Towards optimal ranking metrics

N. SEBE, M. LEW, D.P. HUIJSMANS

Department of Computer Science, Leiden University, Postbus 9512, 2300 RA, Leiden, The Netherlands
nicu,mlew,huijsman@wi.leidenuniv.nl

Abstract. Euclidean metric is frequently used in Computer Vision, mostly ad-hoc without any justification. However we have found that other metrics like double exponential metric or Cauchy one provide better results, in accordance with the maximum likelihood approach. In this paper we experiment with different modeling functions for similarity noise and compute the accuracy of different methods using these modeling functions in three kinds of applications: content-based image retrieval from a large database, stereo matching and video sequences. We provide a way to determine the modeling distribution which fits best the similarity noise distribution according to the ground truth. In the optimum case, when one has chosen the best modeling distribution, its corresponding metric will give the best ranking results for the ground truth provided.

Keywords: maximum likelihood, optimal metric, similarity noise distribution, content-based search, stereo matching, video sequence

1 Introduction

In general, image retrieval by content requires algorithms for extracting and comparing features. Extracted features from the imagery may be associated with entire digital images, or perhaps with specific regions of interest that are identified interactively, semi-automatically, or in a completely automatic manner. The QBIC effort ([6], [1]) is one project that has developed several methods for doing this. As an example, the texture of an image is represented by a feature vector that can be compared to texture feature vectors from other database images using Euclidean distance, thereby allowing the retrieval of images with ”similar” textures.

Another feature vector, color content, is typically described using a histogram. In [8], a histogram of the colors contained in each image is computed, and a L_1 metric is used to compare these color histograms. Also, in [7], efficient techniques for comparing histograms using quadratic measures of similarity have been proposed. A method for calculating the similarity between two digital images using a global signature which includes the texture, shape and color content is described in [4]. A normalized distance between probability density functions of feature vectors is used to match signatures. The authors present four possible distances that can be used to compare signatures without discussing how each of these distances influences the retrieval results.

In [3] the authors compare different retrieval methods, using feature vectors composed of projections, Local Binary Pattern and 2D-Trigrams, and evaluate image indexing and retrieval performance for similarity matches in a large database, as a function of the database size. Similar in [5] different methods for image copy location are compared and evaluated, taking into account their computational efficiency and accuracy with respect to real noise experiments.

In some of these applications it was assumed that the distribution of the similarity noise, defined like pixel-by-pixel difference between two similar images or their corresponding feature elements, is gaussian. In other applications, even when other metrics than L_2 were considered, there were no arguments why these were better.

A general application involving similarity matching might follow the scheme presented in figure 1. First one has to establish ground truth and extract the feature vectors to be used for comparison. The associated similarity noise distribution can be approximated with a modeling distribution and its associated metric (L_k). In the optimum case, when one has chosen the best modeling distribution, its corresponding metric will give the best ranking results for the ground truth provided.

1.1 Image Retrieval

In image retrieval the problem of finding the optimum modeling distribution for the similarity noise distribution is burdened by the fact that defining a ground truth and determining a testset is difficult. This difficulty is due to the subjectivity of the task: different users may have a completely different idea about the similarity of two images. To have a complete system for the evaluation of the performance of content-based similarity matching one would like:

- to have a clear definition of similarity
- a noise threshold to distinguish similar from dissimilar

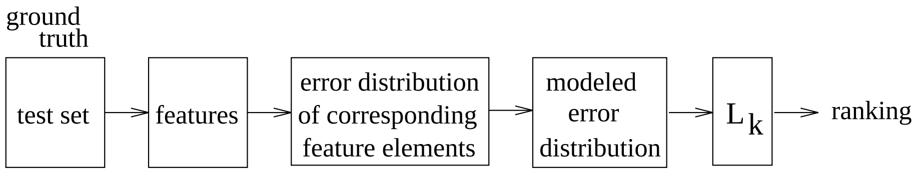

ground truth
test set
features
error distribution of corresponding feature elements
modeled error distribution
L_k
ranking

Figure 1: The steps in a similarity matching algorithm

- to have the right modeling function for similarity noise, resulting in a corresponding metric to base ranking on.

Choosing the right modeling function for the noise is a problem of robust estimation and consists of finding the maximum likelihood estimator for the defined ground truth.

1.2 Stereo Matching

This approach could be applied not only to the applications where the ground truth is defined according to subjectivity but, also in cases where a ground truth is already known. For instance, in stereo datasets the ground truth for matching corresponding points may be provided by the laboratory where these images were taken. This ground truth is a result of mapping the world coordinates, in which the camera is moving, to the image coordinates, using the 3D geometry relations of the scene. In this case one can test automatic stereo matchers that are able to detect the corresponding point pairs registered in stereo images of the testset scenes. For this stereo matcher one can determine the best metric when comparing different image regions to find the similar ones. The optimum metric in this case will give the most accurate stereo matcher.

1.3 Video sequences

The same approach can be also used for video sequences. Here we have a dynamic scene due to the camera movement or due to the movement of some objects in the scene. We have therefore, the neighboring images from the sequence being correlated and the closer we get to the current image in the sequence, the higher the correlation will be. The similarity matching in a video sequence can be used for example in detecting the movement of the objects in the image and also for obtaining occlusion maps. In this case the proper metric will also provide the most accurate results.

Section 2 describes the mathematical support for maximum likelihood approach. In Section 3 we apply these theoretical results to determine the influence of the similarity noise model on the accuracy of the retrieval methods in a large database. In Section 4 we study the similarity noise model to be chosen in a stereo matching application. The same approach is then applied on a video sequence (Section 5). Conclusions are given in Section 6.

2 Maximum likelihood estimator

From the mathematical-statistical point of view, the problem of finding the right model for the similarity noise comes down to the maximization of the similarity probability.

Consider first, two subsets of M images from the database (D) : X ⊂ D, Y ⊂ D which according to the ground truth are similar:

$$X \equiv Y \tag{1}$$

This can be written:

$$x_i \equiv y_i, \qquad i = 1, ..., M \tag{2}$$

where $x_i \in$ X, $y_i \in$ Y represent the images from the corresponding subsets.

The equation (2) can be further written as:

$$x_i = y_i + n_i, \qquad i = 1, ..., M \tag{3}$$

where n_i represent the “noise” image obtained as the difference between the other two images.

In this context the similarity probability can be defined:

$$P(X, Y) = \prod_{i=1}^{M} \{\exp[-\rho(x_i, y_i)]\} \tag{4}$$

where function ρ is the negative logarithm of the probability density of the noise.

According to (4) we have to find the probability density function of the noise that maximizes the similarity probability: *maximum likelihood* estimator for the noise distribution ([2]).

We can further suppose that this noise distribution is valid for all the database, so using it for all the images in the database one obtains the best possible ranking results.

In all above considerations, we were talking about images but this notion can be extended to feature vectors associated with the images when we are working with image features or, even, can be extended to pixel values in the images in the case of stereo matching.

Taking the logarithm of (4) we find that we have to minimize the expression:

$$\sum_{i=1}^{M} \rho(x_i, y_i) \tag{5}$$

In this case, according to (3), the function ρ depends not independently on its two arguments, query image x_i and the predicted one y_i, but only on their difference. We have thus a *local* estimator and we can replace (5) with:

$$\sum_{j=1}^{M} \rho(z) \tag{6}$$

where z $\equiv$ x_i-y_i and the operation "-" denotes pixel by pixel difference between the images, or an equivalent operation in feature space.

We define the derivative of $\rho(z)$ to be a function $\psi(z) \equiv \frac{d\rho(z)}{dz}$ which will occur as a weighting function when we find the minimum of the expression (5).

In case the noise is gaussian distributed:

$$Prob\{x_i - y_i\} \sim \exp([x_i - y_i]^2) \tag{7}$$

then

$$\rho(z) = z^2 \qquad \psi(z) = z \tag{8}$$

If the errors are distributed as a *double* or *two-sized exponential*, namely

$$Prob\{x_i - y_i\} \sim \exp(-|x_i - y_i|) \tag{9}$$

then, by contrast,

$$\rho(z) = |z| \qquad \psi(z) = sgn(z) \tag{10}$$

In this case the maximum likelihood estimator is obtained by minimizing the *mean absolute deviation*, rather than the *mean square deviation*. Here the tails of the distribution, although exponentially decreasing, are asymptotically much larger than any corresponding Gaussian.

A distribution with even more extensive - therefore sometimes even more realistic - tails is the *Cauchy* distribution,

$$Prob\{x_i - y_i\} \sim \frac{1}{a^2 + (x_i - y_i)^2} \tag{11}$$

where **a** is a parameter which determines the height and the tails of the distribution.

This implies

$$\rho(z) = \log(a^2 + z^2) \qquad \psi(z) = \frac{z}{a^2 + z^2} \tag{12}$$

3 Similarity noise in a large image database

One of the problems with query information retrieval systems is that the result of a query is simply a group of items that are hopefully interesting to the user (a group of images that are *similar* to the query image). Some additional information, such as similarity scores produced by the comparison process, might also be returned to allow a user to gauge the *correctness* of the result. It is, therefore, reasonable for a user to pose a question such as, *"Why do these images look similar ?"*. Using a probability density function approach one can give an objective answer to this question.

We applied the theoretical results described in Section 2 in order to determine the influence of similarity noise model on the similar image retrieval performance in the Leiden 19th Century Portrait Database (LCPD).

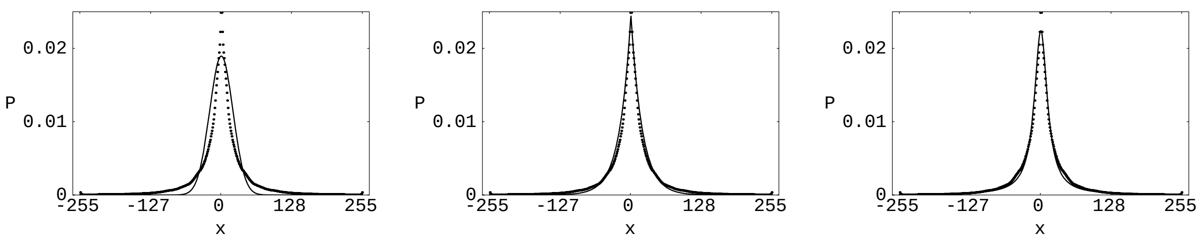
The LCPD is currently composed of 10165 images taken during the 19th century, and will be continually expanded until at least 50,000 images are in the database. Some images are copies of each other. However, due to different storage conditions, the copies have varying kinds and differing amounts of degradation. The degradation varies from intensity and moisture damage to scratches and writing on the images.

For our experiments we used 268 image pairs from the database as the ground truth for similarity. They represent near-copies due to the following reasons:

- all portraits are digitized on a scanner and sampled; the digitization process introduces sampling artifacts, noise and a number of geometric and photometric degrees of freedom;
- copies within an original set deteriorated differently over time: some have faded more then others, some are handled more so they became dirtier or some developed stains;
- some copies got written on, or have labels pasted on;
- some got cut in size to make them better fit for the a photo album

According to this ground truth, we determined the real distribution of the similarity noise considering three different spaces: intensity space, gradient and features space. For each of them we tried to match this real distribution best with one of the known distributions: normal, double exponential, cauchy. We then applied the corresponding similarity noise model to the entire database and inspected how it affected the retrieval results.

For comparing the retrieval results we used the performance measures given in [3]. We consider the **visible fraction** (F_v) that counts how many of the test-pairs **T** have counterparts that appear in the top $L = [\log_2 n]$



(a) gauss (b) exponential (c) cauchy (a=14.2057)

Figure 2: Similarity noise distribution in intensity space modeled by three theoretical distributions

ranks: the number of these visible test-pairs are called (T_v) and $n =$ database size. So

$$F_v = T_v/T \tag{13}$$

and is normalized to lie within [0,1].

A second performance measured is the **visible position** (P_v) that is defined like the ranking accuracy within the display window.

$$P_v = (L - R_v)/(L - 1) \tag{14}$$

where R_v is the average rank for visible test-pairs. P_v lies within [0,1]; 0 when $R_v = L$ (all test pairs just visible) and is 1 when $R_v = 1$ (all visible test-pairs on top).

Finally as a global measure we use the combined **retrieval quality** Q_r:

$$Q_r = (F_v + P_v)/2 \tag{15}$$

We will consider that a method provides better results when Q_r is bigger.

3.1 Intensity space

In figure 2 and table 1 we have the matching of the real similarity noise distribution in the intensity space considering the three distributions. The dotted lines represent the measured noise distribution. The approximation error for the measured noise distribution was calculated using the Euclidean distance between the two corresponding distributions.

gauss	exp	cauchy
0.033	0.013	0.012

Table 1: The approximation error for the measured noise distribution in intensity space using different modeled distributions

As one can notice the tails of the real distribution are prominent so the gaussian distribution cannot be a good match. Instead, the double exponential distribution and also, the Cauchy one are more suitable as approximations. These observations are in accordance with the theory described in the section 2.

One expects, therefore, to obtain better overall retrieval results using an L_c or L_1 metric than with a ranking based on L_2. As one can see in the first three columns of table 4 the results according to (13), (14) and (15) confirm this supposition.

But related to the theory, one could ask about the value of the parameter **a** in the Cauchy distribution that should be used to obtain the best results using this distribution.

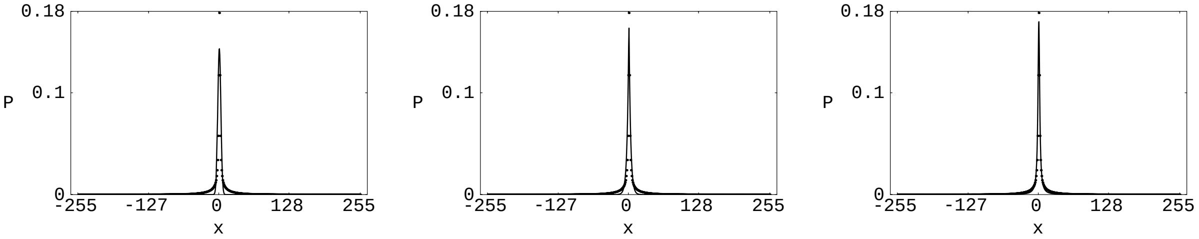
We consider the following steps to solve this problem:

- perform the matching between the measured noise distribution and the one modeled by a Cauchy distribution and determine the value of the parameter **a** that minimizes the difference between these distributions.
- use this value of parameter **a** in the corresponding metric, L_c.

We were also interested about how the value of this parameter will influence the retrieval quality when we use the Cauchy distribution as a model. This is illustrated in figure 5(a). One can observe that for a wide scale of values for parameter **a** the results using L_c are better that the one using L_2. One can also notice that around the optimum value for the parameter **a** (a $\approx$ 14) the results are the best; even better that the ones obtained using L_1.

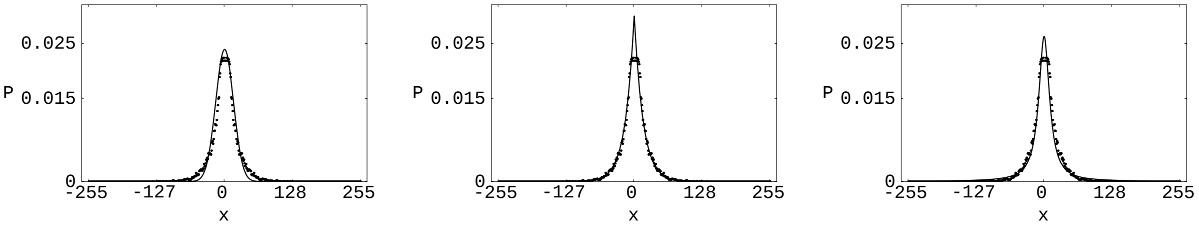
3.2 Gradient space

The second space where we examine the influence of the similarity noise model was the gradient space considering



(a) gauss (b) exponential (c) cauchy (a=1.4327)

Figure 3: Similarity noise distribution in gradient space modeled by three theoretical distributions



(a) gauss (b) exponential (c) cauchy (a=12.1248)

Figure 4: Similarity noise distribution in feature space modeled by three theoretical distributions

the gradient images obtained using a Sobel gradient. We follow the same steps as before.

gauss	exp	cauchy
0.1203	0.0698	0.0459

Table 2: The approximation error for similarity noise distribution in gradient space

The measured noise distribution (figure 3) in this case is steeper. So, consequently, the error using the gaussian approximation is even bigger. The retrieval quality measures reflect this (table 4 and figure 5(b)). From the table 2 we deduce that the results obtained with L_c can be better that the one obtained with L_1, fact illustrated in figure 5(b) around the optimum value for the parameter.

3.3 Projection space

In the feature space we consider the projection features from the images. We used average row- and column values (line integrals) as a feature vector. The procedure in this case will use the difference distribution between the corresponding feature elements instead of pixel by pixel difference distribution. In figure 4 one can see that the gaussian distribution in this case more closely matches the measured noise distribution so, the results obtained with L_2 will be close to the ones obtained with the other metrics. Also one can notice that in this case the double exponential distribution matches the measured noise distribution better than the Cauchy one, illustrated consequently in figure 5(c).

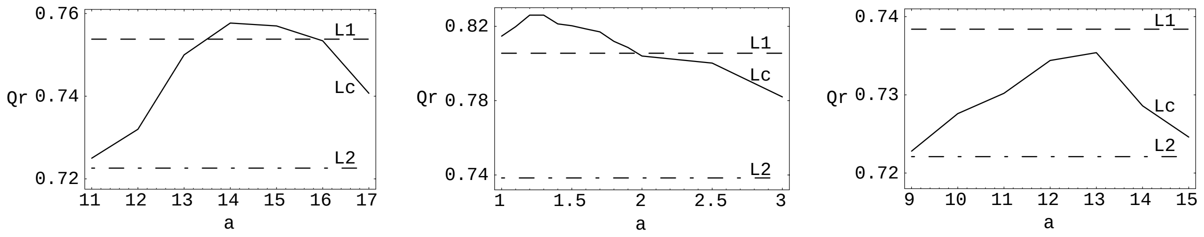
gauss	exp	cauchy
0.0306	0.0228	0.0273

Table 3: The approximation error for the measured noise distribution in feature space

4 Similarity noise in a stereo matching application

Stereo matching is the process of determining correspondences between entities in related images. One can test automatic stereo matchers that are able to detect these correspondences. The choice of the optimum metric when compare different image regions in order to find the similar ones will give the most accurate stereo matcher.

We used two standard stereo data sets (Castle set and Tower set) provided by Carnegie Mellon University. These datasets contain multiple images of static scenes with accurate information about object locations in 3D. The images were taken with a scientific camera in an in-



(a) PbyP-intensity (b) PbyP-gradient (c) Projections

Figure 5: Retrieval quality

Methods	PbyP-int			PbyP-grad			Proj-int		
	L_2	L_1	L_c	L_2	L_1	L_c	L_2	L_1	L_c
F_v	0.614	0.635	0.650	0.650	0.708	0.750	0.614	0.625	0.620
P_v	0.830	0.872	0.865	0.826	0.903	0.902	0.830	0.851	0.850
Q_r	0.722	0.753	0.757	0.738	0.805	0.826	0.722	0.738	0.735

Table 4: Similar image retrieval performance

door setting, the Calibrated Imaging Laboratory at CMU. The 3D locations are given in X-Y-Z coordinates with a simple text description (at best accurate to 0.3 mm), and the corresponding image coordinates (the ground truth) are provided for all eleven images taken for each scene. For each image there are provided 28 points as ground truth in Castle set and 9 points in Tower set.

In this case we already know the ground truth so, according to figure 1 we can skip the first two blocks. Even though in this case the ground truth does not consist of similar images in a database, we can apply directly the theory described in section 2. In this case the role of the database will be played by the total amount of pixels in all eleven images and the corresponding pixels in these images will constitute the ground truth. In each of the images we consider the points which are given by the ground truth and we want to find the proper similarity noise which will assure the best accuracy in finding the corresponding points according to the ground truth.

We cannot use single pixel information but have to use a region around it. So we will perform template matching. Our automatic stereo matcher will match a template defined around one point from an image with the templates around points in the other images in order to find similar ones. If the resulting points are equivalent to those provided by the ground truth we consider that we have a *hit*, otherwise we have a *miss*. The accuracy is given by the number of the *hits* divided by the number of possible *hits* (number of corresponding point pairs).

Because the ground truth is provided with subpixel accuracy we consider therefore that we have a *hit* when the corresponding point found lies in the neighborhood of one pixel around the point provided by the ground truth.

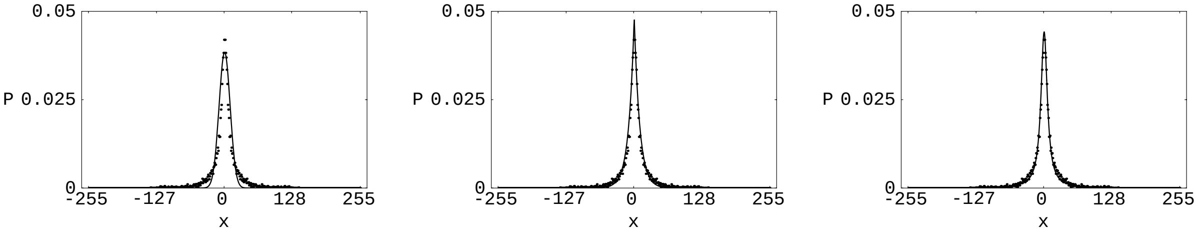
The first step is to compute the similarity noise distribution according to the ground truth. In this case the similarity noise is defined as the difference between two corresponding pixels. The following step is to match this distribution with one of the three distribution and to find out which fits the best.

Image set	gauss	exp	cauchy
Castle stereo_set	0.0486	0.0286	0.0246
Tower stereo_set	0.0649	0.0475	0.0387

Table 5: The approximation error for the corresponding point noise distribution in stereo matching for three distribution models

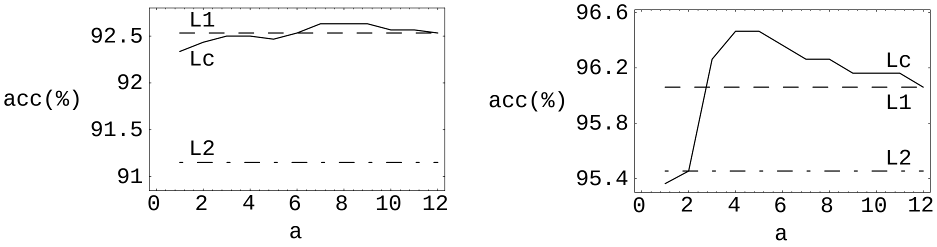
We present the corresponding point matching distribution in figure 6. As one can see from the table 5 the Cauchy distribution fits the the real distribution best. So one expects the accuracy to be biggest when using L_c (table 6). The best values for the accuracy using L_c are obtained around the optimum value computed for the matching step (figure 7).

As one can see from the table 5 the double exponential and Cauchy distribution results are almost identical using the first dataset and deviate a bit more using the second test set; this is illustrated in the figure 7.



(a) gauss (b) exponential (c) cauchy (a=7.2007)

Figure 6: Noise distribution in the stereo matcher using Castle dataset



(a) Castle dataset (b) Tower dataset

Figure 7: The accuracy of the stereo matcher (%)

Image set	gauss	exp	cauchy
Castle stereo_set	91.15	92.53	92.63
Tower stereo_set	95.45	96.06	96.46

Table 6: The accuracy of the stereo matcher (%)

5 Similarity noise in a video sequence

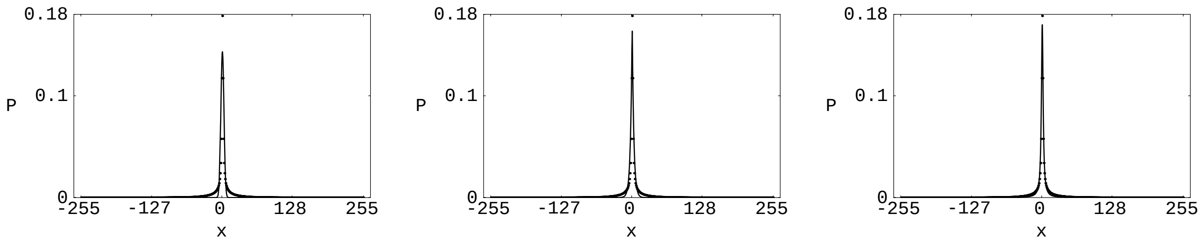
We used a video sequence containing 19 images on a talking head in a static background. For each image in this video sequence there are provided 14 points as ground truth.

Unlike in the stereo data sets where we could consider that all the images from a set are similar, here, we have to take into account that only neighboring images will be similar. This means that taking a current image in the sequence the further we consider an image in the sequence, the more different it is. In this application we consider only the first order of neighboring (nearest neighbors): we consider to be similar images only the images which are sequential into the sequence. This means that for the current image we consider only the previous and the following one into the sequence.

In this application we also perform a template matching. So we consider a region around one point provided by the ground truth and try to find automatically the correspondent region in the neighboring images from the sequence. If the central pixel of the region found lies in the neighborhood of one pixel beside the corresponding pixel provided by the ground truth than we have a *hit*, otherwise we have a *miss*. As in the previous application, the accuracy is computed as the number of the *hits* divided by the number of possible *hits*.

We consider the same steps as in the previous application. First step is to compute the similarity noise distribution considering only the corresponding pixel values into similar regions provided by the ground truth. The following step is then, to find the best fit for this real distribution taking into account one of the three known distribution: normal, double exponential, Cauchy. Having now the right model for the similarity noise we can use consequently the corresponding metric in the similarity matching step. The usage of this metric will assure the best accuracy in matching.

In figure 8 we have the matching of the real similarity noise considering the three distributions. As one can



(a) gauss (b) exponential (c) cauchy (a=2.88)

Figure 8: Similarity noise distribution in the video sequence modeled by three theoretical distributions (the approximation error is: (a) 0.0379 ; (b) 0.0376 ; (c) 0.0327)

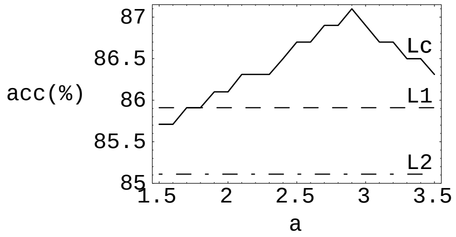


Figure 9: The accuracy of the matching process in video sequence (%)

notice the best match is the Cauchy distribution and the double exponential distribution is a better match than the normal one. So one expects that the accuracy will be bigger when using L_c then when using L_1 and L_2. This is illustrated in figure 9. One can also notice that the biggest value for the accuracy when using L_c is obtained around the value of the parameter **a** (a=2.88) which assure the best match between Cauchy distribution and real distribution.

6 Conclusion

In this paper we showed how to derive a suitable metric given a visual query problem when ground truth is available. The final similarity ranking can be improved this way. From our experience L_1 clearly outperform L_2. Using a parametrized metric like L_c one can often further improve the result.

We showed how maximum likelihood estimation can be successfully applied in content-based query applications.

It was also shown that the behavior of the performance measures given in [3] are in accordance with the maximum likelihood approach.

References

[1] C. Faloutsos, M. Flicker, W. Niblack, R. Barder, W. Equitz, and D. Petrovic. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, pages 231–262, 1994.

[2] P.J Huber. *Robust Statistic*. NewYork: Wiley, 1981.

[3] D.P. Huijsmans, M.S. Lew, and D. Denteneer. Quality measures for interactive image retrieval with a performance evaluation of two 3x3 texel-based methods. In *Lectures Notes in Computer Science*, volume 1311(2), pages 22–29. Springer-Verlag, 1997.

[4] P.M. Kelly, T.M. Cannon, and D.R. Hush. Query by image example: the CANDID approach. *SPIE - Storage and Retrieval for Image and Video Databases*, 2420(3):238–248, 1995.

[5] M.S. Lew, D.P. Huijsmans, and D. Denteneer. Optimal keys for image database indexing. In *Lectures Notes in Computer Science*, volume 1311(2), pages 148–155. Springer-Verlag, 1997.

[6] W. Niblack, R. Barder, W. Equitz, M. Flicker, E. Glasman, D. Petrovic, P. Yanker, C. Faloutsos, and G. Yaublin. The QBIC project: Querying images by content using color, texture and shape. *SPIE - Storage and Retrieval for Image and Video Databases*, 1908:173–181, 1993.

[7] H.S. Sawhney and J.L. Hafner. Efficient color histogram indexing. In *Proc. of 1994 IEEE International Conference on Image Processing*, volume 2, pages 66–70, 1994.

[8] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.