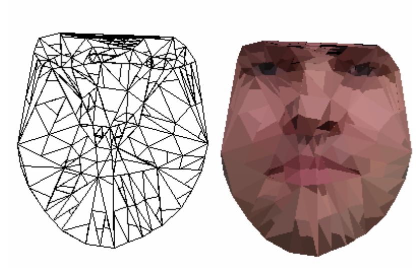
Wedgelet Compression for AAM Trainingset

By Sune Darkner



Wedgelets

Basically we want to do a multiscale representation of the texture component of the AAM. We use wedgelets as the basis for compression as suggested by David Donoho. The wedgelet-approach involves a very simple template which consists of three basic configurations as shown below in fig. 1.

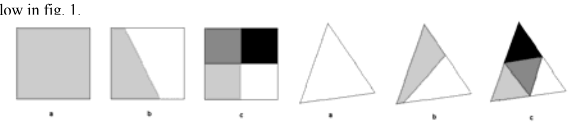


Fig. 1. *Dyadic basis and triangulated basis.*

The first template (name this template a) is just an area represented by its corners and its mean value. The second (name this template b) is a wedge, an area represented by its corners, an edge and two mean values. The third (name this template c) is actually just a collection of four of the two first templates. When using wedgelets we are looking for edges in an image. Since template a and c are trivial we have to somehow construct template b. This is done by doing and exhaustive search in the domain (se the fig. 2.) to find the edge that minimizes the residual sum of squares (RSS) by dividing the domain into to areas with different mean values. These are the buildingblocks of the wedgelet compression.

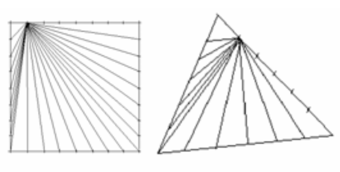


Fig. 2. *Possible edges from one point.*

The way it works

When employing the wedgelet basis we are able to embed the composition into a quad-tree structure where the nodes inside the tree are of type c template and the leaves are of type a and b. We are now able to construct a regression tree which we will fit to the image using the standard penalized residual sum of squares se eq. 1. Fig. 3 shows how a result on a binary image might look. David Donoho introduced this technique on a dyadic domain, but by using barycentric coordinates commonly known from computer graphics we can easily move to a triangulated domain and still maintain correspondence through the training set of the AAM.

$$PRSS = \|y - \mu\|^2 + \lambda \# P \qquad \text{Where:} \qquad \lambda = const \cdot \sigma^2$$

Eq. 1. *Penalized sum of squares*

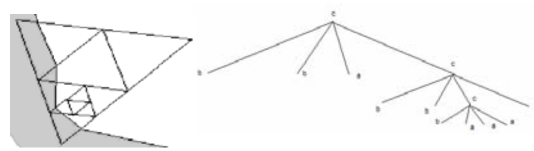


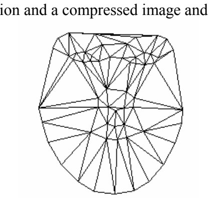
Fig. 3. *Wedgelet representation and corresponding quad-tree.*

Compression results

Fig. 4. Show the result of compressing an image. Since the basis of the compression is triangular and the image is squared the initialization is to split the image into two along the diagonal from the upper left corner to the lower right.

Compression results using AAM annotation

The following results have been obtained using the annotation for the AAM. This means that we have a minimum triangulation of 95 triangles which are compressed individually. A tree-formulation of this is simply that we have a root with 95 branches. Fig. 5. show the initial triangulation and a compressed image and its wire frame. The level of compression is roughly 97%.



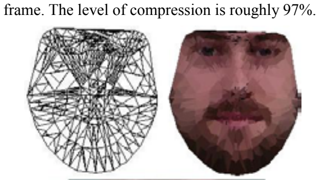


Fig. 5. *Left the initial triangulation. Right the compressed image that consists of 494 triangles with 295 wedges. Initial area contains approximatly 30000 pixels giving a compression of 97%.*

Informatik og Matematisk Modellering

Applying it to the AAM

Since the area of application is the training set for the AAM we have to make sure that each image is represented in the same way. If we just compress the images individually we get a different tree for each image se Fig. 6. This means that the constructed trees must be isomorphic. There are several ways to ensure this. The statistical approach is to create the tree-structure for the compression for all the images at once. This will also enable us to calculate lambda in eq. 1. using cross validation. This would give a plausible result. A second approach is to view the problem at hand as a pure graph-matching problem. In terms this means that we could separately compress each image and then find the largest isomorphic sub-tree. The last idea is instead of just finding the largest isomorphic sub-tree we could, assuming that we define a suitable cost function, reshape, grow and collapse branches and nodes find the minimum cost deformation. This last approach is very close to the regression approach.

Fig. 6. *Left four non-isomorphic imag-trees. Right four isomorphic image-trees*

Conclusion and discussion

The results show that this kind of intelligent compression can be used with success to compress images. Furthermore it keeps a high level of detail in areas with high information and low resolution in areas with low information. Further more we get a reduction in noise for free. The results so far are very promising.

Future work

We want to implement all three approaches described in the previous section and test them against the result achieved by Mikkel B. Stegmann using wavelets. Furthermore it has been suggested by Ali Shokoufandeh that by fitting this approach into a certain framework an almost complete decompression might be achievable. Finally we want to introduce a region excluder for noisy regions and a constraint that favors edges across boundaries.

References:

David Donoho: "Wedgelets: Nearly-Minimax Estimation of Edges" Stanford University and U. C. Berkley Aug 1997

Acknowledgements:

Rasmus Larsen and Mikkel B. Stegmann

Further information: *www.imm.dtu.dk/~proj28* ***or*** *sune@darkner.dk*