Robust Shape Matching

N. Sebe and M. Lew

Leiden Institute of Advanced Computer Science,
Leiden University, NielsBohrweg 1, 2333CA Leiden,
The Netherlands,
{nicu, mlew}@liacs.nl

Abstract. Many visual matching algorithms can be described in terms of the features and the inter-feature distance or metric. The most commonly used metric is the sum of squared differences (SSD), which is valid from a maximum likelihood perspective when the real noise distribution is Gaussian. However, we have found experimentally that the Gaussian noise distribution assumption is often invalid. This implies that other metrics, which have distributions closer to the real noise distribution, should be used. In this paper we considered a shape matching application. We implemented two algorithms from the research literature and for each algorithm we compared the efficacy of the SSD metric, the SAD (sum of the absolute differences) metric, and the Cauchy metric. Furthermore, in the case where sufficient training data is available, we discussed and experimentally tested a metric based directly on the real noise distribution, which we denoted the maximum likelihood metric.

1 Introduction

We are interested in using shape descriptors in content-based retrieval. Assume that we have a large number of images in the database. Given a query image, we would like to obtain a list of images from the database which are most similar (here we consider the shape aspect) to the query image. For solving this problem, we need two things - first, a measure which represents the shape information of the image and second a similarity measure to compute the similarity between corresponding features of two images.

The similarity measure is a matching function and gives the degree of similarity for a given pair of images (represented by shape measures). The desirable property of a similarity measure is that it should be a metric (that is, it has the properties of symmetry, transitivity, and linearity). The SSD (L_2) and the SAD (L_1) are the most commonly used metrics. This brings to mind several questions. First, under what conditions should one use the SSD versus the SAD? From a maximum likelihood perspective, it is well known that the SSD is justified when the additive noise distribution is Gaussian. The SAD is justified when the additive noise distribution is Exponential (double or two-sided exponential). Therefore, one can determine which metric to use by checking if the real noise distribution is closer to the Gaussian or the Exponential. The common assumption is that the real noise distribution should fit either the Gaussian or the Exponential, but what if there is another distribution which fits the real noise distribution better? Toward answering this question, we have endeavored to use international test sets and promising algorithms from the research literature.

In this paper, the problem of image retrieval using shape was approached by active contours for segmentation and invariant moments for shape measure. Active contours were first introduced by Kass et al. [1] and were termed snakes by the nature of their movement. Active contours are a sophisticated approach to contour extraction and image interpretation. They are based on the idea of minimizing energy of a continuous spline contour subject to constraints on both its autonomous shape and external forces derived from a superposed image that pull the active contour toward image features such as lines and edges.

Moments describe shape in terms of its area, position, orientation, and other parameters. The set of invariant moments [2] makes a useful feature vector for the recognition of objects which must be detected regardless of position, size, or orientation. Matching of the invariant moments feature vectors is computationally inexpensive and is a promising candidate for interactive applications.

2 Active Contours and Invariant Moments

Active contours challenge the widely held view of bottom-up vision processes. The principal disadvantage with the bottom-up approach is its serial nature; errors generated at a low-level are passed on through the system without the possibility of correction. The principal advantage of active contours is that the image data, the initial estimate, the desired contour properties, and the knowledge-based constraints are integrated into a single extraction process.

In the literature, del Bimbo et al. [3] deformed active contours over a shape in an image and measured the similarity between the two based on the degree of overlap and on how much energy the active contour has to spend in the deformation. Jain et al. [4] used a matching scheme with deformable templates. Our work is different in that we use a Gradient Vector Flow (GVF) based method [5] to improve the automatic fit of the snakes to the object contours.

Active contours are defined as energy-minimizing splines under the influence of internal and external forces. The internal forces of the active contour serve as a smoothness constraint designed to hold the active contour together (elasticity forces) and to keep it from bending too much (bending forces). The external forces guide the active contour towards image features such as high intensity gradients. The optimal contour position is computed such that the total energy is minimized. The contour can hence be viewed as a reasonable balance between geometrical smoothness properties and local correspondence with the intensity function of the reference image.

Let the active contour be given by a parametric representation $v(s) = (x(s), y(s))$, with s the normalized arc length of the contour. The expression for the total energy can then be decomposed as follows:

$$E_{total} = \int_0^1 [E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))]\, ds \tag{1}$$

where E_{int} represents the internal forces (or energy) which encourage smooth curves, E_{image} represents the local correspondence with the image function, and E_{con} represents a constraint force that can be included to attract the contour to specific points in the image plane. In the following discussions E_{con} will be ignored. E_{image} is typically defined such that locations with high image gradients or short distances to image gradients are assigned low energy values.

2.1 Internal Energy

E_{int} is the internal energy term which controls the natural behavior of the active contour. It is designed to minimize the curvature of the active contour and to make the active contour behave in an elastic manner. According to Kass et al. [1] the internal energy is defined as

$$E_{int}(v(s)) = \alpha(s) \left| \frac{dv(s)}{ds} \right|^2 + \beta(s) \left| \frac{d^2 v(s)}{ds^2} \right|^2 \tag{2}$$

The first order continuity term, weighted by $\alpha(s)$, makes the contour behave elastically, while the second order curvature term, weighted by $\beta(s)$, makes it resistant to bending. Setting $\beta(s) = 0$ at a point s allows the active contour to become second order discontinuous at that point and to develop a corner. Setting $\alpha(s) = 0$ at a point s allows the active contour to become discontinuous. Active contours can interpolate gaps in edges phenomena known as subjective contours due to the use of the internal energy. It should be noted that $\alpha(s)$ and $\beta(s)$ are defined to be functions of the curve parameter s, and hence segments of the active contour may have different natural behavior. Minimizing the energy of the derivatives gives a smooth function.

2.2 Image Energy

E_{image} is the image energy term derived from the image data over which the active contour lies and is constructed to attract the active contour to desired feature points in the image, such as edges and lines. The edge based functional attracts the active contour to contours with large image gradients - that is, to locations of strong edges.

$$E_{edge} = - \left| \nabla I(x, y) \right| \tag{3}$$

2.3 Problems with Active Contours

There are a number of fundamental problems with the active contours and solutions to these problems sometimes create problems in other components of the active contour model.

Initialization. The final extracted contour is highly dependent on the position and shape of the initial contour due to the presence of many local minima in the energy function. The initial contour must be placed near the required feature otherwise the contour can become obstructed by unwanted features like JPEG compression artifacts, closeness of a nearby object, etc.

Non-convex shapes. How do we extract non-convex shapes without compensating the importance of the internal forces or without a corruption of the image data? For example, pressure forces [6] (addition to the external force) can push an active contour into boundary concavities, but cannot be too strong or otherwise weak edges will be ignored. Pressure forces must also be initialized to push out or push in, a condition that mandates careful initialization.

The original method of Kass et al. [1] suffered from three main problems: dependence on the initial contour, numerical instability, and lack of guaranteed convergence to the global energy minimum. Amini et al. [7] improved the numerical instability by minimizing the energy functional using dynamic programming, which allows inclusion of hard constraints into the energy functional. However, memory requirements are large, being $O(nm^2)$, and the method is slow, being $O(nm^3)$ where n is the number of contour points and m is the neighborhood size to which a contour point is allowed to move in a single iteration. Seeing the difficulties with both previous methods, Williams and Shah [8] developed the *greedy algorithm* which combines speed, flexibility, and simplicity. The greedy algorithm is faster $O(nm)$ than the dynamic programming and is more stable and flexible for including constraints than the variational approach of Kass et al. [1]. During each iteration, a neighborhood of each point is examined and a point in the neighborhood with the smallest energy value provides the new location of the point. Iterations continue till the number of points in the active contour that moved to a new location in one iteration is below a specified threshold.

2.4 Gradient Vector Flow

Since the greedy algorithm easily accommodates new changes, there are three things we would like to add to it: (1) the ability to inflate the contour as well as to deflate it, (2) the ability to deform to concavities, and (3) the ability to increase the capture range of the external forces. These three additions reduce the sensitivity to initialization of the active contour and allow deformation inside concavities. This can be done by replacing the existing external force (image term) with the gradient vector flow (GVF) [5]. The GVF is an external force computed as a diffusion of the gradient vectors of an image, without blurring the edges.

Xu and Prince [5] define the gradient vector flow (GVF) field to be the vector field $\mathbf{v}(i,j) = (u(i,j), v(i,j))$ which is updated with every iteration of the diffusion equations:

$$u_{i,j}^{n+1} = (1 - b_{i,j})u_{i,j}^{n} + (u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}) + c_{i,j}^{1} \quad (4)$$

$$v_{i,j}^{n+1} = (1 - b_{i,j})v_{i,j}^{n} + (v_{i+1,j}^{n} + v_{i,j+1}^{n} + v_{i-1,j}^{n} + v_{i,j-1}^{n} - 4v_{i,j}^{n}) + c_{i,j}^{2} \quad (5)$$

where $b_{i,j} = G_i(i,j)^2 + G_j(i,j)^2$, $c_{i,j}^1 = b_{i,j}G_i(i,j)$, and $c_{i,j}^2 = b_{i,j}G_j(i,j)$ with G_i and G_j the first and the second elements of the gradient vector.

The second term in (4) and (5) is the Laplacian operator. The intuition behind the diffusion equations is that in homogeneous regions, the first and third terms are 0 since the gradient is 0, and within those regions, u and v are each determined by Laplace equation. This results in a type of "filling-in" of information taken from the boundaries of the region. In regions of high gradient v is kept nearly equal to the gradient.

Creating GVF field yields streamlines to a strong edge. In the presence of these streamlines, blobs and thin lines in the way to strong edges do not form any impediments to the movement of the active contour. It can be considered as an advantage if the blobs are in front of the shape, nevertheless it can be considered as a disadvantage if the active contour enters the silhouette of the shape.

2.5 Invariant Moments

Perhaps the most popular method for shape description is the use of invariant moments [2] which are invariant to affine transformations. In the case of a digital image, the moments are approximated by

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad (6)$$

where the order of the moment is $(p+q)$, x and y are the pixel coordinates relative to some arbitrary standard origin, and $f(x,y)$ represents the pixel brightness.

To have moments that are invariant to translation, scale, and rotation, first the central moments μ are calculated

$$\mu_{pq} = \sum_x \sum_y (x - \overline{x})^p (y - \overline{y})^q f(x,y), \ \overline{x} = \frac{m_{10}}{m_{00}}, \ \overline{y} = \frac{m_{01}}{m_{00}} \quad (7)$$

Further, the normalized central moments η are calculated

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\lambda}},\ \lambda = \frac{(p+q)}{2},\ p+q \geq 2 \tag{8}$$

From these normalized parameters a set of invariant moments $\{\phi\}$ found by Hu [2], can be calculated. The 7 equations of the invariant moments contain terms up to order 3:

$$\begin{aligned}
\phi_1 =& \eta_{20} + \eta_{02} \\
\phi_2 =& (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 =& (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 =& (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \\
\phi_5 =& (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right) + \\
& (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right) \\
\phi_6 =& (\eta_{20} - \eta_{02})\left((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 =& (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})\left((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right) + \\
& (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})\left(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right)
\end{aligned} \tag{9}$$

Global (region) properties provide a firm common base for similarity measure between shapes silhouettes where gross structural features can be characterized by these moments. Since we do not deal with occlusion, the invariance to position, size, and orientation, and the low dimensionality of the feature vector represent good reasons for using the invariant moments in matching shapes. The logarithm of the invariant moments is taken to reduce the dynamic range.

3 Maximum Likelihood Approach

In the previous sections we were discussing about extracting the shape information in a feature vector. In order to implement a content-based retrieval application we still need to provide a framework for selecting the similarity measure to be used when the feature vectors are compared.

In our previous work [9], we showed that the maximum likelihood theory allows us to relate a noise distribution to a metric. Specifically, if we are given the noise distribution then the metric which maximizes the similarity probability is

$$\sum_{i=1}^{M} \rho(n_i) \tag{10}$$

where n_i represents the i^{th} bin of the discretized noise distribution and ρ is the maximum likelihood estimate of the negative logarithm of the probability density of the noise. Typically, the noise distribution is represented by the difference between the corresponding elements given by the ground truth.

To analyze the behavior of the estimate we take the approach described in [10] and based on the influence function. The influence function characterizes the bias that a particular measurement has on the solution and is proportional to the derivative, ψ, of the estimate

$$\psi(z) \equiv \frac{d\rho(z)}{dz} \tag{11}$$

In the case where the noise is Gaussian distributed:

$$P(n_i) \sim \exp(-{n_i}^2) \tag{12}$$

then,

$$\rho(z) = z^2 \qquad \text{and} \qquad \psi(z) = z \tag{13}$$

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

$$P(n_i) \sim \exp(-|n_i|) \tag{14}$$

then,

$$\rho(z) = |z| \qquad \text{and} \qquad \psi(z) = \operatorname{sgn}(z) \tag{15}$$

In this case, using (10), we minimize 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 tails is the Cauchy distribution,

$$P(n_i) \sim \frac{a}{a^2 + {n_i}^2} \tag{16}$$

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

This implies

$$\rho(z) = \log\left(1 + \left(\frac{z}{a}\right)^2\right) \qquad \text{and} \qquad \psi(z) = \frac{z}{a^2 + z^2} \tag{17}$$

For normally distributed errors, (13) says that the more deviant the points, the greater the weight. By contrast, when tails are somewhat more prominent, as in (14), then (15) says that all deviant points get the same relative weight, with only the sign information used. Finally, when the tails are even larger, (17) says that ψ increases with deviation, then starts decreasing, so that very deviant points - the true outliers - are not counted at all.

Maximum likelihood gives a direct connection between the noise distributions and the comparison metrics. Considering ρ as the negative logarithm of the probability density of the noise, then the corresponding metric is given by Eq. (10).

Consider the Minkowski-form distance L_p between two vectors x and y:

$$L_p(x, y) = \left(\sum_i |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{18}$$

If the noise is Gaussian distributed, so $\rho(z) = z^2$, then (10) is equivalent to (18) with $p = 2$. Therefore, in this case the corresponding metric is L_2. Equivalently, if the noise is Exponential, so $\rho(z) = |z|$, then the corresponding metric is L_1 (Eq. (18) with $p = 1$). In the case the noise is distributed as a Cauchy distribution with scale parameter a, then the corresponding metric is no longer a Minkovski metric. However, for convenience we denote it as L_c:

$$L_c(x, y) = \sum_i \log\left(1 + \left(\frac{x_i - y_i}{a}\right)^2\right) \tag{19}$$

In practice, the probability density of the noise can be approximated as the normalized histogram of the differences between the corresponding feature vectors elements. For convenience, the histogram is made symmetric around zero by considering pairs of differences (e.g., $x-y$ and $y-x$). Using this normalized histogram, we extract a metric, called *maximum likelihood* (ML) *metric*. The ML metric is given by Eq. (10) where $\rho(n_i)$ is the negative logarithm of $P(n_i)$:

$$\rho(n_i) = -\log(P(n_i)). \tag{20}$$

The ML metric is a discrete metric extracted from a discrete normalized histogram having a finite number of bins. When n_i does not exactly match any of the bins, for calculating $P(n_i)$ we perform linear interpolation between $P(n_{inf})$ (the histogram value at bin n_{inf}) and $P(n_{sup})$ (the histogram value at bin n_{sup}), where n_{inf} and n_{sup} are the closest inferior and closest superior bins to n_i, respectively:

$$P(n_i) = \frac{(n_{sup} - n_i)P(n_{inf}) + (n_i - n_{inf})P(n_{sup})}{n_{sup} - n_{inf}} \tag{21}$$

4 Experiments

We assume that representative ground truth is provided. The ground truth is split into two non-overlapping sets: the training set and the test set. First, for each image in the training set a feature vector is extracted. Second, the real noise distribution is computed as the normalized histogram of differences from the corresponding elements in feature vectors taken from similar images according to the ground truth. The Gaussian, Exponential, and Cauchy distributions are fitted to the real distribution. The Chi-square test is used to find the fit between each of the model distributions and the real distribution. We select the model distribution which has the best fit and its corresponding metric L_k is used in ranking. The ranking is done using only the test set.

It is important to note that for real applications, the parameter in the Cauchy distribution is found when fitting this distribution to the real distribution from the training set. This parameter setting would be used for the test set and any future comparisons in that application.

As noted in the previous section, it is also possible to create a metric based on the real noise distribution using maximum likelihood theory. Consequently, we denote the maximum likelihood (ML) metric as (10) where ρ is the negative logarithm of the normalized histogram of the absolute differences from the training set. Note that the histogram of the absolute differences is normalized to have area equal to one by dividing the histogram by the total number of examples in the training set. This normalized histogram is our approximation for the probability density function.

For the performance evaluation let $\mathcal{Q}_1, \cdots, \mathcal{Q}_n$ be the query images and for the i-th query $\mathcal{Q}_i$, $\mathcal{I}_1^{(i)}, \cdots, \mathcal{I}_m^{(i)}$ be the images similar with $\mathcal{Q}_i$ according to the ground truth. The retrieval method will return this set of answers with various ranks. As an evaluation measure of the performance of the retrieval method we used recall vs. precision at different scopes: For a query $\mathcal{Q}_i$ and a scope $s > 0$, the recall r is defined as $|\{\mathcal{I}_j^{(i)} | rank(\mathcal{I}_j^{(i)}) \leq s\}|/m$, and the precision p is defined as $|\{\mathcal{I}_j^{(i)} | rank(\mathcal{I}_j^{(i)}) \leq s\}|/s$.

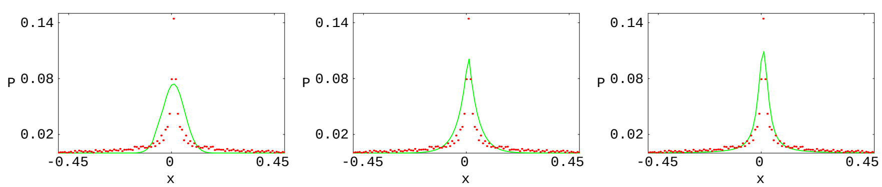
In our experiments we used a database of 1,440 images of 20 common house hold objects from the COIL-20 database [11]. Each object was placed on a turntable and photographed every 5° for a total of 72 views per object. Examples are shown in Fig. 1.

Fig. 1. Example of images of one object rotated with 60°

In creating the ground truth we had to take into account the fact that the images of one object may look very different when an important rotation is considered. Therefore, for a particular instance (image) of an object we consider as similar the images taken for the same object when it was rotated within $\pm r \times 5°$. In this context, we consider two images to be r-similar if the rotation angle of the object depicted in the images is smaller than $r \times 5°$. In our experiments we used $r = 3$ so that one particular image is considered to be similar with 6 other images of the same object rotated within $\pm 15°$. We prepared our training set by selecting 18 equally spaced views for each object and using the remaining views for testing.

The first question we asked was, "Which distribution is a good approximation for the similarity noise distribution?" To answer this we needed to measure the similarity noise caused by the object rotation and depending on the feature extraction algorithm (greedy or GVF). The real noise distribution was obtained as the normalized histogram of differences between the elements of feature vectors corresponding to similar images from the training set.

Fig. 2 presents the real noise distribution obtained for the greedy algorithm. The best fit Exponential had a better fit to the noise distribution than the Gaussian. Consequently, this implies that L_1 should provide better retrieval results than L_2. The Cauchy distribution is the best fit overall, and the results obtained with L_c should reflect this. However, when the maximum likelihood metric (ML) extracted directly from the similarity noise distribution is used we expect to obtain the best retrieval results.



(a) Gauss (b) Exponential (c) Cauchy (**a**=2.43)

Fig. 2. Similarity noise distribution for the greedy algorithm compared with (a) the best fit Gaussian (approximation error is 0.156), (b) the best fit Exponential (approximation error is 0.102), and (c) the best fit Cauchy (approximation error is 0.073)

In the case of GVF algorithm the approximation errors for matching the similarity noise distribution with a model distribution are given in Table 1. Note that the Gaussian is the worst approximation. Moreover, the difference between the Gaussian fit and the fit obtained with the other two distributions is larger than in the previous case and therefore the results obtained with L_2 will be much worse. Again the best fit by far is provided by the Cauchy distribution.

The results are presented in Fig. 3 and Table 2. In the precision-recall graphs the curves corresponding to L_c are above the curves corresponding to L_1 and L_2 showing that the method using L_c is more effective. Note that the choice of the noise model significantly affects the retrieval results. The Cauchy distribution was

Gauss	Exponential	Cauchy
0.0486	0.0286	0.0146

Table 1. The approximation error for matching the similarity noise distribution with one of the model distributions in the case of GVF algorithm (for Cauchy **a**=3.27)

the best match for the measured similarity noise distribution and the results in Table 2 show that the Cauchy model is more appropriate for the similarity noise than the Gaussian and Exponential models. However, the best results are obtained when the metric extracted directly from the noise distribution is used. One can also note that the results obtained with the GVF method are significantly better than the ones obtained with the greedy method.

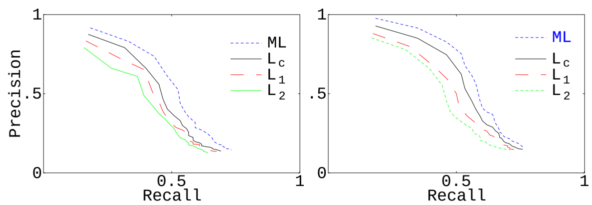


Fig. 3. Precision/Recall for COIL-20 database using the greedy algorithm (for L_c **a**=2.43) (left) and the GVF algorithm (for L_c **a**=3.27) (right)

		Precision			Recall		
Scope		6	10	25	5	10	25
Greedy	L_2	0.425	0.258	0.128	0.425	0.517	0.642
	L_1	0.45	0.271	0.135	0.45	0.542	0.675
	L_c **a**=2.43	0.466	0.279	0.138	0.466	0.558	0.692
	ML	0.525	0.296	0.146	0.525	0.592	0.733
GVF	L_2	0.46	0.280	0.143	0.46	0.561	0.707
	L_1	0.5	0.291	0.145	0.5	0.576	0.725
	L_c **a**=3.27	0.533	0.304	0.149	0.533	0.618	0.758
	ML	0.566	0.324	0.167	0.566	0.635	0.777

Table 2. Precision and Recall for different Scope values

In summary, L_c performed better than the analytic distance measures, and the ML metric performed best overall.

5 Conclusions

In this paper we showed that the GVF based snakes give better retrieval results than the traditional snakes. In particular, the GVF snakes have the advantage in that it is not necessary to know apriori whether the snake must be expanded or contracted to fit the object contour. Furthermore, the GVF snakes have the ability to fit into concavities of the object which traditional snakes cannot do. Both of these factors resulted in significant improvement in the retrieval results.

We also addressed the problem of finding the appropriate metric to use for computer vision applications in shape based retrieval. From our experiments, L_2 is typically not justified because the similarity noise distribution is not Gaussian.

We showed that better accuracy was obtained when the Cauchy metric was substituted for the L_2 and L_1. Minimizing the Cauchy metric is optimal with respect to maximizing the likelihood of the difference between image elements when the real noise distribution is equivalent to a Cauchy distribution. Therefore, the breaking points occur when there is no ground truth, the ground truth is not representative, or when the real noise distribution is not a Cauchy distribution. We also make the assumption that one can measure the fit between the real distribution and a model distribution, and that the model distribution which has the best fit should be selected. We used the Chi-square test as the measure of fit between the distributions, and found in our experiments that it served as a reliable indicator for distribution selection.

In conclusion, we showed that the prevalent Gaussian distribution assumption is often invalid, and we proposed the Cauchy metric as an alternative to both L_1 and L_2. In the case where representative ground truth can be obtained for an application, we provided a method for selecting the appropriate metric. Furthermore, we explained how to create a maximum likelihood metric based on the real noise distribution, and in our experiments we found that it consistently outperformed all of the analytic metrics.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV **1** (1988) 321–331
2. Hu, M.: Visual pattern recognition by moment invariants. IRA Trans. on Information Theory **17-8** (1962) 179–187
3. Del Bimbo, A., Pala, P.: Visual image retrieval by elastic matching of user sketches. IEEE Trans. Pattern Analysis and Machine Intelligence **19** (1997) 121–132
4. Jain, A., Zhong, Y., Lakshmanan, S.: Object matching using deformable template. IEEE Trans. Pattern Analysis and Machine Intelligence **18** (1996) 267–278
5. Xu, C., Prince, J.: Gradient vector flow: A new external force for snakes. CVPR (1997) 66–71
6. Cohen, L.: On active contour models and balloons. CVGIP: Image Understanding **53** (1991) 211–218
7. Amini, A., Tehrani, S., Weymouth, T.: Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints. ICCV (1988) 95–99
8. Williams, D., Shah, M.: A fast algorithm for active contours and curvature estimation. CVGIP: Image Understanding **55** (1992) 14–26
9. Sebe, N., Lew, M., Huijsmans, D.: Toward improved ranking metrics. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 1132–1141
10. Hampel, F., Ronchetti, E., Rousseeuw, P., Stahel, W.: Robust Statistic: The Approach Based on Influence Functions. John Wiley and Sons, New York (1986)
11. Murase, H., Nayar, S.: Visual learning and recognition of 3D objects from appearance. IJCV **14** (1995) 5–24