01/05/2014

**Hard K-means vs Fuzzy K-means**

* Input silhouettes:



**Shape Context descriptor** applied on my silhouette image .. (SC2\_demo.m) .( Ref: Malik paper named "Shape Matching and Object Recognition Using Shape Contexts"(PAMI 2002))

Shape descriptor as a silhouette at a single sample point produces a result of N no of 12\*5 bins like this:



What we need is a single descriptor to describe the whole of this representation.

Just to visulize the impact of all these shape descriptors on the silhouette,just concatenated each row of every 12\*5 bins and formed 60 binned feature descriptor. Now, did the same on every shape descriptor and thus gets a 40\*60 histogram like this.



Note that the first two histograms were from the same person and the last two are from another person. We can visualize and interpret the same from these histograms.

Still, our idea is yet to compress these 40\*60 histograms and make a single feature vector .So, we are adopting the idea from Bill triggs paper entitled'Recovering 3D Human Posefrom Monocular Images'(PAMI 2006) ( http://lear.inrialpes.fr/people/triggs/pubs/Agarwal-pami05.pdf) .

As mentioned in the paper,a second level of histogramming is performed on the 60D histograms via **vector** quantization technique. Thus, the shape context space is vector quantized and the distribution of each silhouette id redueced to an N dimensional histogram. The N center codebook is learned once and for all by running k-means on the combined set of context vectors of all of the training silhouettes.

1) **Hard-K means** based vector quantizatioin.

In which the contribution of each point to each cluster is either 0 or 1.

( Ref: KmeansdemoVQ\_athira.m)

Here, N is selected as 10First 2 are similar and the last 2 are similar, which validates our sanity check.



Fig: Hard-K means based vector quantizatioin.

Now, some problems I came across while dealing with Hard K-means VQ :

1) Each execution with K-means is initialized with different initial codebooks(centroids) each time (random initialization) and hence results in diffrent results of clustering.

* To resolve this issue**, 'Replicates'** is used. It shows the number of times to repeat the clustering, each with a new set of initial cluster centroid positions. kmeans returns the solution with the lowest value for sumd, where 'sumd' is the within-cluster sums of point-to-centroid distances.
* Optimal value of 'nClus' ? Too less or too much is not good!
* K-means won't work very well with non-globular(dont have well defined structures--- chainlike ) clusters...

2) **Fuzzy-K means** based vector quantizatioin.

Here, a feature vector x can have a degree of membership in each cluster. It makes more sense to say that x is 40% in cluster 1 and 60% in cluster2.

(Ref: test\_fuzme\_ATHIRA.m)

The performance of K-means and Fuzzy K-means are compared and the results:



Fig: Hard/Crispy K-means VQ



Fig: Soft/ Fuzzy K-means VQ

in order to produce a confusion matrix like this.



**To do:**

1) How many times the correct result was produced? seems like fuzzy results are more stable.

(Try to make a loop of runs and get the accuracy in each run)

**Hard K-means in 10 trials:**

Execution no= 1 NN Classifier Accuracy: 66.67%

Execution no= 2 NN Classifier Accuracy: 66.67%

Execution no= 3 NN Classifier Accuracy: 100.00%

Execution no= 4 NN Classifier Accuracy: 66.67%

Execution no= 5 NN Classifier Accuracy: 100.00%

Execution no= 6 NN Classifier Accuracy: 100.00%

Execution no= 7 NN Classifier Accuracy: 100.00%

Execution no= 8 NN Classifier Accuracy: 66.67%

Execution no= 9 NN Classifier Accuracy: 100.00%

Execution no= 10 NN Classifier Accuracy: 66.67%

**Fuzzy K-means in 10 trials:**

NN Classifier Accuracy: 100.00%

Execution no= 1

NN Classifier Accuracy: 100.00%

Execution no= 2

NN Classifier Accuracy: 100.00%

Execution no= 3

NN Classifier Accuracy: 100.00%

Execution no= 4

NN Classifier Accuracy: 100.00%

Execution no= 5

NN Classifier Accuracy: 100.00%

Execution no= 6

NN Classifier Accuracy: 100.00%

Execution no= 7

NN Classifier Accuracy: 100.00%

Execution no= 8

NN Classifier Accuracy: 100.00%

Execution no= 9

NN Classifier Accuracy: 100.00%

Execution no= 10

Only 50% of the results provides 100% accuracy in the Hard K-means whereas with the Soft K-means, 100% accuracy with all trials.!!

2)Make more silhouettes for our test

14/05/2014

**Problem: Silhouette imperfection**

**1) How to obtain better silhouettes in the existing dataset?**

* Active contours
* Other similar techniques to precisely get the curvatures

2) **Is there a possibility to get perfect silhouettes**(demo version) and carry out our Shape Context Algorithm in order to get an intuitive idea of the impact of the features such as neckedness, chestwidth,baldness etc in the REID process.

* Unity simulator and virtual reality
* Tried with Unity3d simulator and created AVATARS (mixamo packages) and did some tuning over the scale of body parts to create ~20 avatar models - .avi video, attention mode and the frames extracted from video.



1)normal 2) broad chest 3)long neck

* A trial run with the silhouette cropping and sampling.

21/05/2014

* Unity 3d simulator
* 6 AVATARS- 4 different features, different scales, 10 variations in biometrics.
* 4 image frames per person
* total 6\*10\*4=240 different avatars

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Person  Index | **Neckness** | **Chest size** | | | **Body size** | | | **Headlength** | | Comment on Biometrics | How to edit inUnity3d? |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Normal | Original |
| 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Necksize2 | Neck:1,2,1  Neck1:1,0.5,1 |
| 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Necksize3 | Neck:1,3,1  Neck1:1,0.33,1 |
| 4 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | Chest 2 | Leftarm:1,2,1  Rightarm:1,2,1 |
| 5 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | Chest 3 | Leftarm:1,3,1  Rightarm:1,3,1 |
| 6 | 1 | 1 | 1 | 1 | 0.25 | 1 | 1 | 1 | 1 | Bodysize1/4(thin) | Person:0.25,1,1 |
| 7 | 1 | 1 | 1 | 1 | 0.5 | 1 | 1 | 1 | 1 | Bodysize1/2(thin) | Person:0.5,1,1 |
| 8 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | Bodysize2 (fat) | Person:2,1,1 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1.25 | Long head | Head:1,1.25,1 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1.25 | 1 | Broad head | Head:1.25,1,1 |



* Now, the same VQ technique is applied to check the re-identification rate.

(PHD\codes\2014\ShapeContext\SC\SC\fuzzyKmeans\fuzme\_matlab) the code used: test\_fuzme\_ATHIRA\_Kmeans.m

Options to do normal K-Means or soft/fuzzy K-means.

results: (CMwithPby4withSoftKmeans) 

* Its seen that the best result is obtained with fuzzy rather than normal K-means
* Normalization has a great impact on the result: say, when R=Perimeter/4, the best result observed. This is important to have a scale invariant shape matching.

30/06/2014

* Planning to do:

1. Do without VQ stage- directly do the bipartite graph matching as per Malik paper.

* Can be a comparative analysis to see if the compression was really lossy or not!(w.r.to that one with Vquantization)

1. Adaptively select the interest points rather than taking into account of all the sample points (shoulders and other contour extremas..CODON?)
2. Model the relationship between SC and biometrics using REGRESSION.

Stages:

a) Collect the AVATAR dataset . Done! 6\*10\*4=240.

b) Compression via PCA

c) Regression

[SC].[A]=[BF]

**This week:**

1) Read on CODONS and other similar algorithms to select the interest points.

2) Directly go for Bipartite Graph Matching among the Avatar shape silhouettes.!

10/07/2014

1)Bipartite Graph Matching using Hungarian Algorithm (Code from Jacinto)

* Normalized the silhouette height so that the radius of the SC can be fixed

(In the SC paper (normalization by mean distance ofn2 point pairs))

* tested with 48 people- each one with 4 frames out of which 2 are trainset and the rest 2 are test set.
* Accuracy of 85% , with R=Perimeter/2.



* Why not 100%
* Open shape SC?
* slight changes in Cropping position ?
* Optimal Radius??(As per the paper, mean distance among the point pairs in the shape in order to get scale invariance)...
* Imperfect enclosing BB?
* Whats the problem with the 'extremely thin' guys??
* What next?
* Sanity test verified depicting better results with the shape context as a feature in matching the shape similarities of the people
* Codons/ Keypoints may help to reduce the computational cost but, alwaysthe best results with many points as the way now verified.
* Most important is to check the correlation of the biometric features simulated against the SC values. via REGRESSION?

Time to proceed??

17/07/2014

1) Double-checking of the ShapeContext also by using Malik's original code.

Borrowed the functions for SC\_feature\_extraction(normalization using mean\_dist) , hungarian algorithm and cost\_matching.

Process going on!!

18/07/2014- 24/07/2014

1. Basic 6 Avatars made to compare among themselves. 100?



1. Check with the same avatar, but try with different cropping positions? Does it has influence on the person re-id?



Yes… it does. So, the cropping position should be very well defined.

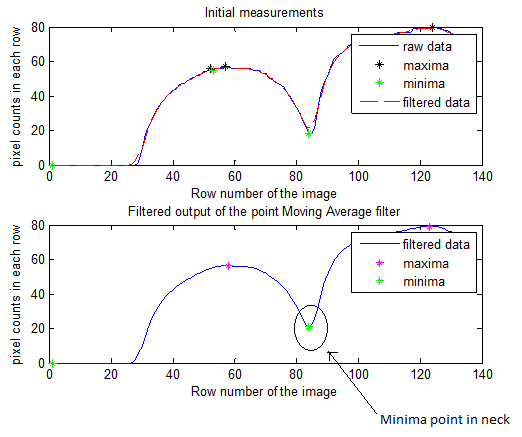


1. How to automate that??

* Have a predefined rule of cropping the person silhouette—for example, obtain the keypoint in the shoulder, and that top part…, or the equal length of head part towards the chest part from the shoulder line…etc.
* Also, avaoid the white space behind
* And, normalize the size of silhouette.
* **What I did??**
* **Instead of MANUAL cropping , do it AUTOMATICALLY, and NORMALIZE the height as well.**

1) Take the image (different size for different images.)

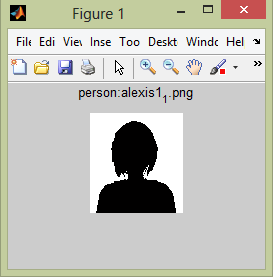
2) Obtain the pixel count info along the height of the image, in order to get the keypoint in the neck.



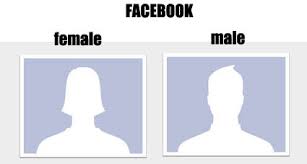
3) Now, add a predefined height towards the chest from the minima point,...(I took half the size of head.).

Also, applied a tolerance value in order not to miss the edge points towards all directions.

And, normalized the height of the person keeping the aspect ratio.



Found its the usual AVATAR format in social networking websites.

 (google) or 

**Result:** Could improve the results from **85% to 90%** :-)



(By avoiding the worst ones... With the Extremely LONG necks, met with a problem in finding the minimum and thus the cropping is not perfect.. So, avoided such samples!!)