

Curve Fitting by a Sum of Gaussians

ARDESHIR GOSHTASBY AND WILLIAM D. O'NEILL

Electrical Engineering and Computer Science Department, University of Illinois at Chicago, Chicago, Illinois 60607-7053

Received April 14, 1992; revised April 13, 1994; accepted April 28, 1994

Gaussians are useful in multiscale representation of video data. An algorithm is presented which approximates a sequence of uniformly spaced single-valued data by a sum of Gaussians with a prescribed accuracy. The scale-space image [6] of the data is used to estimate the number of Gaussians and their initial parameters. The Marquardt algorithm (*J. SIAM* 11(2), 1963, 431–441) is then used to optimize the parameters. © 1994 Academic Press, Inc.

1. INTRODUCTION

Given a sequence of uniformly spaced single-valued data, we wish to approximate the data by a sum of Gaussians with a prescribed accuracy. Various methods for approximation of single-valued data have been proposed, and Cox [2] and McLain [5] provide excellent reviews of the methods. In data approximation, usually the form of the approximating function is not an issue, but rather the accuracy, speed, and ease of use of the function are. The problem addressed in this paper, however, specifically requires that the approximating function be a sum of Gaussians.

Unlike curve fitting by piecewise polynomials, where the coefficients of the polynomials hardly characterize the data, parameters of Gaussians approximating a data set, like parameters of sinusoids in the Fourier transform, characterize the data well. While the sinusoids in the Fourier transform describe the frequency characteristics of a data set, the Gaussians in the proposed method describe the data at varying resolutions. Multiscale representation of image data is becoming more and more in demand for browsing of video data at varying resolutions. When a data set is represented by a sum of Gaussians, a subset containing only very wide Gaussians will generate a coarse representation, and as narrower Gaussians are included, finer representations will be obtained.

In curve fitting by a sum of Gaussians the objective is to find the minimum number of Gaussians that can approximate a data set with a prescribed accuracy. The problem to be solved is, therefore, one of optimization, for which we will use the scale-space analysis of Witkin [6] and the least-squares optimization algorithm of Marquardt [4; 3, pp. 218–228]. Most least-squares optimization algo-

gorithms are based on one of two approaches. In one approach, the given function is expanded in a Taylor series and corrections are made at each iteration based on the assumption of local linearity. In another approach, a modification of the steepest-descent method is used to make the corrections. Algorithms based on Taylor series expansion may diverge, and those based on steepest descent may become very slow after a few iterations. The Marquardt algorithm combines both approaches to achieve a fast and definite convergence: steepest descent is used to determine the direction, and Taylor series expansion is used to determine the size of a correction.

Like many other optimization algorithms, the Marquardt algorithm depends on the provided initial values to determine a solution. For one set of initial values it may find the optimal solution, while for another set it may find only a suboptimal solution. In this paper, by scale-space analysis a set of initial parameters are determined which will lead the Marquardt algorithm either to the optimal solution or to a good suboptimal solution. A suboptimal solution is considered good if it approaches the optimal solution as the approximation error is reduced. The contribution of this work is considered to be the combination of scale-space analysis to identify the Gaussians composing a data set and optimization to determine the Gaussian parameters.

2. STATEMENT OF THE PROBLEM

Given a sequence of uniformly spaced points $\{x_k : k = 1, \dots, N\}$ with associated data values $\{y_k : k = 1, \dots, N\}$, we would like to approximate the data by a sum of Gaussians in the form

$$y = f(x) \equiv \sum_{i=1}^n A_i e^{-(x-x_i)^2/2\sigma_i^2}, \quad (1)$$

such that

$$\sqrt{\frac{1}{N} \sum_{k=1}^N (f(x_k) - y_k)^2} < \varepsilon. \quad (2)$$

$y = f(x)$ is a single-valued curve with parameters $\{A_i, x_i, \sigma_i : i = 1, \dots, n\}$ to be determined so that the curve fits the data with prescribed accuracy ε . Note that in this approximation, the number of Gaussians used, n , also has to be determined, and it is desirable to find the smallest such n .

Since a Gaussian is determined by three parameters (A_i, x_i, σ_i), for a known value of n , this problem is equivalent to that of solving a system of $3n$ nonlinear equations by the least-squares method using $N \gg 3n$ data points. To solve a system of nonlinear equations, we use the Marquardt algorithm. In this algorithm, starting from an initial set of parameters, the parameters are iteratively modified until a local minimum is reached in the sum-of-squared error between the data and the approximating function. The obtained solution depends on the provided initial parameters. In the following, a method that determines a set of initial parameters to produce a good suboptimal solution is described.

3. ESTIMATION OF THE INITIAL PARAMETERS

We first use a data set that represents uniformly spaced samples from a sum of n known Gaussians and try to determine the Gaussians. A Gaussian has two inflection points, causing its second derivative to produce two zero-crossings. When n Gaussians are combined, the second derivative of the combination will have at most $2n$ zero-crossings. Two Gaussians very close to each other may produce only one pair of zero-crossings as demonstrated in Figs. 1a and 1c, and if one Gaussian has a considerably smaller amplitude than its immediate neighbor, only zero-crossings from the larger Gaussian will be obtained as shown in Figs. 1b and 1d.

When a function contains only one Gaussian, from its zero-crossings the position and the standard deviation of the Gaussian can be accurately determined. However, when a function contains two or more Gaussians, it is not obvious which two zero-crossings belong to the same Gaussian, and even when zero-crossings of each Gaussian are identified, it is not possible to determine the positions and the standard deviations of the Gaussians accurately. This is because zero-crossings move from their true positions as Gaussians are combined.

Observe that we are not given a continuous function but rather a set of equally spaced samples from it and from the samples we have to determine the Gaussians. By determining the second derivatives of the data and locating the zero-crossings we will determine the number of Gaussians in the data. To study the zero-crossings of the second derivative of a data set at varying resolutions, the scale-space image of the data will be used [6]. The scale-space image of a data set shows the zero-crossings of the second derivative of the data at varying resolutions.

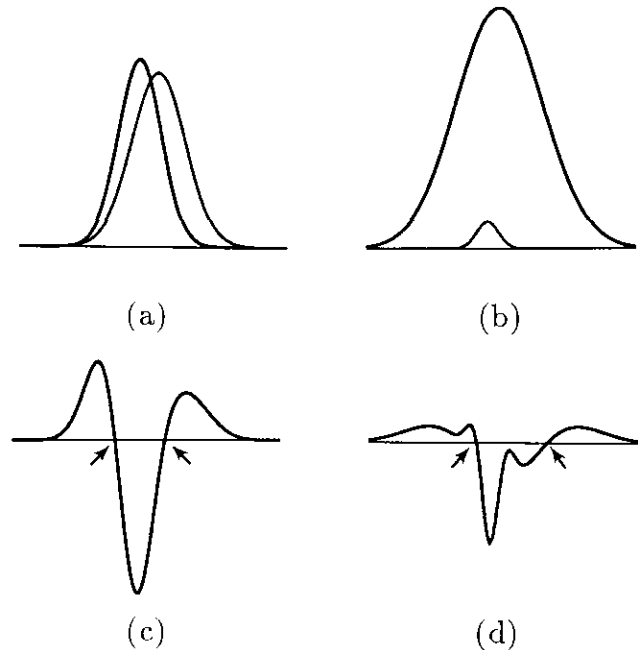


FIG. 1. (c) and (d) are the second derivatives of the sums of Gaussians shown in (a) and (b), respectively. These figures show cases where fewer zero-crossing pairs than the number of Gaussians in a function are obtained. Two very close Gaussians produce only one pair of zero-crossings as shown in (a) and (c). Two Gaussians with very different amplitudes produce only one pair of zero-crossings as shown in (b) and (d). The arrows indicate the zero-crossings.

We use the scale-space image of a data set to estimate the number and the parameters of Gaussians in the data. Then we will use the estimated parameters as initial values in the Marquardt algorithm to determine the optimal parameters.

If a data set is smoothed with a Gaussian filter [1, p. 156], as the standard deviation of the filter is increased, zero-crossings from smaller Gaussians disappear and only zero-crossings from dominant Gaussians remain [6]. The standard deviation of the filter may be taken large enough that only two zero-crossings remain in the smoothed data. These zero-crossings correspond to the most dominant Gaussian in the data. This is visually demonstrated in Fig. 2. Figure 2a shows nine Gaussians which when combined and sampled at equally spaced intervals produce the data of Fig. 2b. The amplitudes, positions, and standard deviations of these Gaussians are shown in columns 2, 3, and 4 of Table 1, respectively. To construct the scale-space image of Fig. 2b, the data are smoothed with Gaussian filters of increasing standard deviations, starting from $\sigma = 1.0$ with increments of $\Delta\sigma = 1.0$ until the second derivative of the smoothed data produces only two zero-crossings. An image is constructed from the zero-crossings such that the rows show the standard deviations of the smoothing filters and the columns show the positions

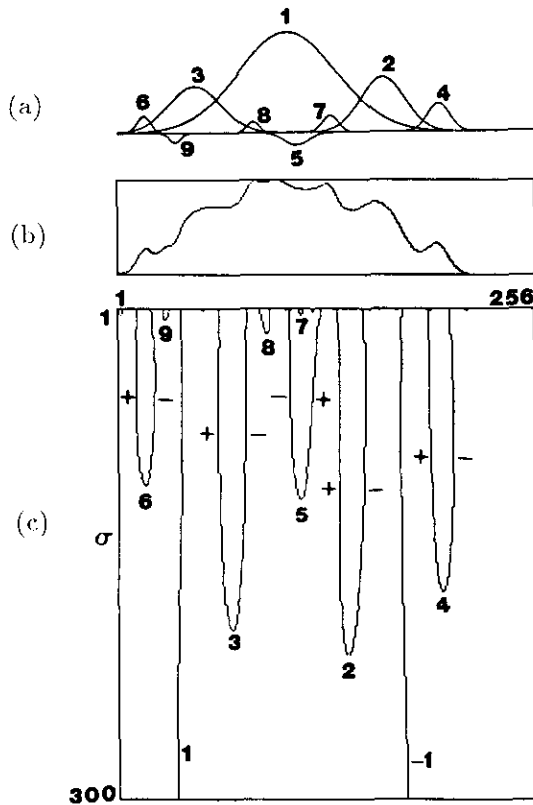


FIG. 2. Relation between Gaussians in a data set and the scale-space image of the data. (a) Nine arbitrary Gaussians. (b) Data obtained by combining the nine Gaussians and taking 256 samples at equally spaced intervals. (c) The scale-space image of the data. The vertical axis in (c) shows the standard deviations of the smoothing filters and the horizontal axis shows the zero-crossings of the second derivative of the smoothed data. Row σ in the scale-space image shows the zero-crossings of the second derivative of the data after being smoothed with a Gaussian of standard deviation σ . Note the correspondence between the arches in (c) and the Gaussians in (a).

of obtained zero-crossings. This image is known as the scale-space image [6] and contains valuable information about Gaussian components in the data.

Note that by decreasing the resolution of the data (by increasing the standard deviation of the smoothing filter), two zero-crossings that belong to the same Gaussian join in an arch. It is known that when a Gaussian filter is used in the smoothing, a zero-crossing contour never disappears or branches into two contours when the resolution of the data is increased [6, 8]. Therefore, the arches in a scale-space image may be used to determine the zero-crossing pairs of a Gaussian in the data. Although the positions of arches may not fall exactly under their corresponding Gaussians, they approximate them well enough to estimate the number of Gaussians present in the data.

The scale-space image of a data set may miss small Gaussians or produce only one arch for two very close Gaussians, but it can detect most Gaussians in the data. It will be shown that Gaussians that are not detected in the first attempt can be found later by subtracting the detected Gaussians from the data and using the residuals as new data in the scale-space analysis.

The arches in a scale-space image are labeled in the order in which they emerge while the resolution of the data is increased. One Gaussian is considered more "dominant" than another in a data set if its corresponding arch has a smaller label. Each arm of an arch can be given a "+" or "-" tag to show the sign of the amplitude of the underlying Gaussian. An arm of an arch is tagged "+" if the first derivative of the data at the corresponding zero-crossing is positive and is tagged "-" if its first derivative is negative. Tagging in this manner makes the "+" arm of a positive Gaussian appear to the left of its "-" arm, and the "+" arm of a negative Gaussian appear to the right of its "-" arm; see Fig. 2c.

TABLE 1
Approximation of Data of Fig. 2b by the Sum of Nine Gaussians Using Different Initial Values in the Marquardt Algorithm

i	Original			Optimal			Suboptimal			Arbitrary		
	A_i	x_i	σ_i	A_i	x_i	σ_i	A_i	x_i	σ_i	A_i	x_i	σ_i
1	64.0	108.0	40.0	60.7	105.1	36.8	60.7	105.1	36.8	-14.9	34.9	7.3
2	34.0	170.0	16.0	40.3	169.9	17.9	40.3	169.9	17.9	-5.9	6.1	5.9
3	30.0	50.0	16.0	31.3	50.5	16.4	31.3	50.5	16.4	54.3	56.1	26.4
4	16.0	200.0	10.0	14.8	201.4	9.7	14.8	201.4	9.7	29.4	93.8	11.9
5	-8.0	112.0	8.0	-5.0	113.1	7.3	-5.0	113.1	7.3	81.6	145.9	28.4
6	10.0	18.0	6.0	9.7	18.8	5.8	9.7	18.8	5.8	-29.4	148.7	11.0
7	10.0	132.0	7.0	14.1	132.6	7.9	7.4	89.3	5.4	15.5	206.4	15.8
8	-6.0	36.0	4.0	-6.4	37.0	4.3	-6.4	37.2	4.3	-5.3	214.4	7.0
9	7.0	88.0	5.0	7.4	89.3	5.4	14.1	132.6	7.9	-6.2	224.1	10.5

Note. Columns 2-4 show amplitudes, positions, and standard deviations of Gaussians of Fig. 2a from which data of Fig. 2b were sampled. Columns 5-7 show parameters of Gaussians computed by the Marquardt algorithm when columns 2-4 of the table were used as the initial values. Columns 8-10 show Gaussians computed by the Marquardt algorithm when parameters of the nine most dominant Gaussians obtained by relations (3)-(5) were used as the initial values. Columns 11-13 show obtained Gaussians when initial parameters computed from relation (6) were used in the Marquardt algorithm.

To estimate the positions and standard deviations of Gaussians approximating a data set, the zero-crossings of the data at the highest resolution are used. This is because when a data set is smoothed with a wider filter, the zero-crossings usually move by larger amounts. Assuming the positions of labels $+i$ and $-i$ in the first (top-most) row of a scale-space image are x_{i_1} and x_{i_2} , the position and standard deviation of the i th Gaussian are estimated, correspondingly, by

$$x_i = (x_{i_1} + x_{i_2})/2 \quad (3)$$

and

$$\sigma_i = |x_{i_1} - x_{i_2}|/2. \quad (4)$$

In this manner, the positions and standard deviations of all Gaussians in a data set can be estimated. Once these parameters are estimated, they are used in Eq. (1), and by the standard least-squares method the amplitudes of the Gaussians are estimated too. The amplitudes of n Gaussians, whose positions and standard deviations are known and minimize the sum of the squared error between their sum and the given data, satisfy the system of linear equations

$$\sum_{i=1}^n A_i \sum_{k=0}^N e^{-(x_k-x_i)^2/2\sigma_i^2} - (x_k-x_i)^2/2\sigma_i = \sum_{k=0}^N y_k e^{-(x_k-x_j)^2/2\sigma_j^2} \quad j = 1, \dots, n. \quad (5)$$

We see that if given data are samples from a sum of Gaussians, by analyzing the scale-space image of the data we can estimate the number and the parameters of the Gaussians from formulas (3)–(5). If an arbitrary data set is given, by analyzing its scale-space image, we will approximate the data by a sum of Gaussians the scale-space image of which is similar to that of the given data.

Figure 3b shows the scale-space image of the data shown in Fig. 3a. The contour labeled "0" is not an authentic contour, but rather is a side effect of the Fourier transform used to compute the zero-crossings by image convolution [1, p. 169]. The Fourier transform assumes that the given data set cyclically repeats itself. In other words, it assumes that the first data point follows the last data point, and if the two are not the same, an artificial discontinuity will be obtained, creating a false contour in the scale-space image. A false contour is easy to detect because it always appears in either the first or the last column of the scale-space image in high resolution. We label a false contour "0" and discard it at the time of analysis. Also, as can be observed in Fig. 3b, sometimes only one arm of an arch appears in the scale-space image. This happens when a large portion of a Gaussian falls

outside the window where data are sampled (see label "2" in Fig. 3b). In the calculations, we assume that the missing arm of an arch is at the opposite side of the nearest image border and at the same distance from the border.

Figure 3e shows the scale-space image of Fig. 3d, which is the sum of the Gaussians of Fig. 3c. The Gaussians in Fig. 3c were obtained by the Marquardt algorithm approximating the data of Fig. 3a with $\varepsilon = 2$. Note the close resemblance between the scale-space images. The difference between them is due to the fact that $\varepsilon \neq 0$ and small Gaussians are not used in the approximation. This shows that two data sets that approximate each other, one arbitrary and the other a sum of Gaussians, produce very similar scale-space images. Therefore, using information from the scale-space image of a data set, one can approximate the data by the predicted Gaussians.

The number of Gaussians used in an approximation determines its accuracy, and higher accuracy may be achieved by increasing the number of Gaussians used. In the following algorithm, the exact number of Gaussians needed to produce an approximation with a prescribed accuracy is determined.

4. THE APPROXIMATION ALGORITHM

Given a sequence of uniformly spaced single-valued data, the following algorithm approximates the data by a sum of Gaussians with a prescribed accuracy.

Input: Data points with associated values $\{(x_k, y_k) : k = 1, \dots, N\}$ and prescribed root-mean-squared error ε .

Output: The number n and parameters $\{A_i, x_i, \sigma_i : i = 1, \dots, n\}$ of Gaussians such that

$$e_n \equiv \sqrt{\frac{1}{N} \sum_{k=1}^N \left[\sum_{i=1}^n A_i e^{-(x_k-x_i)^2/2\sigma_i^2} - y_k \right]^2} < \varepsilon.$$

Method:

1. Determine the scale-space image of the data and label the arches as they emerge in the image from low to high resolution. Suppose m arches are obtained.

2. Estimate the parameters of the m Gaussians using formulas (3)–(5).

3. Set $n = 1$.

4. If $n > m$ then stop. Otherwise, use the parameters of the n most dominant Gaussians determined in step 2 as the initial values for the Marquardt algorithm and refine the parameters.

5. Suppose the error obtained by the Marquardt algorithm using the n Gaussians is e_n . If $e_n < \varepsilon$ stop. Otherwise, increment n by one and go to Step 4.

This algorithm increments the number of Gaussians used in the approximation by one at each iteration until the prescribed accuracy is reached. If the algorithm stops

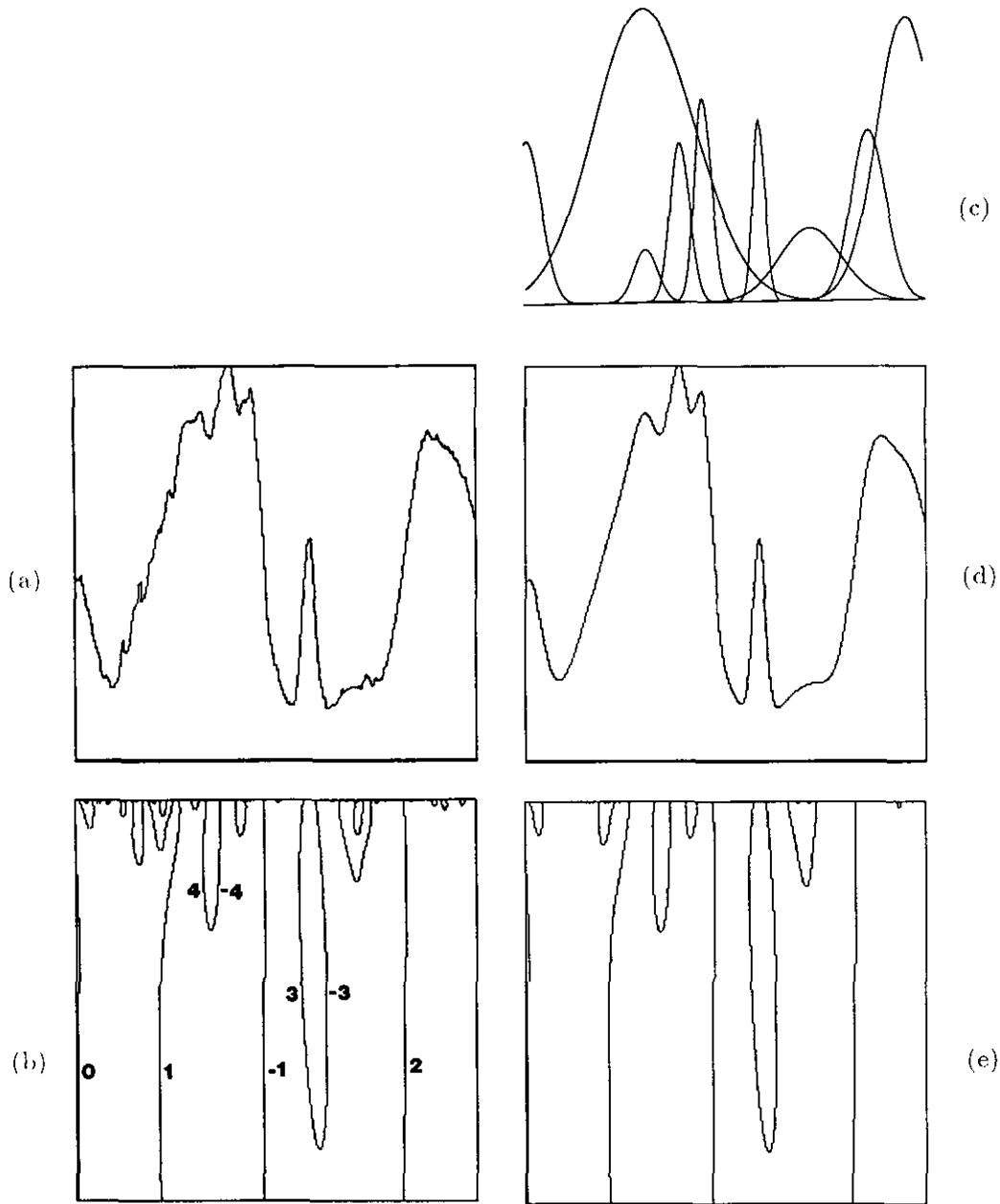


FIG. 3. Comparison of the scale-space images of an arbitrary data set and its approximation by a sum of Gaussians. (a) A data set with 256 samples representing one line of a natural 2-D image with values ranging from 0 to 255. (b) Scale-space image of (a). (c) Approximation of (a) by nine Gaussians. (d) Data obtained by combining Gaussians of (c) and taking 256 equally spaced samples. (e) Scale-space image of (d).

at Step 4, it means that the Gaussians detected by the scale-space analysis are not sufficient to approximate the data with the prescribed accuracy. In such a case, the m Gaussians determined thus far are subtracted from the original data, and the residuals are used in scale-space analysis to estimate the remaining Gaussians. The newly estimated Gaussians are added to those obtained in Step 2, and the algorithm is initiated again starting from Step 4. This process is repeated until the required accuracy is reached and the algorithm stops at Step 5.

This algorithm selects Gaussians in ascending order of

their arch labels from the scale-space image. The arches were labeled in the order they emerged in the scale-space image while increasing the resolution of the data. This ordering selects the next most dominant Gaussian in the data at each iteration.

5. RESULTS

Using data of Fig. 2b as input to the above algorithm and using nine Gaussians, the results of columns 8–10 in Table 1 were obtained. Columns 2–4 of the table show

the parameters of the original Gaussians in Fig. 2a. The labels on the Gaussians in Fig. 2a are not necessarily the same as the row numbers in Table 1. The labels in Fig. 2a were assigned according to the order in which their arches emerged in the scale-space image from low to high resolution. The row numbers in Table 1 show the order in which the proposed algorithm determined the Gaussians. Columns 5–7 show Gaussians obtained when initial values shown in columns 2–4 were used. Columns 8–10 show Gaussians obtained when initial values determined by relations (3)–(5) were used. Observe the close similarity between the optimal and suboptimal solutions. Although the order in which the Gaussians were obtained is different, both solutions contain the same set of Gaussians.

To compare the suboptimal solution with a solution that uses an arbitrary set of initial values, next, a set of initial values, obtained by

$$A_i = \frac{1}{N} \sum_{k=1}^N y_k, \quad x_i = \frac{x_N - x_1}{n+1} i, \quad \sigma_i = \frac{x_N - x_1}{n+1},$$

$$i = 1, \dots, n, \quad (6)$$

TABLE 2
Root-Mean-Squared Error between Data of Fig. 2b and Approximations to Them by a Sum of n Gaussians

n	ϵ_n		
	Optimal	Suboptimal	Arbitrary
1	6.72	6.72	6.72
2	3.65	3.65	5.44
3	2.85	2.85	3.00
4	2.65	2.65	2.90
5	1.73	1.73	2.49
6	0.63	0.63	1.84
7	0.53	0.60	0.74
8	0.31	0.33	0.54
9	0.27	0.27	0.52

Note. The second column shows rmse when parameters in columns 2–4 of Table 1 were used as initial values to the Marquardt algorithm. The third column shows rmse when initial values computed from relations (3)–(5) were used. The fourth column shows rmse when initial values obtained from relation (6) were used.

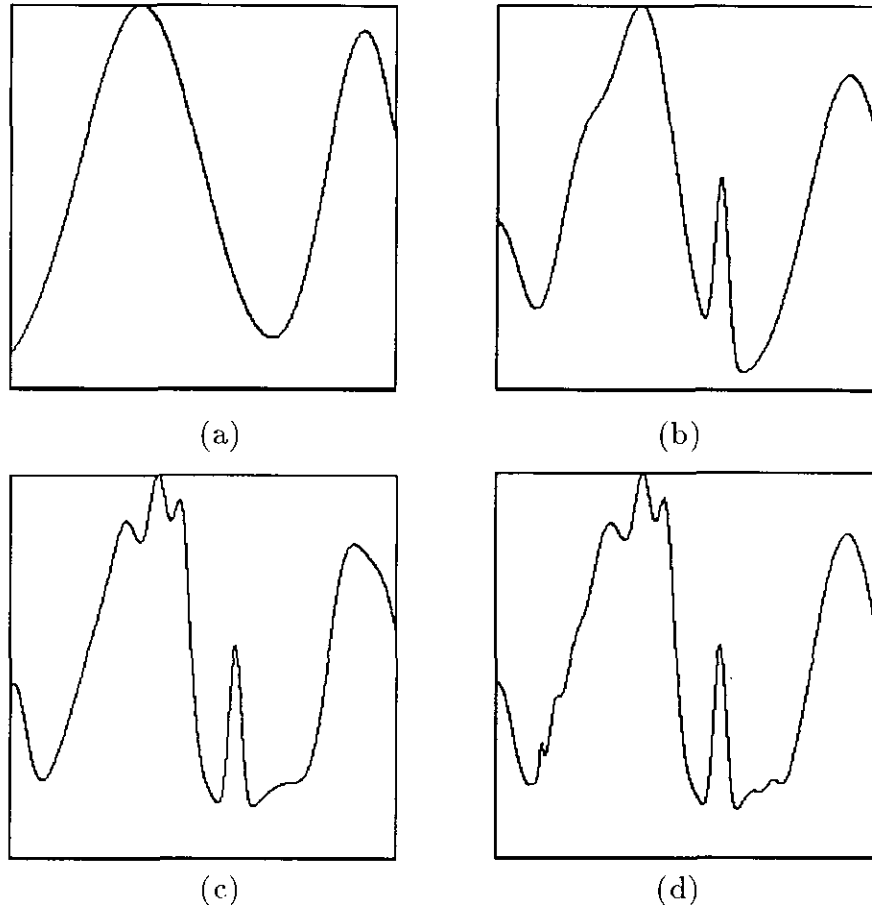


FIG. 4. (a), (b), (c), and (d) are intermediate results of the proposed algorithm, approximating data of Fig. 3a with 2, 5, 9, and 13 Gaussians, respectively. Errors in these approximations are $\epsilon = 9, 5, 2,$ and $1,$ respectively.

were used. These are reasonable initial values that one may use in the Marquardt algorithm. They show uniformly spaced Gaussians with amplitudes equal to the average of the data values and standard deviations equal to the spacing between the Gaussians. With these initial values, the Gaussians of columns 11–13 in Table 1 were obtained. As can be observed, the obtained Gaussians are completely different from the original Gaussians. The Gaussians obtained by the optimal and suboptimal solutions are slightly different from the original Gaussians because errors were introduced into the data when the sums of the Gaussians were sampled and quantized.

The root-mean-squared error (rmse) in these experiments is shown in the last row of Table 2. Table 2 shows rmse obtained by approximating the data of Fig. 2b with n Gaussians using different initial values. The second column of the table shows rmse when the first n initial values shown in columns 2–4 of Table 1 were used. The third column shows the rmse when parameters of the n most dominant Gaussians determined by the scale-space analysis were used. The fourth column shows the rmse when

initial values computed from relation (6) were used. The optimal and suboptimal solutions have produced the same results except for $n = 7$ and $n = 8$. Mistakes have occurred in selecting some of the smaller Gaussians. The solution obtained by the arbitrary initial values except when $n = 1$ is completely different from the optimal solution.

Applying the proposed algorithm to the data of Fig. 3a with $\varepsilon = 1$, the scale-space analysis detected 31 Gaussians. Thirteen of these were sufficient to approximate the data with the required accuracy. Some of the intermediate steps of the approximation are shown in Fig. 4. Root-mean-squared error between the original data (Fig. 3a) and its approximations shown in Figs. 4a, 4b, 4c, and 4d were 9, 5, 2, and 1, respectively.

In a third experiment, 256 uniformly spaced samples from two cycles of a sine wave with amplitude 128 were used as input data (see Fig. 5a). The scale-space image of this data is shown in Fig. 5b. There are only four zero-crossings in the image. These zero-crossings provided estimates to three Gaussians, which when used in the Marquardt algorithm produced the approximation of Fig.

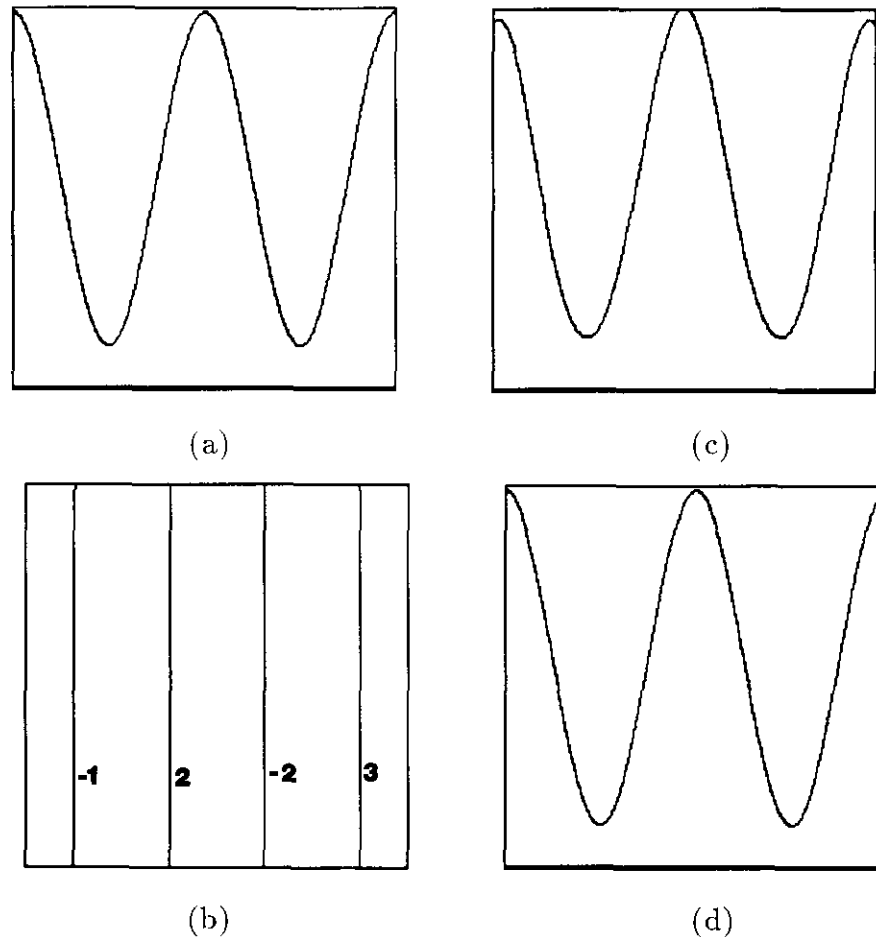


FIG. 5. (a) and (b) show a sine wave and its scale-space image, respectively. (c) and (d) show approximations to (a) with three and seven Gaussians, respectively.

5c with $\varepsilon = 2$. Subtracting 5c from 5a and analyzing the scale-space image of the residuals resulted in five more Gaussians. Figure 5d shows approximation to data of 5a with seven Gaussians and $\varepsilon = 1$.

The computational complexity of the proposed algorithm varies with the input data. In the above two examples, it took 10.96 and 1.63 μs to obtain approximations to Figs. 3a and 5a, respectively, on an IBM 3090 computer. Of this time, 0.5 μs was spent in each case in obtaining the scale-space images, and the rest was spent by the Marquardt algorithm in optimizing the parameters.

6. CONCLUSIONS

Gaussians are useful in multiscale representation of video data. An algorithm for fitting a sum of Gaussians to a sequence of uniformly spaced single-valued data with a prescribed accuracy was described. The algorithm estimates the number and the parameters of Gaussians by the scale-space analysis of the data. It then uses the Marquardt algorithm to optimize the parameters. The algorithm is suboptimal and finds the dominant Gaussians in the data correctly while occasionally making mistakes in

determining the smaller Gaussians. Results indicate that as the approximation error is reduced, the suboptimal solution approaches the optimal one.

REFERENCES

1. K. R. Castleman, *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
2. M. G. Cox, A survey of numerical methods for data and function approximation, in *The State of the Art in Numerical Analysis* (D. Jacobs, Ed.), pp. 627-668, Academic Press, New York, 1977.
3. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.
4. D. W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, *J. SIAM* **11**(2), 1963, 431-441.
5. D. H. McLain, Interpolation methods for erroneous data, in *Mathematical Methods in Computer Graphics Design* (K. W. Brodlie, Ed.), pp. 87-104, Academic Press, New York, 1980.
6. A. P. Witkin, Scale space filtering: A new approach to multi-scale description, *Proc. Image Understanding 1984*, 79-95.
7. A. L. Yuille and T. A. Poggio, Fingerprint theorems for zero-crossings, *J. Opt. Soc. Am.* **2**(5), 1985, 683-692.
8. A. L. Yuille and T. A. Poggio, Scaling theorems for zero-crossings, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**(1), 1986, 15-25.