Image Search by Features of Sorted Gray level Histogram Polynomial Curve

Awais Adnan, Muhammad Ali, and Amir Hanif Dar

***Abstract*—**Image Searching was always a problem specially when these images are not properly managed or these are distributed over different locations. Currently different techniques are used for image search. On one end, more features of the image are captured and stored to get better results. Storing and management of such features is itself a time consuming job. While on the other extreme if fewer features are stored the accuracy rate is not satisfactory. Same image stored with different visual properties can further reduce the rate of accuracy. In this paper we present a new concept of using polynomials of sorted histogram of the image. This approach need less overhead and can cope with the difference in visual features of image.

***Keywords*—**Sorted Histogram, Polynomial Curves, feature points of images, Grayscale, visual properties of image.

I. Introduction

USE of digital contents from different sources is increasing rapidly. We daily use thousands of images collected from different sources and channels like scanners, digital cameras, CD, DVDs, and also from digital libraries. In many situations we have and image and we want to search that image in our database for example trademark and logo of some organization, flags, cartoon etc.

In all the above mentioned and many other situations we have an image and we want to search the image in a databases (it can be a well maintained database or just a collection of images stored at various locations). An efficient image search mechanism is needed for all such requirements.

One popular image feature is color histogram that can be used to characterize the color distribution of an image [1]. Color histogram is used in various domains like HSV, YUM etc. [2], [3]. This color histogram can be used to retrieve image from database. However there are two problems with such systems. The first one is to collect all images, store, and managed as database. Even if the original image is not stored, its color histogram needed to be stored properly.

Speed of searching is the other limitation of such systems. Matching of color histogram of the query image with the database which can contain hundred thousands images takes enough time.

Some fast algorithms [4]-[6] are proposed to address such problems. All these methods eliminate unnecessary resolution and work with a low resolution image. These are only applicable for removing of early candidates to reduce the search space. However computation is still required at different level for different resolution.

In this paper we proposed an efficient system for image searching using polynomial equation of sorted grayscale histogram of image.

We have selected features of the image in such a way that even two different version of the image with different contrast, colors, intensity , size and other visual properties are comparable.

We have two set of features. First is the basic one that can be used for indexing. These features are directly extracted from sorted histogram. The other set is coordinates of polynomial curve of the degree n that best fits for second half of sorted histogram curve. This polynomial curve can be from degree 1 to degree 5. The degree of polynomial curve fits with less absolute error is the second level index.

II. Curve Fitting

Curve fitting is a procedure in which we pass a curve through a set of points in such a way that the estimated curve shows the relationship between the two quantities as correctly as possible. Although it is possible to pass the curve through all points (interpolation) but in our case it is not desirable.

The first thing is to decide how many degrees of freedom is required. In out case we estimate curves from degree 1 to 5 and select the one with least error We have used least square method for curve approximation.

Least square mth degree polynomial is

$$y = a_0 + a_1 x + a_2 x^2 + \dots a_m x^m \tag{1}$$

To approximate the given set of data, $(x_1, y1)$, (x_2, y_2), …(x_n, y_n), where, the best fitting curve *f(x)* has the least square error, i.e.

$$\prod = \sum_{i=1}^{n} [y_i - f(x_i)]^2 = \min \tag{2}$$

Where a_0, a_1... a_m are unknown coefficients to obtain the least square error, the unknown coefficients a_0, a_1, a_2, … and a_m must yield zero first derivatives.

A. Adnan and M. Ali are with Institute of Management Sciences Peshawar, Pakistan.

Dr. Amir Hanif Dar is with College of Electrical & Mechanical Engineering, National University of Sciences & Technology (NUST), Pakistan.

$$\begin{cases}\dfrac{\partial \Pi}{\partial a_0} = 2\sum_{i=1}^{n}\left[yi - (a_0 + a_1x_i + a_2x_i^2 + \ldots + a_mx^m)\right] = 0 \\ \dfrac{\partial \Pi}{\partial a_1} = 2\sum_{i=1}^{n} x_i\left[yi - (a_0 + a_1x_i + a_2x_i^2 + \ldots + a_mx^m)\right] = 0 \\ \dfrac{\partial \Pi}{\partial a_2} = 2\sum_{i=1}^{n} x_i^2\left[yi - (a_0 + a_1x_i + a_2x_i^2 + \ldots + a_mx^m)\right] = 0 \\ \vdots \\ \dfrac{\partial \Pi}{\partial a_m} = 2\sum_{i=1}^{n} x_i^m\left[yi - (a_0 + a_1x_i + a_2x_i^2 + \ldots + a_mx^m)\right] = 0\end{cases} \quad (3)$$

Expanding the above equations, we have

$$\begin{cases}\sum_{i=1}^{n} y_i = a_0\sum_{i=1}^{n}1 + a1\sum_{i=1}^{n}x_i + a2\sum_{i=1}^{n}x_i^2 + \cdots + a_m\sum_{i=1}^{n}x_i^m \\ \sum_{i=1}^{n} x_iy_i = a_0\sum_{i=1}^{n}x_i + a1\sum_{i=1}^{n}x_i^2 + a2\sum_{i=1}^{n}x_i^3 + \cdots + a_m\sum_{i=1}^{n}x_i^{m+1} \\ \sum_{i=1}^{n} x_i^2y_i = a_0\sum_{i=1}^{n}x_i^2 + a1\sum_{i=1}^{n}x_i^3 + a2\sum_{i=1}^{n}x_i^4 + \cdots + a_m\sum_{i=1}^{n}x_i^{m+2} \\ \vdots \\ \sum_{i=1}^{n} xi^my_i = a_0\sum_{i=1}^{n}x_i^m + a1\sum_{i=1}^{n}x_i^{m+1} + a2\sum_{i=1}^{n}x_i^{m+2} + \cdots + a_m\sum_{i=1}^{n}x_i^{m+m}\end{cases} \quad (4)$$

The unknown coefficients a_0, a_1, a_2 … and a_m can hence be obtained by solving the equations given in (4).

III. Overall System Architecture

This section is the brief summary about how the system works. We have a set of images that is distributed at various location (on the same machine in our case, however it can be distributed over various machines), these images are stored as a simple image files (images stored in some databases can also be used). In our system database contains only the features of the images. These features are extracted at the time of image creation, or when it was used for the first time. These can also be collected by issuing some explicit command for feature extraction. Even it can be done automatically as background process. Database contains links of the image not the actual image which reduce the size of database.

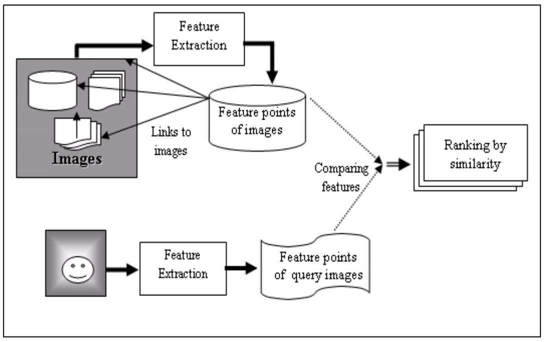


Fig. 1 Architecture of the system

When an image is put as query, the system extracts the features of the query image using same algorithms with some relaxing parameters. These features are then used as SQL query to extract results from the database with an increasing tolerance from 0.01 to 0.10. Best three results are selected. The whole system is summarized in Fig. 1.

IV. Feature Extraction from an Image

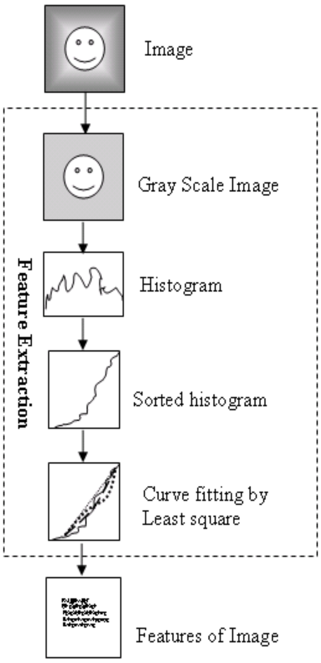


Fig. 3 Feature extraction

A. Gray Scale Image

We have different type of images stored in different formats for example color images and gray level images stored as BMP , JPEG , TIFF, PNG. In first step we convert the image into 8-bit gray level. Color image from RGB is converted to grayscale using (5).

$$Rn = 0.3 \times Rn + 0.59 \times Gn + 0.11 \times Bn \quad (5)$$

Where R_n, G_n , and B_n are Red , Green , and Blue components of the pixel n respectively.

This grayscale image or an image that is already in grayscale is then converted into 256 Graylevel using (6).

$$R_{n(256)} = (R_{n(256)} \times 256) \div R_{\max(256)} \quad (6)$$

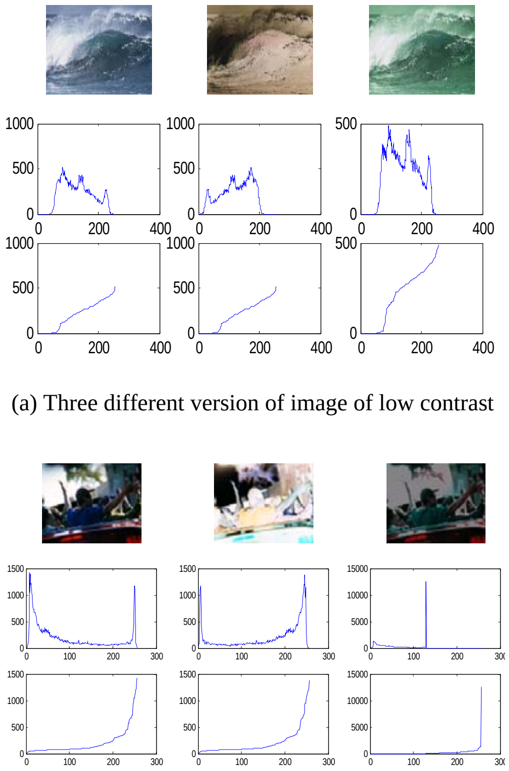
Where $R_{\max(256)}$ is the Maximum gray level in the image.

B. Histogram Calculation

In the next step histogram is calculated for the images with the gray levels in the range 0-256 using a function h(R_k) = N_k, where R_k is the kth level and N_k is the number of pixels in the image having gravel level R_k

C. Histogram Sorting.

Histogram is a function of gray level and number of pixels in that gray level. If we have same image with different color and visual properties we will get different histogram as shown in Fig. 2. First row of the image (Fig 2 (a) and Fig. 2 (b)) shows the different version of images. Second row shows histogram and third row shows sorted histogram. It is clear from the figure that histogram of different version of same image my be different but the sorted histogram is similar having the same slope and features for all version of same image having different visual properties.



(a) Three different version of image of low contrast

(b) Three different version of image of high contrast

Fig. 2 Same image having different visual properties and its affect on histogram and sorted histogram

Number of pixels in an image is related with the size of image. If we double the size of image, number of pixels increase with the same ratio. To overcome the possible difference in the size of query-image and actual imagehaving link in the database, we translated the height of the histogram in the range 0-256 by the (7).

$$M_k = ((N_k - N_{min}) \div N_{max}) \times 256 \qquad (7)$$

Where N_k is the number of pixels having gray levels k,
N_{min} is the smallest value of N_j for j = 1 - 256
N_{max} is the largest valve of N_j for j = 1 - 256

In this way we got histogram for 256 gray levels (at x axes) with values from 0 to 256 (at y axes). This histogram is sorted in ascending order with 0 at initial value and 256 as the final value. Form this sorted histogram; we get first set of feature points.

D. Curve Fitting by the Method of Least Square

In the next step we estimated the polynomial curve of degree 1 to 5 for the sorted histogram using method of least squares. And then error is calculated by following equation

$$E_n = \sum |M_k - P_n(k)| \qquad \text{For } n=1 \text{to} 256 \qquad (8)$$

where M_k is value in sorted histogram for k and $P_n(k)$ is value calculated by polynomial of degree n for point k
Polynomial for least value of E is selected i.e.

$$P_{min}(x) = b_0 + b_1x + b_2x^2 + b_3x^3 + \cdots + b_nx^n \qquad (9)$$

Where $P_{min}(x)$ is polynomial equation for which E is minimum $b_0, b_1, \cdots, b_n$ are coefficients of the curve which are second set of feature points

V. Feature Points of Sorted Histogram

We used sorted histogram for feature point extraction. Two groups of feature points are used. First group is used for indexing and second is used for searching

Lower bound of the curve is the 1st point above threshold t_1 i.e. Lower bound = M_k if $Mk > t_1$.

Midpoint is the first point where difference between two successive values is grater than second threshold t_2 i.e.

Mid = M_k if $M_k - M_{k-1} >$ t2

Slope of the line from Lower-bound to midpoint is calculated by using following equation

$$\text{Slope} = M_{mid} \;/\; \text{Mid} - (\text{Lower-bound}) \qquad (10)$$

Equation (7) is used to calculate Min which is the smallest value in the histogram.

Next the best-fit curve of degree 1 to degree 5 is calculated. The curve with smallest error is selected. Curve degree and coordinates of the polynomials are other feature points that are used for image search. From these coordinates, mfactor is calculated which is the multiplying factor to change the curve in standard way to overcome difference of brightness and contrast in the image in database and query images.

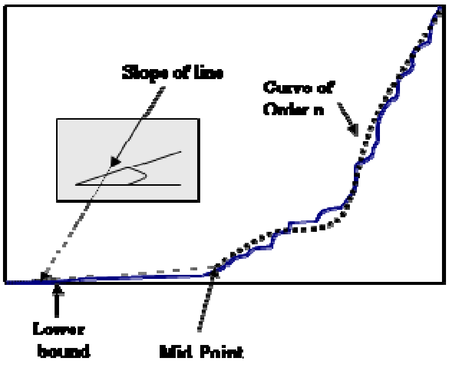


Fig. 4 Feature points from sorted curve

VI. Database for Image Links

To maintain the links for the images we have used a very simple database of only one table. Length of a record is 290 Bytes in which only 35 Bytes are used to store features of image other 255 are used to store link of the image. Dynamic length for this field can be used to reduce record length. Structure of the table is shown in Table I.

TABLE I
TABLE STRUCTURE USED IN DATABASE

SNo	Field	Description
i.	IM_NO	Serial number in database
ii.	Image_link	Pointer (link) to the image
iii.	Lower_Bound	Starting of curve
iv.	Mid_value	Change of slope from linear to higher order
v.	Slope	Slope of 1st half
vi.	Min	Minimum gray level value
vii.	Curve_Degree	Degree of curve
viii.	mFactor	Factor that should be multiplied for getting standard curve
ix.	c1	First coefficient for curve , 0 if not used
x.	c2	Second coefficient for curve , 0 if not used
xi.	c3	Third First coefficient for curve , 0 if not used
xii.	c4	Fourth coefficient for curve , 0 if not used
xiii.	c5	Fifth coefficient for curve , 0 if not used
xiv.	c6	Sixth coefficient for curve , 0 if not used

VII. Results

This concept is tested through a small application developed in MATLAB 6.5. More than 4,000 images are used this test.

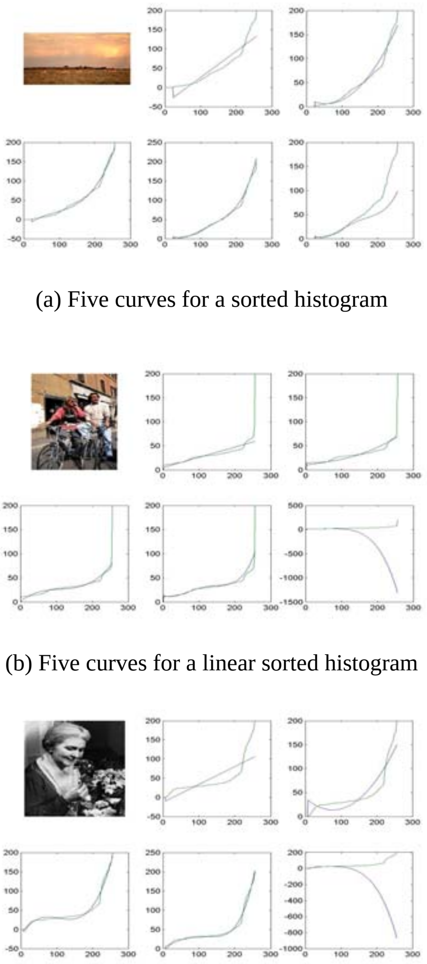
Microsoft Access is used as Database management System. ODBC is used for linking the database to the Application. File name along with path of the file is used as link to the image.

From Fig. 6 to Fig. 8 show different type of results. Fig. 6 shows the results where the target images were acquired correctly. Form some images the correct image could not be founded and search engine gives us wrong results. This is because some time curve of other images becomes more close to the query image than the actual one. However in most of failure cases the actual image was extracted from the database but could not be ranked in best 3 candidates. There are some cases where the actual image could not even obtained by the query.

Fig. 8 shows the result where query image have different visual properties but still the correct images were obtained.

In Fig 8(a) and Fig 8(b) query image is the inverse of the original one. In Fig 8(c) the query image if much lighter than the original. In Fig. 8 (d-f) query image is trimmed and the original image was identified by querying with some part of image (more than 60% of the original image).

By selecting a random querying image this system was tested for about 1000 times. We got 92% time accurate results when the query was made by using image similar to the original one. When this system was tested with the image having 50% similar images and 50% with some modification in visual properties it gives 86% accurate results.



(a) Five curves for a sorted histogram

(b) Five curves for a linear sorted histogram

(c) Five curves for a sorted histogram of higher degree

Fig. 5 Curve fitting

References

[1] M. Swain and D. Ballard, “Color Indexing”, *International journal of Computer Vision* Vol., 7 no 1, 1991.

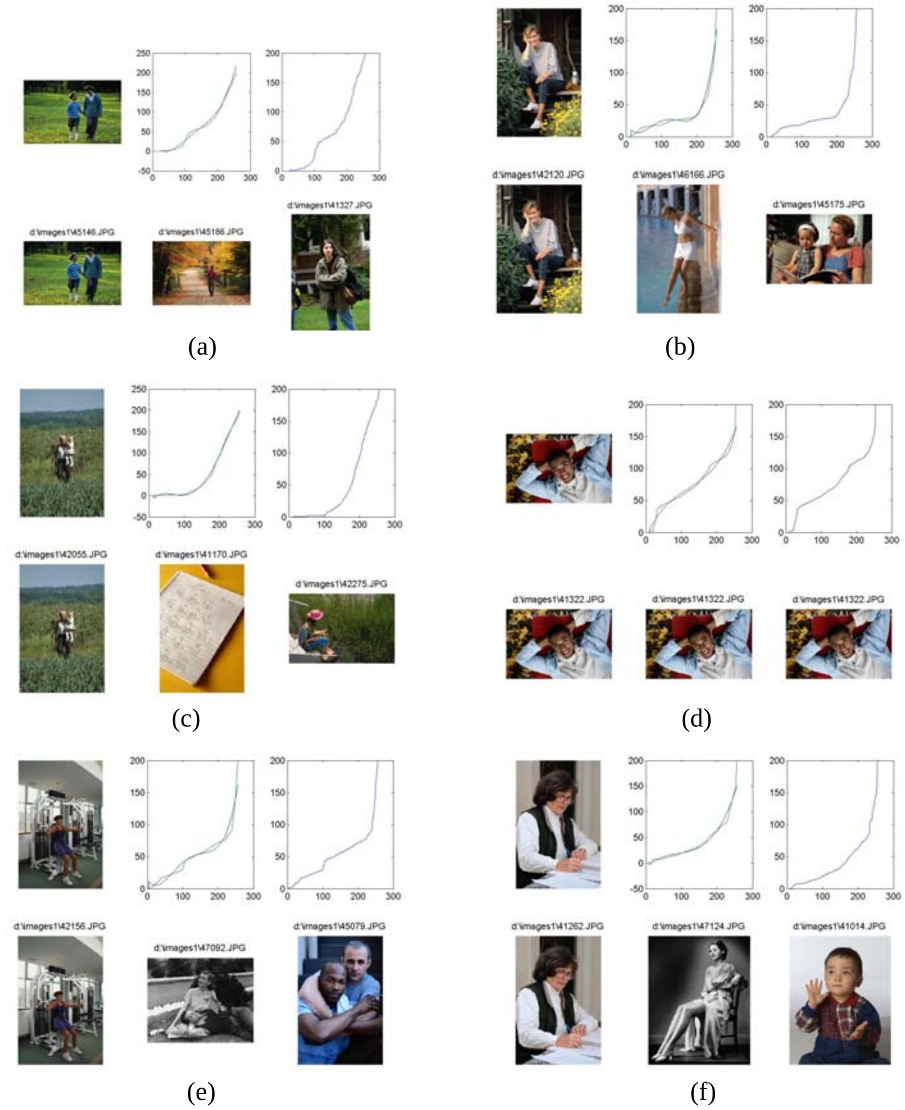
[2] R. W. G. Hunt, “Measuring Color”, *Ellis Harwood Limited, England, 1987.*

[3] K. M. Wong, C. H. Cheung and L. M Po “merged-color histogram for color image retrieval”, *Proceedings In International Conference, Image Processing* , Vol III pp.949-952, Sep 2002.

[4] J. Hafner, H. S. Sawhney, W. Equilz, M. Flicker and W. Niblack, “Efficient color histogram indexing for quadratic form distance functions”, *IEEE Trans. Pattern Anal. Machine Intell,* vol. 17, pp. 729-736, July, 1995.

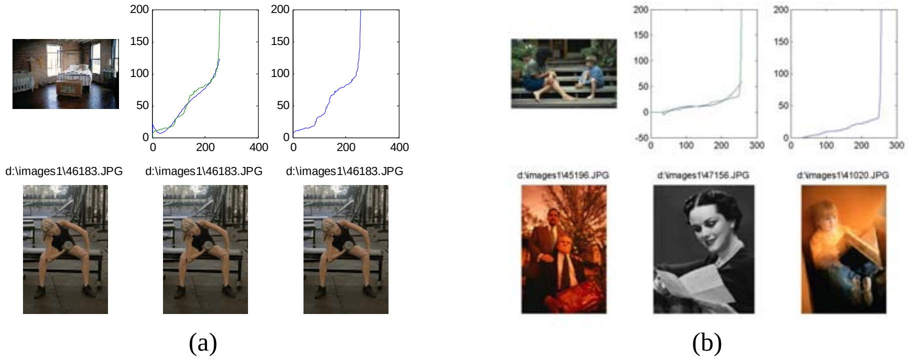
[5] A. P. Berman and L. G. Shapiro, “Efficient image retrieval with multiple distance measures”, *Proc. SPIE Storage and Retrieval for Image and Video Database,* vol. 3022, pp.12-21. Feb. 1997.

[6] B. C. Song, M. J. Kim and J. B. Ra, “A fast multi resolution feature matching algorithm for exhaustive search in large image databases”, *IEEE Trans. Circuits Syst Video Technol,* vol. 11, pp.673-678, May 2001.



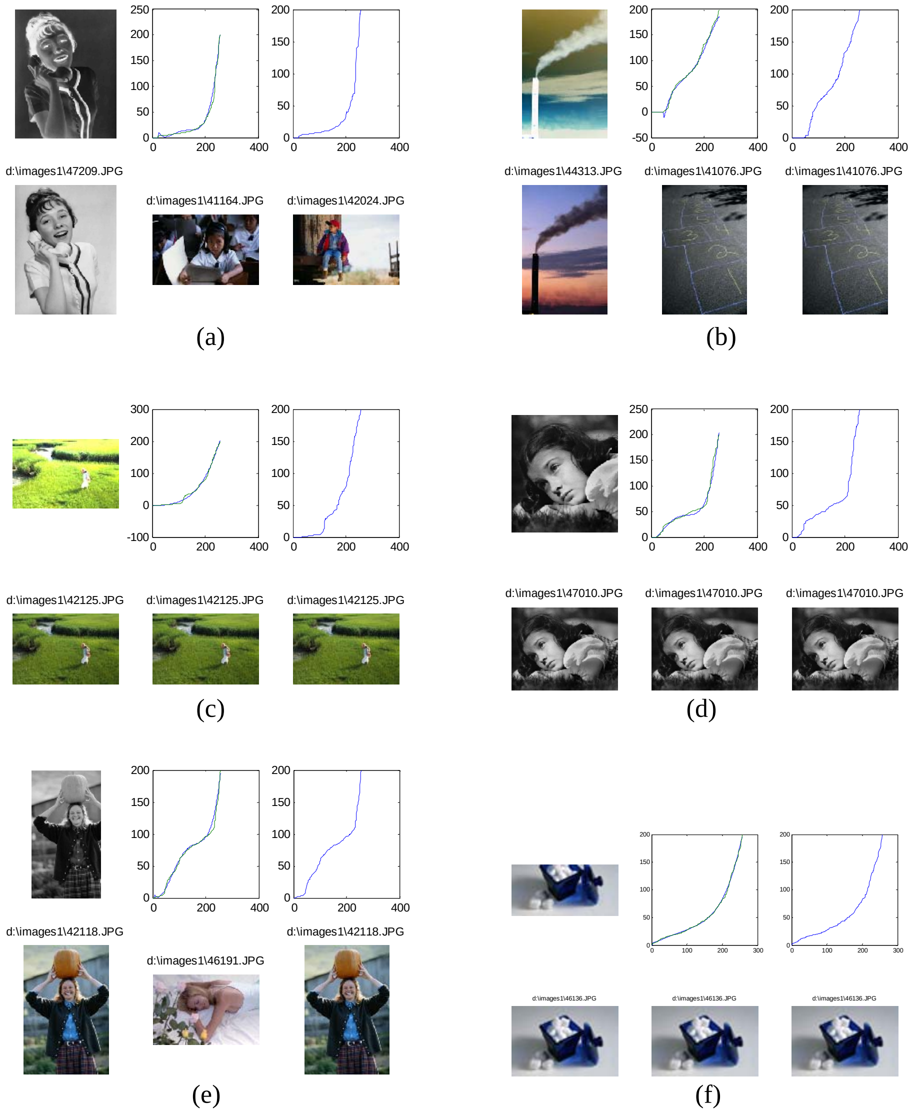
(a) (b) (c) (d) (e) (f)

Fig. 6 Search results where we get the correct image



(a) (b)

Fig. 7 Search results where correct image could not be founded



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8 Image search results using same image as query with different visual properties