a single example).
- Edit the model names as appropriate in "skull.aamprops".
- Use `am_build_aam skull` to build an AAM.

Annotating further images

- From the models directory, run `am_markup skull`
- Use "File->Load Image Set" to load in an SMD file describing all images to annotate(eg skulltest.smd)
- Use the arrow button above the main window to move to the next image (the first image has been annotated.
- Use the "Props->Search Props" menu item to get the search panel up.
- Run the “search” to match the model to the image.
- Edit the point positions using either the "Move All" mode, “Warp points”, or the "Move One" mode.
- When happy, press the "Save,Add,Update" button, to save the points and update the training data SMD file automatically.
- Move to the next image.
- Periodically close the program and rebuild the model + AAM. As more examples are added, the model should match to new images better.
- At any point, use am_view_shape_model to display the shape model or am_view_apm to display the appearance model.

And there you have it - your own lovely skull model.

For additional information see «How to start building models».

How to start building models

Preparatory Tasks

- Create a project directory (directory and file names must not have spaces), e.g. `group1`

- Create three subdirectories for models (e.g. `models`), point files (e.g. `points`) and images `images`.
- Copy templates of the following files to the `models` directory: xxxx.smd,xxxx.aamprops. Replace xxx with an appropriate model name, e.g. `cars`.
- Edit the .smd file and add the model name (e.g. cars),the parts file name (e.g. cars) and the various required directory names. Also add one pair of point/image files under training_set. The image file is the first image to be analyzed.
- Edit the .aamprops file and add the model name (e.g. cars) to am_name and aam_name and the directory name for the models.
- Create a dummy points file called "`points/dummy.pts`" as follows:

```
version: 1
n_points: 2
{
0.0 0.0
1.0 1.0
}
```

Annotating the First Image

It is assumed that you are running a shell or command prompt. Change directory to the `models` directory you have created.
See also the details on the am_markup start program.

- Run the `am_markup start` program, invoking it with the name of the SMD file (eg "`am_markup start cars`")
- Choose Add Curve and mark the first point contour. Position a final point on the first point to close the contour.
- Choose Move One to adjust the location of individual points. Use other tools as indicated.
- Create additional contours as required. If the desire is to build an active shape model, the contours around the desired shapes are sufficient. For an active appearance model, area around the shapes also needs to be identified.
- Save the point contours by choosing "File>Save Points As" select the above created points subdirectory and use the filename of the image with the extension .pts.
- Save the contour descriptions by choosing File>Save Parts As; select the above created models subdirectory and use the filename of the model with the extension .parts, e.g. cars.parts.
- Exit am_markup.

Create a First Model

- Run the appearance model builder with the model name (e.g. "`am_build_apm cars`")
- Build the active appearance model by running "`am_build_aam cars`".
- The appearance model can be viewed with am_view_apm cars.

Annotating Further Images

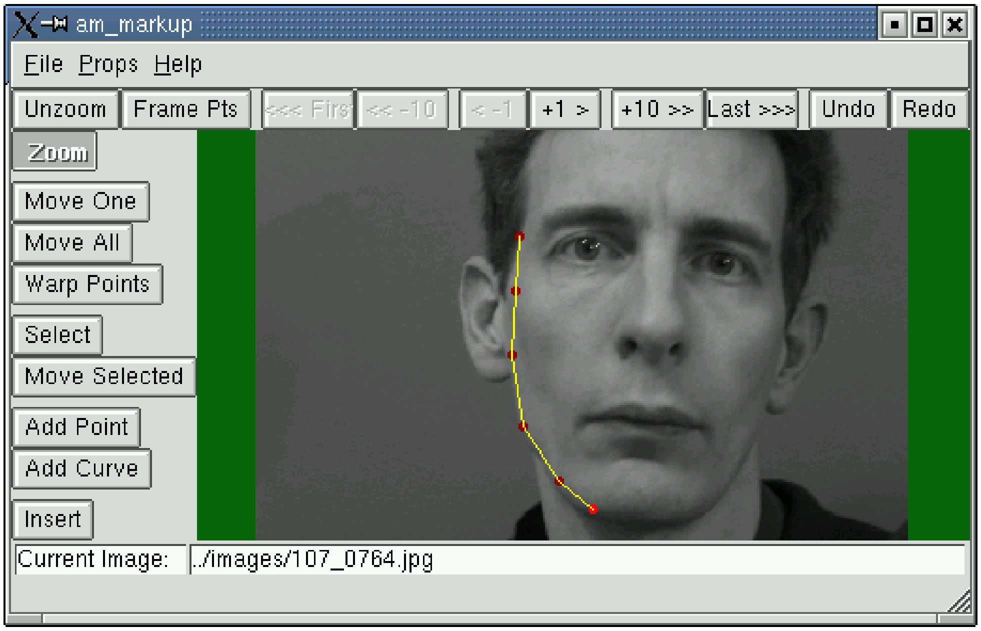
- Create an SMD file (called, for instance `data_todo.smd`) containing the list of all images you wish to work on. Set the points files to "dummy.pts". This simply makes it easier to go through the data.
- Run am_markup again (e.g. `am_markup cars`). The first image previously marked and its model in yellow are displayed in the window.
- Use the "File->Load Image Set" to load in the list of files to annotate (data_todo.smd)
- Enlarge the model-point sizes (green) by going to Props>Graphics Props and press the + button to the right of Model Point Radius repeatedly.
- Use the "Props->Search Props" menu item to get the search panel.
- Press the "Move All" button on the bar to the left to go to a mode in which the points can be moved (drag the cursor with left button) or scaled/rotated (drag with right button). Move the points to roughly the right position.
- Run the search using the "Search" button on the search panel.
- Edit the points (eg by going into "Move One" mode) until satisfied.
- Press the button "Save,Add,Update" or "Save,Add,Update,Next,Labels,Frame". The latter button will automatically load the next image on the list.
- Go through the remaining images and exit am_markup.

The search will not work very well if trained with only a single image. It is recommended that one rebuild the model after adding a few images. To do this, exit am_markup and re-run the am_build_apm/am_build_aam tools. The model SMD file will have been automatically updated to include the images you have annotated (but it is worth checking it to make sure there aren't duplicates or any problems). Delete the annotated files from the data_todo.smd file. Then re-run am_markup and continue as above.

As more images are added the model becomes more flexible and better able to generalise.

am_markup start

A tool for annotating images and creating connectivity (.parts) files.



Current Image: ../images/107_0764.jpg

Invoke from the command line with

```
am_markup start
```

Loading images to annotate

To load a single image, use the "File->Load Image" menu item.

We are so often interested in sets of images, possibly with existing annotations. The simplest method of dealing with them is to set up a basic SMD file listing the images of interest. This can then be loaded using the "File->Load Image Set". The first image (and any associated annotation points listed in the SMD file) will then be displayed. You can move through the set using the arrow buttons above the main window. This is the preferred technique when marking up a set of images, as the tool can automatically create a suitable file name for any saved points.

Manipulating the Points

The buttons on the left of the main window allow selection of a variety of interactions.
The main methods of adding points are as follows:

- Add Point: Each click adds a point (unconnected with others)
- Add Curve: First click adds a point to start a curve. Subsequent clicks add more points. Double click to complete (open) curve, or click on the first point in the curve to create a closed curve.

You can also interact with the image and points as follows:

- Zoom: Select a box to zoom in. Double click or hit the "Unzoom" button to display the whole image

- Move All: Dragging with the left mouse button translates the model, dragging with the right mouse button rotates and scales the model about its centre
- Select: Press and drag to select all points in the resulting box
- Move Selected: As "Move All", but only affects selected points
- Move One: Click near a single model point to drag it around

Saving Results

- The current positions of the points can be saved to a points file using the "File->Save Points" menu item.
- The current connectivity can be saved to a parts file using the "File->Save Parts" menu item.

Display Issues

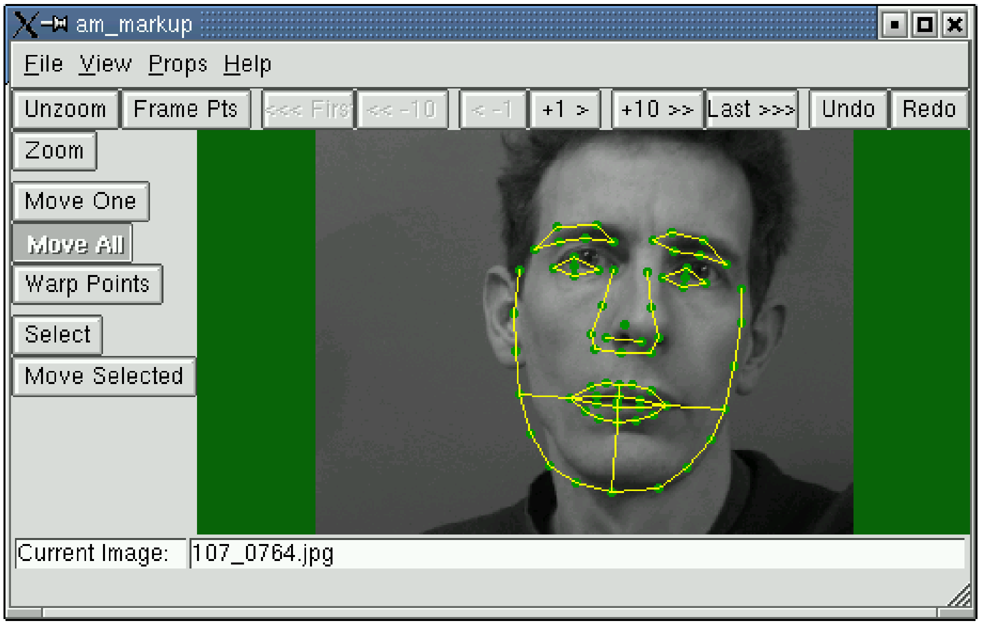
The "Props->Graphics" menu item brings up a panel allowing control of the graphics (line width, point radius etc)

Generating screenshots

- "File->Save Screenshot" will save the image in the current window, together with any graphical overlays.

am_markup

A tool for interactive search with Active Appearance Models.



Current Image: 107_0764.jpg

Invoke from the command line with

```
am_markup
```

or

```
    am_markup xxxx
```

where `xxxx.smd` is an SMD file.

Loading a model

The filenames defining the various components of each model are encapsulated by the SMD file.

To load in a model, either pass in the SMD file path as a parameter when running the command, or use the "File->Load Model" menu option.

Loading images to search

To load a single image, use the "File->Load Image" menu item.

We are so often interested in sets of images, possibly with existing annotations. The simplest method of dealing with them is to set up a basic SMD file listing the images of interest. This can then be loaded using the "File->Load Image Set". The first image (and any associated annotation points listed in the SMD file) will then be displayed. You can move through the set using the arrow buttons above the main window. This is the preferred technique when marking up a set of images, as the tool can automatically create a suitable file name for any saved points.

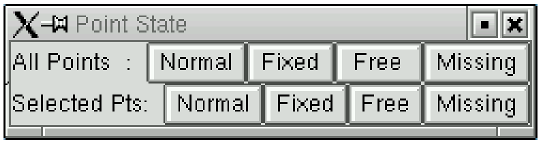
Note: When you load in a model, the SMD file defines an initial set of images, the first of which is automatically loaded.

Manipulating the Model

Because the AAM is a local search algorithm, the model must be started in roughly the correct position. The buttons on the left of the main window allow selection of a variety of interactions as follows:

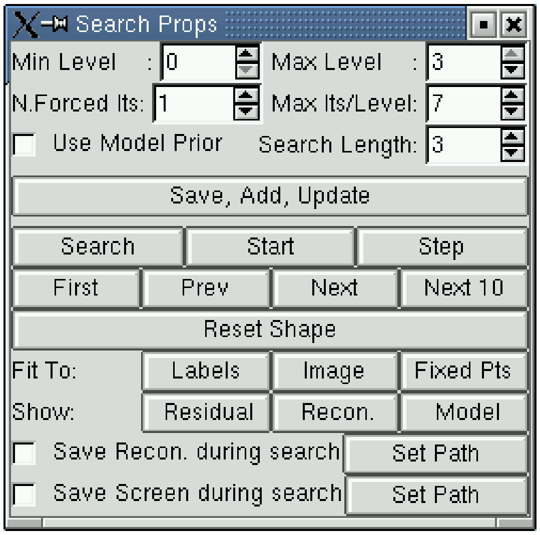
- Zoom: Select a box to zoom in. Double click or hit the "Unzoom" button to display the whole image
- Move All: Dragging with the left mouse button translates the model, dragging with the right mouse button rotates and scales the model about its centre
- Select: Press and drag to select all points in the resulting box
- Move Selected: As "Move All", but only affects selected points
- Move One: Click near a single model point to drag it around

Note that when you move individual points, they go red. This indicates they are "Fixed" and will not be moved during the search. The assumption is that if the user has moved them, they are moved to a particular desired location, and the machine would not be so presumtios as to move them again. To return them to a "Normal" state, use the buttons in the panel obtained using the "Props->Point State" menu item. Note: When a new image is loaded, all points return to a Normal state.



Searching with the Model

Use the "Props->Search" menu item to pop up a panel of search controls.



Min Level : 0
Max Level : 3
N.Forced Its: 1
Max Its/Level: 7
☐ Use Model Prior
Search Length: 3
☐ Save Recon. during search
☐ Save Screen during search

These allow you to define the parameters controlling the search (though the defaults will usually work reasonably).

To run the search, simply press the "Search" button.

Search parameters

- Max. Level: The coarsest model resolution to use. Defaults to the coarsest model resolution available.
- Min. Level: The finest model resolution to use
- Max. Its: The maximum number of iterations to use at each resolution
- Forced Its: The number of iterations to run without checking that the predicted step actually improves the match - setting to 1 or 2 has been found to be useful for jumping out of local minima.

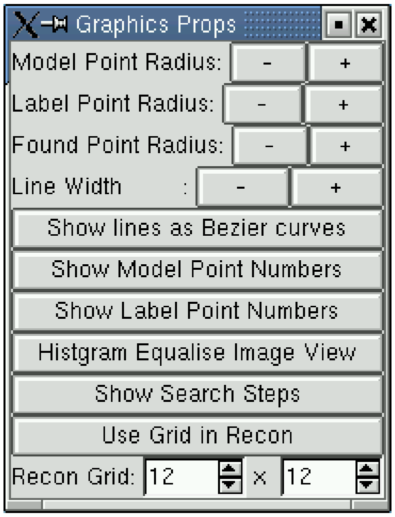
Saving Results

The current positions of the model points can be saved to a points file using the "File->Save Points" menu item.

When images are loaded using an SMD file, mechanisms exist for automatically generating an appropriate file name (based on the image name) and saving the points.

Display Issues

The "Props->Graphics" menu item brings up a panel allowing control of the graphics (line width, point radius etc)



Recon Grid: 12 x 12

- Displaying the results at each iteration can be enabled (but slows down the search).
- The "View Recon" button pops up a window showing the image with the current model reconstruction overlaid on it. Note: This is only sensible when the model is representing (and can thus synthesize) raw texture regions. It will not work well if the model represents gradients or other more complex features.

Generating screenshots and video sequences

- "File->Save Screenshot" will save the image in the current window, together with any graphical overlays.
- Use the buttons provided to allow saving of every screenshot during a search, so as to generate sequences of frames for videos.
- This can also be done with the reconstruction image.

Shape Model Data Files

A Shape Model Data (SMD) file describes all the information required to build a shape or appearance model.

SMD files require lists of corresponding points and image files.

Here's a typical SMD file:

```
// Text file describing data to build model
// Note that comments can appear almost anywhere

// Define the base model name and directory into which it will be put
model_dir: /bim/data/BMVC02/RawGrey_90/
```

```
model_name: isbe

// Define the base name of the parts file
// The file cars.parts will be loaded from the model directory
parts_file: cars

// Define the base name of the triangles file
// The file cars.tri will be loaded from the model directory
tri_file: cars

// Directory containing images
image_dir: /home/data/Faces/images/
// Directory containing points
points_dir: /home/data/Faces/points/

// shape_aligner can be None, align_cog_2d, align_similar_2d, align_affine_2d,...
shape_aligner: align_similar_2d

// Define how to choose the number of modes to retain for the
// shape, texture and combined appearance models
// The number will be chosen to explain proportion of the variance,
// constrained by min and max
shape_modes:  { min: 0 max: 999 prop: 0.90 }
tex_modes:  { min: 0 max: 999 prop: 0.90 }
combined_modes:  { min: 0 max: 999 prop: 0.999 }

// Define how the vector models will be limited to plausible values
// Valid limiters are mdpm_box_limits or mdpm_prob_limits.
// The default is mdpm_box_limits with sd_limit set to 3.0
// However the example given here is believed to be much better.
params_limiter: mdpm_prob_limits
{
  acceptance_prob: 0.99
}

// Number of pixels to represent with highest resolution texture model
n_pixels: 5000

colour: Grey //greyscale model

// Texture Sampler can be vapm_triangle_sampler<vxl_byte>,
// vapm_tri_grad_sampler<float>, vapm_profile_set_sampler, etc.
tex_sampler: vapm_profile_set_sampler
{
  profile_sampler: vsml_profile_sampler_2d<vxl_byte>
  prof_hi: 2
  prof_lo: -2
  width: 1
  step_size: 1.0
  use_gradient: 0
}

// Define the normalisation to be applied when building and using the texture model.
// tex_aligner can be None, align_linear_1d, ...
tex_aligner: align_linear_1d

// shape_wts define how to compute relative scaling of shape & tex.
// shape_wts can be `EqualVar', `EqualEffect',...
shape_wts: EqualVar
// Image Pyramid Builder can be gauss_byte, gauss_float, grad_float ...
pyr_builder: gauss_byte
```

```
// Define the maximum number of image pyramid levels to build
max_im_pyr_levels: 5

// Define the model resolution levels to build
// Level 0 contains  pixels
// Level 1 is half the size in each dimension
// Level 2 is half again etc
min_level: 0
max_level: 2

// Details of points : images
training_set:
{
zoel/zoel090o00.pts : zoel/zoel090o00.jpg
zaig/zaig090o00.pts : zaig/zaig090o00.jpg
warm/warm090o00.pts : warm/warm090o00.jpg
tonl/tonl090o00.pts : tonl/tonl090o00.jpg
timc/timc090o00.pts : timc/timc090o00.jpg
}
```

AAM Property Files

AAM Property (*.aamprops) files describe how to build an AAM, including various parameters necessary for the builder.
They are used by the build_aam tool.
Here's an example:

```
// Define type of AAM
// (basic_aam,shape_aam,comp_aam,direct_aam,raw_regression_aam,
// grad_regression_aam)
aam_type: basic_aam

// Define model directory and base name of APM and AAM files
model_dir: ./
am_name: isbe_comp
aam_name: isbe_comp

// Parameter displacements used during search
displacement_x: 2
displacement_y: 2
max_disp: 0.125
scaledisplacement_x: 0.05
scaledisplacement_y: 0.05
tex_displacement: 0.05
displacement_params: 0.5
n_per_eg: 2

// Maximum number of images to sample from (overrides that supplied in SMD)
max_n_samples: 10
rob_kernel: None
```

The types of AAM available include:

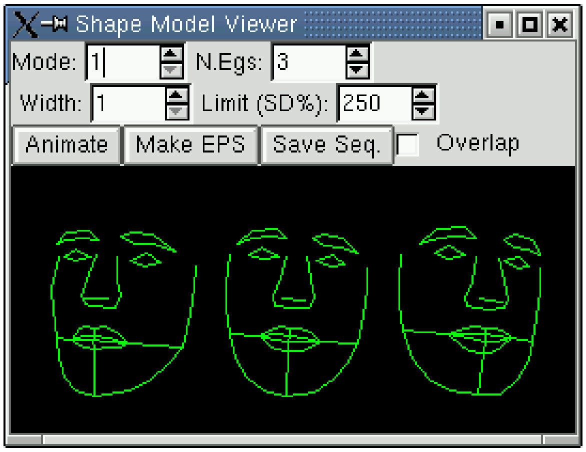
- basic_aam : Plain AAM, trained by estimating gradient d(residual)/d(param), as descibed in PAMI paper

- raw_regression_aam: Plain AAM but trained using regression (as described in early papers)
- grad_regression_aam: Plain AAM but trained using regression to estimate gradient (as described in BMVC02 paper)
- shape_aam: AAM in which linear shape update predicted from current texture error
- comp_aam: AAM in which shape update predicted from current texture error and applied using a compositional approach
- direct_aam: AAM in which shape predicted from current texture parameters

Recent experiments on faces (described in BMVC02 paper and related report) suggest that the grad_regression_aam is probably the best default option, though the basic_aam performs well and is quicker to train.
Surprisingly the shape_aam and comp_aam seem to match texture slightly better, though it is somewhat problem dependent.

am_view_shape_model

A tool to display the modes of variation of a shape model



Mode: 1 N.Egs: 3
Width: 1 Limit (SD%): 250
□ Overlap

It reads the shape model from the *.apm file (built by am_build_apm).
It also reads an associated parts file, describing how to "join the dots".

It has controls which allow you to

- Select the mode of interest
- Vary the number of examples showing that mode
- Vary the range of variation of the mode
- Vary the line thickness used to display the shapes

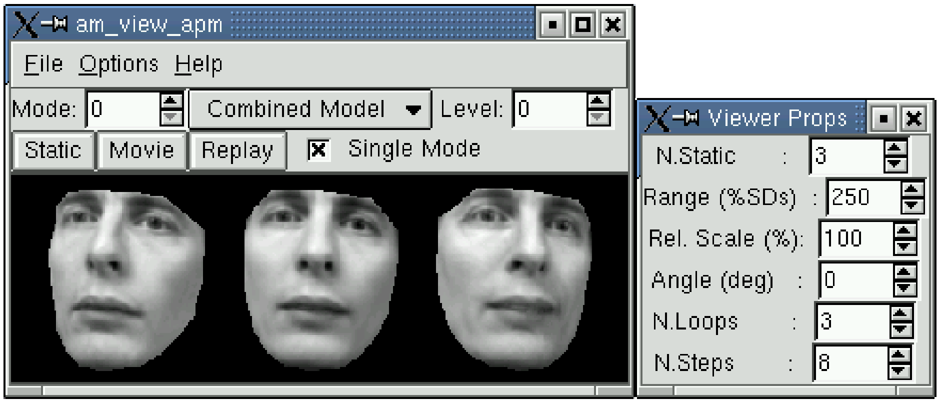
In addition it has buttons which allow you to

- Animate the shapes to show the variation in a movie

- Generate a Encapsulated Postscript (EPS) version of the current figure (useful for inclusion in documents)
- Save a sequence of images suitable for creating a movie (note: best to arrange to display only one shape when doing this)

am_view_apm

A tool to display the modes of variation of an appearance model



Mode: 0 Combined Model Level: 0
☒ Single Mode

N.Static : 3
Range (%SDs) : 250
Rel. Scale (%): 100
Angle (deg) : 0
N.Loops : 3
N.Steps : 8

It reads the shape model from the *.apm file (built by build_apm).

The tool allows the display of three types of model:

- A shape model (using the mean texture)
- A texture model (using the mean shape)
- A combined shape and texture model

It has controls which allow you to

- Select the mode of interest
- Vary the number of examples showing that mode
- Vary the range of variation of the mode

In addition it has buttons which allow you to

- Animate the model to show the variation in a movie
- Save a sequence of images suitable for creating a movie (note: best to arrange to display only one shape when doing this)

Images and files for groups

First group: skulls:

Model name: skull
SMD file: skull.smd

SMD file for tests (and learning): skull_test.smd
Images: /images/AXYY.JPG (YY=number)
Points – You must create points, using am_markup, and save in the points directory. As landmarks use the eye sockets, nose, chin or teeth. The landmarks have to be easily found on the images.
Attention: use small number of points for each “curve”.

Second group: cars

Model name: cars
SMD file: cars.smd
SMD file for tests (and learning): carstest.smd
Images: /images/carYY.jpg (YY=number, 1-35).
Points – You must create points, using am_markup, and save in the points directory.
As landmarks use the wheels, windows, doors, and the car outline.
Attention: use small number of points for each “curve”.
Practical remarks

Third group: light cars

Model name: cars
SMD file: cars.smd
SMD file for tests (and learning): carstest.smd
Images: /images/carYY.jpg (YY=number, 1-48).
Points – You must create points, using am_markup, and save in the points directory.
As landmarks use the wheels, windows, doors, and the car outline.
Comment: pictures for the second group, represents cars (side-view) of different colours, and in the third group only light cars.
Attention: use small number of points for each feature.

Fourth group: faces

Model name: faces
SMD file: face.smd
SMD file for tests (and learning): facetest.smd
Images: /images/faceYY.jpg (YY=number, 1-26).
Points – You must create points, using am_markup, and save in the points directory.
As landmarks use the lips, eyes, nose and face outline.
Attention: use small number of points for each feature.