Comparison Study Demosaicking of Bayer Color Filter Array

D. T. Naman
Department of Computer Science
College of Education Ibn Al-Haitham
University of Baghdad

Abstract

Image capturing in most still and motion digital cameras employs a charge coupled device covered with a monolithic array of color filters such that each pixel samples only one color band. Interpolation is necessary to restore the missing color samples in the resulting mosaic, this process referred to as demosaicking.

This paper gives an overview of demosaicking, presents few methods proposed in the literature, and concludes with a quantitative comparison of these methods for a set of images.

Introduction

Digital cameras have come a long way since their inception, and became so popular that they are considered the number one tool for taking images for mainstream consumers. However, in the split second it takes the camera to render the image, a few processes take place. These include color filter array interpolation, color calibration, anti-aliasing, infrared rejection, and white-point correction. Interested reader is referred to (1).

Color images are usually represented by a combination of at least three colors that are fairly representative of the gamut of the human visual system (HVS). One such composition is red, green, and blue, which was chosen because it corresponds nicely to the HVS's three cone types. Of course, they cannot cover the whole HVS gamut, i.e., they cannot represent all the possible visible colors.

Obviously, to sample these three colors, the camera needs three sensors; each having chromatic response that corresponds to one of the colors. The problem with such design is that the cost is quite high due to three main reasons; the cost of each sensor (1), the cost of the optical elements that split the incoming light, and the cost of aligning these elements properly. Although, this concept is used in professional high-end cameras, an alternative that can bring digital cameras to mainstream consumers is sought.

The solution is to use one sensor, but instead of sensing all the colors in each pixels simultaneously, only one color is sampled at one pixel. This is achieved by covering the sensor with a mosaic-colored mask, called color filter array (CFA). Thus, each pixel has its own spectrally selective filter to peer through. Fig. (1) shows two of the popular CFA patterns in use today (1). However, in the rest of this article, the focus goes to the Bayer CFA (2) shown in Fig. (1.a).

The reasons for sampling the green channel twice as much in the Bayer CFA (2) in Fig. (1.a) is due to the HVS higher luminance acuity compared to its chromatic acuity (3), which is better represented by the green channel since the luminance response curve of the eye peaks at around the frequency of the green light (around 550nm) (4).

Clearly, a CFA-sampled image is useless and each pixel's missing colors need to be recover; a process referred to as demosaicking. It is the topic of this article to discuss a few demosaicking methods for the Bayer CFA and compare them over a set of images. A brief literature survey is given next, which is followed by the different selected methods, together with their details and implementation nuisances. Then, results are given together with discussions before concluding the article.

A Brief Literature Survey

Due to the importance of this topic, a good deal of literature was devoted to it and plenitude of methods were devised. However, these methods can be grouped into main three categories, depending on how the demosaicking process is viewed; interpolation-based, image-reconstruction-based, and image-formation-based demosaicking (5).

In interpolation-based demosaicking, demosaicking is viewed as an interpolation problem. The solution vary widely from mathematical interpolation such as the one in method 1 to heuristic-based that attempts to utilize the HVS and the correlation between the different color channels such as the ones in methods 3-5.

In image-reconstruction-based demosaicking, image demosaicking is viewed as a reconstruction problem that attempts to utilize assumptions about the inter-channel correlation or the prior knowledge about the image in solving the problem mathematically (5) such as the one is method 6.

In image-formation-based demosaicking, image demosaicking is viewed as an inverse problem of the image formation process. These methods formulates a model that takes into account the transformations performed by the color filters, lens distortions, and sensor noise and determine the most likely output image given the measured CFA image (5). One such method is given by (6).

The Demosaicking Methods

This method presents the six different demosaicking methods, which are the focus of this paper.

Bilinear Demosaicking (BD)

Demosaicking through linear interpolation is probably one of the simplest demosaicking methods. For every color channel, the value at the missing pixels is found by taking the average of the nearby pixels of the same color. Thus, referring to Fig. (1.a), the interpolation for the missing red samples at pixels R43, R44, and R54 are given by:

$$R43 = \frac{R33 + R53}{2}, \quad R44 = \frac{R33 + R35 + R53 + R55}{4}, \quad R54 = \frac{R53 + R55}{2} \quad [1]$$

A similar procedure can be followed for the blue color. However, for green color, the interpolation is slightly different since there are more samples. The missing green sample at pixel G44 is found by:

$$G44 = \frac{G34 + G43 + G45 + G54}{4} \quad \text{.........[2]}$$

The implementation of bilinear interpolation is straight forward, and can be done in integer arithmetic; however, care should be taken not to cause an overflow during the arithmetic. Demosaicking through linear interpolation smooths edges due to its band-limiting nature, introducing some color infringes around the edges as shown in method 2, which are referred to as the zipper effect (4)(7). It also useless from any correlation between the different colors.

Edge-Directed Bilinear Demosaicking (EDBD)

In this algorithm, the interpolation of the green channel depends on the neighborhood. Where as, the blue and red channels are bilinearly interpolated similar to method 1. The decision of choosing the vertical or the horizontal pixels to interpolate a missing green pixel depends on the difference in their values that is the direction along the smaller difference is chosen to avoid interpolation across the edges and only interpolating along the edges (5).

Referring to Fig. (1.a), the differences along the horizontal and vertical directions are first calculated by $\alpha = |G43 - G45|$ and $\beta = |G34 - G54|$, then based on these values, G44 is found from (5)

$$\mathbf{G44} = \begin{cases} \frac{G43 + G45}{2} & \alpha < \beta \\ \frac{G34 + G54}{2} & \alpha > \beta \\ \frac{G34 + G43 + G45 + G54}{4} & \alpha = \beta \end{cases} \quad [3$$

Implementation wise, it is similar to the bilinear interpolation and the interpolated image also suffers from smoothing around the edges. However, it does not suffer from the zipper effect.

Constant Hue-Based Demosaicking (CHBD)

This method was proposed by Cok (8) and it is one of the first few methods used in digital cameras (4). It is based on the concept that the hue changes gradually and that there are no sudden changes in it except over the edges.

There are two steps in this method; in the first step, the luminance is bilinearly interpolated, and in the second, the hue (R/G and B/G) is interpolated, that is, the ratios of the colors to the green color. It should be noted that this definition is not correct for the hue, but it is valid for this method only (4). Referring to Fig. (1.a), R44 is found from

$$R44 = \mathbf{G44} \bullet \frac{\frac{R33}{\mathbf{G33}} + \frac{R35}{\mathbf{G35}} + \frac{R53}{\mathbf{G53}} + \frac{R55}{\mathbf{G55}}}{4} \quad \ldots\ldots[4]$$

Where the bold font indicates the interpolated value (5). A similar procedure is followed for the missing blue color.

This method has an advantage over the bilinear interpolation because it makes use of the luminance channel in interpolating the chrominance channel, that is, it utilizes the relation between the different channels.

Implementation wise, the second step requires care to avoid division by zero when the green channel is zero, therefore, in the included implementation, whenever a zero green is encountered, bilinear interpolation is used for that particular sample. Also, the implementation uses floating-point arithmetic and limits the range to [0, 255] since overflow might occur.

Gradient-Based Demosaicking (GBD)

This method was proposed by Laroche and Prescott (9) and it is in use in the KodakTM DCS200 digital camera (4). It involves two steps; in the first step, the luminance (green) channel is interpolated, and in the second step the differences between the interpolated luminance channel and the chrominance channels (green minus red and green minus blue) are interpolated. These difference are used to reconstruct the chrominance channels (4).

To find the green value at G44 in Fig. (1.a), a decision need to be made whether the horizontal, the vertical, or both directions are to be used. This depends on the horizontal and vertical gradients, which are called the classifiers, and are defined by $\alpha = |(B42 + B46)/2 - B44|$ and $\beta = |(B24 + B64)/2 - B44|$. Then, G44 is given by Equation 3. For estimating G33, a similar concept is followed, but instead of using the blue samples, the red samples are used for the classifiers.

Once the luminance is determined, the chrominance values are interpolated from the differences between the color (red and blue) and the luminance (green) signal. This is given by:

$$R34 = \frac{(R33 - \mathbf{G33}) + (R35 - \mathbf{G35})}{2} + G34 \quad [5]$$

$$R43 = \frac{(R33 - \mathbf{G33}) + (R53 - \mathbf{G53})}{2} + G43 \quad [6]$$

$$R44 = \frac{(R33 - \mathbf{G33}) + (R35 - \mathbf{G35}) + (R53 - \mathbf{G53}) + (R55 - \mathbf{G55})}{4} + \mathbf{G44} \quad [7]$$

where the bold font indicates the interpolated value (5).

Interpolating color differences and adding the green component has the advantage of maintaining both the color information and the luminance information at the same time. Thus, this method makes use of the fact that the HVS is more sensitive to luminance changes. This gives an obvious advantage since it utilizes to some extent the relation between the different color planes.

Implementation wise, the first step is similar to the bilinear interpolation, however, in the second step special care should be given not to overflow or underflow the range, that is, have a value more than 255 or less than 0. The implementation uses floating-point arithmetic and limits the range to [0, 255].

Adaptive Color Plane Demosaicking (ACPD)

This method is proposed by Hamilton and Adams (10). The method is a modification of the gradient-based interpolation of method 4 and it also involves two steps. In the first step classifiers are modified to include the second order derivative of the chrominance as well as the first order derivative of the luminance, hence, the classifiers are defined by $\alpha = |B44 - B42 + B44 - B46| + |G43 - G45|$ and $\beta = |B44 - B24 + B44 - B64| + |G34 - G54|$.

These second order derivatives are then appropriately scaled and used in the interpolation of the luminance (green) color, as given by:

$$G44 = \begin{cases} \dfrac{G43+G45}{2} + \dfrac{B44-B42+B44-B46}{2} & \alpha < \beta \\ \dfrac{G34+G54}{2} + \dfrac{B44-B24+B44-B64}{2} & \alpha > \beta \; [8] \end{cases}$$

$$\frac{G34+G43+G45+G54}{4} + \frac{B44-B24+B44-B42+B44-B46+B44-B64}{8}$$

$\alpha = \beta$

For estimating G33, a similar concept can be followed, but instead of using the blue samples, the red samples are used.

Once the luminance channel is interpolated, the second step of interpolating the chrominance channels can start. This step also utilizes the second order derivative in interpolation. So, referring to Fig. (1.a), for the chromatic missing pixels B43 and B54 which has horizontally and vertically adjacent pixels, respectively, the interpolation can be found from

$$\mathbf{B43} = \frac{B42 + B44}{2} + \frac{G43 - \mathbf{G42} + G43 - \mathbf{G44}}{2}$$

[9]

$$\mathbf{B54} = \frac{B64 + B44}{2} + \frac{G54 - \mathbf{G64} + G54 - \mathbf{G44}}{2} \quad ...[10]$$

where the bold font indicates interpolated values. Estimating such red pixels follow the same concept. However, for pixel which has only diagonally adjacent pixels such as B53, a concept similar to estimating the luminance channel is employed, that is, by first calculating classifiers from $\alpha = |G53 - G44 + G53 - G62| + |B44 - B62|$ and $\beta = |G53 - G42 + G53 - G64| + |B42 - B64|$ and then based on these classifiers, B53 is found from:

$$\mathbf{B53} = \begin{cases} \dfrac{B44 + B62}{2} + \dfrac{\mathbf{G53} - \mathbf{G44} + \mathbf{G53} - \mathbf{G62}}{2} & \alpha < \beta \\ \dfrac{B42 + B64}{2} + \dfrac{\mathbf{G53} - \mathbf{G42} + \mathbf{G53} - \mathbf{G64}}{2} & \alpha > \beta \quad .....[11] \end{cases}$$

$$\frac{B42 + B44 + B62 + B64}{4} + \frac{G53 - G42 + G53 - G44 + G53 - G62 + G53 - G64}{4}$$

$\alpha = \beta$

where the bold font indicates interpolated values.

Theoretically, this method should give better interpolation compared to the earlier two since it employs both second order derivative and edge-directed interpolation and it better utilizes the relation between the different color channels.

Implementation wise, special care should be given not to overflow or underflow the range, that is, have a value more than 255 or less than 0.

The implementation uses floating-point arithmetic and limits the range to [0, 255].

Alternating Projections Demosaicking (APD)

This is the most complicated method discussed in this article and it is proposed in (11). This method tries to make use of the high correlation between the color channels in the high frequency band. Interested reader is advised to read (11) as many of the details are omitted here.

This method is also composed of two steps. In the first step, the luminance (green) channel is interpolated as follows:

i) Use either bilinear or edge-directed bilinear interpolation to get an initial estimate.
ii) Use the observed blue samples, such as *B44* in Fig. (1.a), to form a down-sampled version of the blue channel.
iii) Use the interpolated green samples, such as *G44* in Fig. (1.a), at the corresponding blue samples to form a down-sampled version of the green channel.
iv) Filter the down-sampled green channel to get the *LL* subband as shown in Fig. (2).
v) Filter the down-sampled blue channel to get the *LH, HL, HH* subbands as shown in Fig. (2).
vi) Reconstruct the down-sampled green channel using the *LL* subband from the green channel and the LH, HL, HH subbands from the blue channel as shown in Fig. (3).
vii) Insert these pixels in their corresponding locations in the initial green estimate.
viii) Repeat the same procedure for the green pixels at the red samples, such as *G33* in Fig. (1.a).

Note that *H0*(z) and *H1*(z) are the low-pass and high-pass analysis filters, respectively, and that *G0*(z) and *G1*(z) are the low-pass and high-pass synthesis filters, respectively. These filter together constitute a perfect reconstruction bank, which follows the condition:

$$H0(z)G0(z) + H1(z)G1(z) = 1 \quad ....[12]$$

Also, z1 and z2 are the z-transform along the horizontal and vertical directions, respectively.

The second step is an iterative process, which the papers proves to converge since it is defined over a convex set, involves reconstructing the chromatic channel (blue and red) using two steps: the details projection step and the observation projection step.

In the details projection step, the blue and green channels are decomposed into their subbands. Then, the *LL* subband of the blue channel is used together with the *LH, HL*, and *HH* of the green channel to reconstruct the blue channel. In the observation projection step, all the observed samples from the blue channel are inserted back into the blue channel. The same procedure is followed for the red channel.

Theoretically, this method should give the best results since it makes use of the correlation between the colors and iterates on a convex set to reduce the error.

Implementation wise, floating point arithmetic was used for the analysis and the synthesis steps besides the special care that was given so that no overflow or underflow would occur. The implementation limits the range to [0, 255]. Also, five iterations were used as recommended by the paper.

Results and Discussions

This method introduces the metrics used for estimating the quality of the demosaicking method, followed by experimental results, which are discussed in the final part of this method.

Quality Metrics

The mean squared error (MSE) is one of the commonly used performance measures in image and signal processing. For a black and white image of size N ×M it can be defined as:

$$MSE = \frac{1}{NM}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1}(x[n,m]-\hat{x}[n,m])^2 \ ...... \ [13]$$

Where x[n , m] is the original image and ˆx[n , m] is the reconstructed image. However for color images, the MSE can be calculated separately for each color component and the average MSE, also called the color mean squared error (CMSE) (3), is defined as:

$$Average\ MSE = \frac{MSER + MSEG + MSEB}{3} \ ....[14]$$

Where MSER, MSEG, and MSEB stands for the MSE for the red component, green component, and blue component, respectively.

Results

All the implementation details that were highlighted in the various demosaicking methods were implemented. For the alternating projections demosaiking (method 6), the filters in Fig. (2) and Fig. (3) were $h0$ = [1 2 1]/4, $h1$ = [1 −2 1]/4, $g0$ = [−1 2 6 2 −1]/8, and $g1$ = [1 2 −6 2 1]/8.

To test the various algorithms, a set of 4 images was selected, which is shown in Fig. (4). The Bayer CFA is simulated by sampling the images according to Fig. (1.a). Then, the images are demosaicked and compared with the original image using the earlier introduced metrics. An enlarged portion of these images in Fig. (4), after demosaicking are shown in Fig.(5)-(8) to provide visual comparisons for the various algorithms. Where as, a quantitative comparison of the various algorithms are given in Table 1.

Discussions

All the algorithms produce artifacts, which are not that obvious to the non-observant eyes. BD produces the most noticeable zipper effect as can be clearly seen in the baboon facial hair in Fig. (5.b) and on the fence in Fig. (8.b).

Obvious artifacts can also be seen in Fig. (5.b)- (5.f), (7.b)-(7.f), (8.b)-(8.g). Artifacts are not so obvious in Fig. (6.b)- (6.g) since the image is composed of relatively smooth with slowly varying regions.

The baboon Fig (4.a) is very challenging due to its rich color texture, which results in relatively poor performance for all the demosaicking algorithms as indicated by Table1. Also, the fence part of the lighthouse is not easy to interpolate since it is made up of high frequencies and suffers from color aliasing when sampled with Bayer CFA.

In general, it can be seen that the APD gives the best results visually and quantitatively, however, it is probably the least suitable for implementation on cameras due to high memory and computational powers required.

Among the algorithms, it can be noticed that both the GBD and the CHBD implement the same algorithm but the former implements it directly and the latter applies it in the logarithmic domain.

For implementation concerns, the reader is referred to the individual methods as the end of each explains the issues faced in implementing them.

From the complexity point of view, the algorithms are arranged in ascending order of complexity as follows: BD and EDBD is the simplest, GBD since it involves slightly more arithmetic, ACPD since there are even more additions and subtractions, CHBD since it involves divisions and multiplications, and finally APD since it involves plenty of arithmetic in filtering.

Conclusions

Bayer CFA is one of the fundamental steps in image generation of consumer digital cameras. Many methods have been proposed in literature and many more will be devised in the coming years. These methods vary in their motivation and performance.

Six methods have been studied and compared in this article, and their implementation details have been highlighted. Among them, the alternating projections demosaicking is probably the best method in terms of its performance, where as the bilinear demosaicking is the simplest. Some demosaiking algorithms produce color artifacts including zipper effects, which might be a deciding factor in ruling them out.

References

1. Adams, J.; Parulski, K.; Spaulding, K. (1998). IEEE Micro,. 18(6): 20-30.
2. Bayer, B. E. (1976), Color Imaging Array, U.S. Patent No. 3,971,065, Availableathttp:// research.microsoft.com/workshops/MMSP06/MMSP2006 _files/cameraReadyPapersNew/CameraReady_321.pdf
3. Alleysson, A.; S¨usstrunk, S., H´erault, J., (2005), IEEE Transactions on Image Processing, 44, Issue 4, pp. 439-449, April.
4. Ramanath, R.; Snyder, W. E.; Bilbro, G. L., Sander III, W. A. , (2002), Journal of Electronic Imaging,11,(3):306-315, July, Available at http://www.lira. dist.unige.it/teaching/SINA/papers/demosaicking - JEI-02.pdf.
5. Gunturk B. K.; Glotzbach J.; Altunbasak Y.; Schafer R. W., Mersereau R. M. , (2005). IEEE Signal Processing Magazine, 22 (1): 44-54, January,Available at http://www.accidentalmark.com/research/papers/Gunturk05Demosaic .pdf.
6. Taubman, D. (2000), Proceedings of the International Conference on Image Processing, 3, 801-804.
7. Wenmiao Lu, Yap-Peng Tan, (2003),IEEE Transactions on Image Processing, 12 (10): 1194-1210.
8. Cok, D. R. (1987).U.S. Patent No. 4:642,678, Available at http:// www.freepatentsonline.com/ 4642678.html
9. Laroche, C. A., Prescott, M. A. (1994), U.S. Patent No. 5:373,322.
10. Hamiltonl, J. F., Adams, J. E., (1997), U.S. Patent No. 5:629,734.
11. Gunturk, B.K.; Altunbasak, Y. Mersereau, R.M., (2002), IEEE Transactions on Image Processing, 11: (9), 997- 1013, September.

Table(1): Mean squared error comparison of different demosaicking methods

Image No.	Channel	Mean squared error for the different demosaicking methods					
		Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
1	Red	317.42	317.42	325.97	250.22	289.50	224.47
	Green	178.78	212.11	178.78	175.40	212.11	121.83
	Blue	383.47	383.47	306.82	277.89	323.29	242.76
	Average	293.22	304.33	270.52	234.51	274.97	196.35
2	Red	42.71	42.71	140.72	17.27	18.67	18.23
	Green	18.86	15.54	18.86	13.24	15.54	11.00
	Blue	46.32	46.32	82.97	23.62	25.36	24.06
	Average	35.96	34.86	80.85	18.04	19.86	17.77
3	Red	21.47	21.47	191.27	24.07	25.78	31.85
	Green	19.56	19.81	19.56	17.19	19.81	18.77
	Blue	55.71	55.71	114.17	40.27	44.09	37.71
	Average	32.25	32.33	108.33	27.17	29.89	29.44
4	Red	42.34	42.34	78.76	23.16	27.61	14.17
	Green	18.29	20.09	18.29	16.03	20.09	5.29
	Blue	39.62	39.62	24.33	18.11	23.04	8.01
	Average	33.41	34.02	40.46	19.10	23.58	9.16
5	Red	68.56	68.56	59.89	39.04	46.06	36.83
	Green	15.79	17.31	15.79	13.17	17.31	12.11
	Blue	51.45	51.45	39.23	25.32	31.41	23.41
	Average	45.27	45.78	38.30	25.84	31.59	24.11
6	Red	262.85	262.85	223.95	27.50	21.52	6.76
	Green	105.66	40.80	105.66	40.23	40.80	4.01
	Blue	297.23	297.23	188.14	29.94	21.08	6.92
	Average	221.91	200.29	172.58	32.56	27.80	5.90
7	Red	17.50	17.50	15.73	9.19	11.31	11.51
	Green	6.84	7.57	6.84	5.41	7.57	7.83
	Blue	28.80	28.80	26.38	16.83	18.14	19.99
	Average	17.71	17.96	16.32	10.48	12.34	13.11
8	Red	140.08	140.08	62.73	24.13	36.51	14.52
	Green	44.99	30.86	44.99	21.41	30.86	6.43
	Blue	135.52	135.52	80.43	23.65	36.13	12.48
	Average	106.86	102.15	62.72	23.06	34.50	11.14

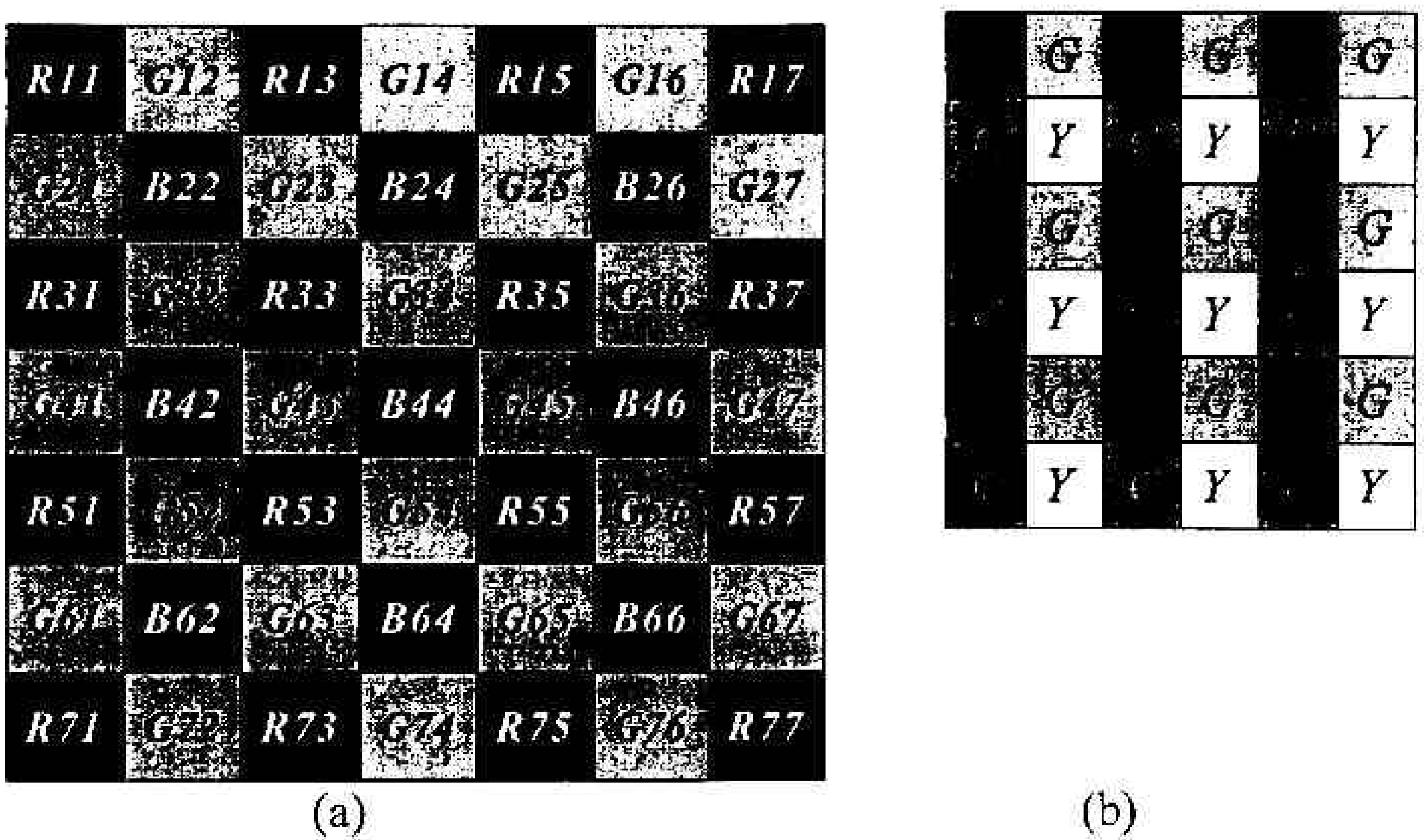

Fig. (1) A region of (a) Bayer color filter array (2), (b) Another popular color filter array(CMYG). R stands for red, G for green, B for blue, C for cyan, M for magenta, and Y for yellow (1).

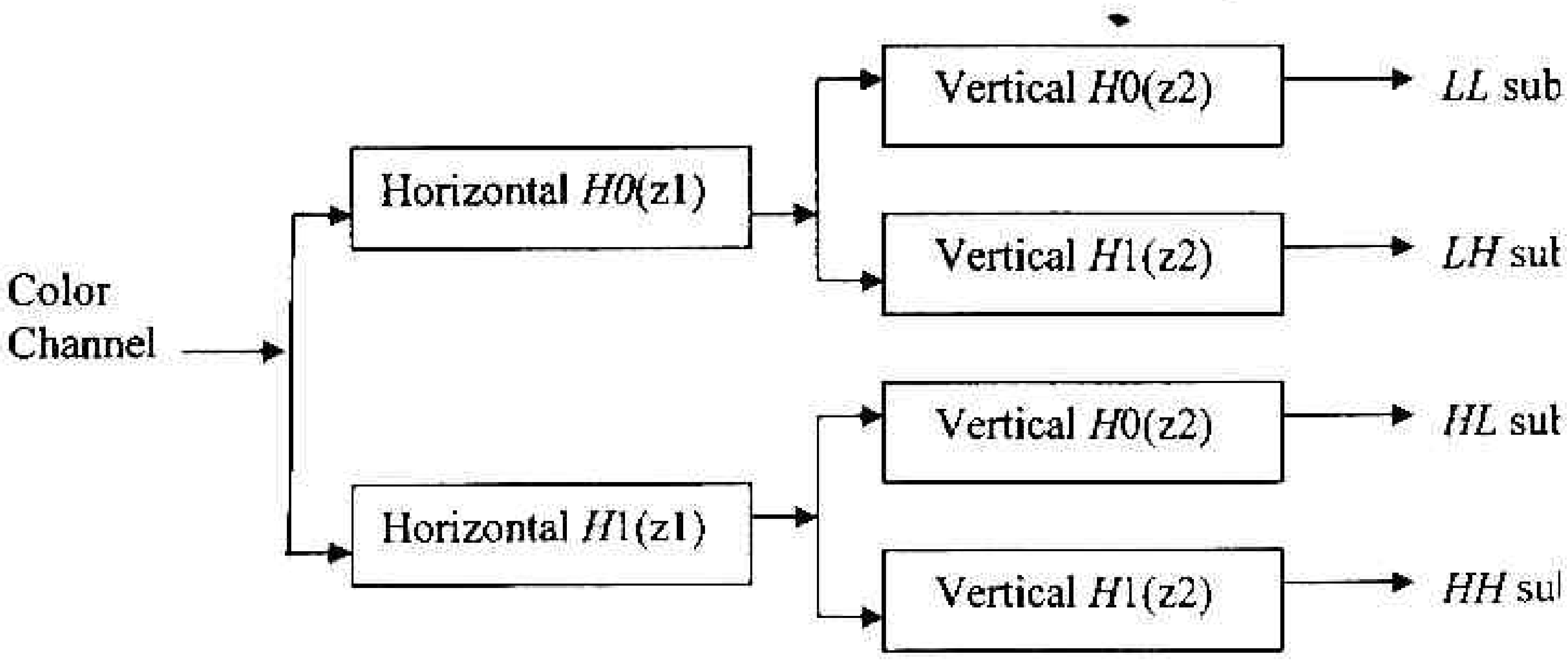

Fig. (2) Decomposing a channel into its subbands (also called analysis).

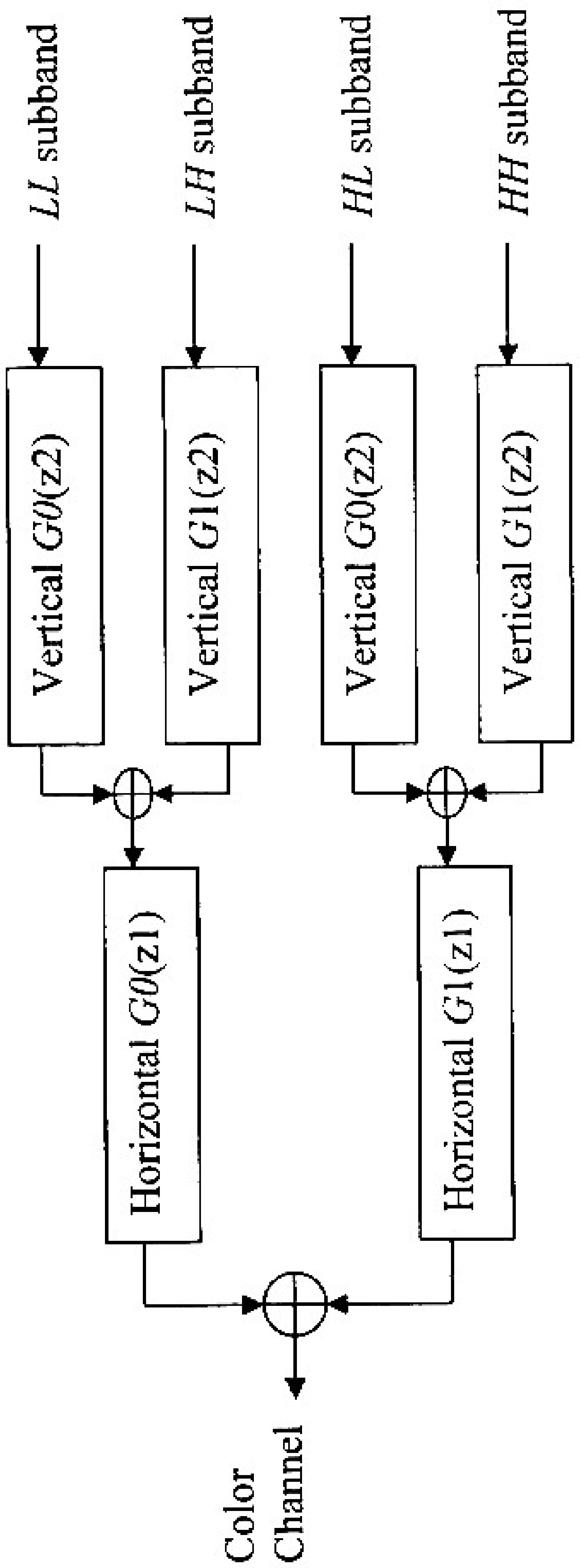

Fig. (3): Reconstruction of subbands (also called synthesis).

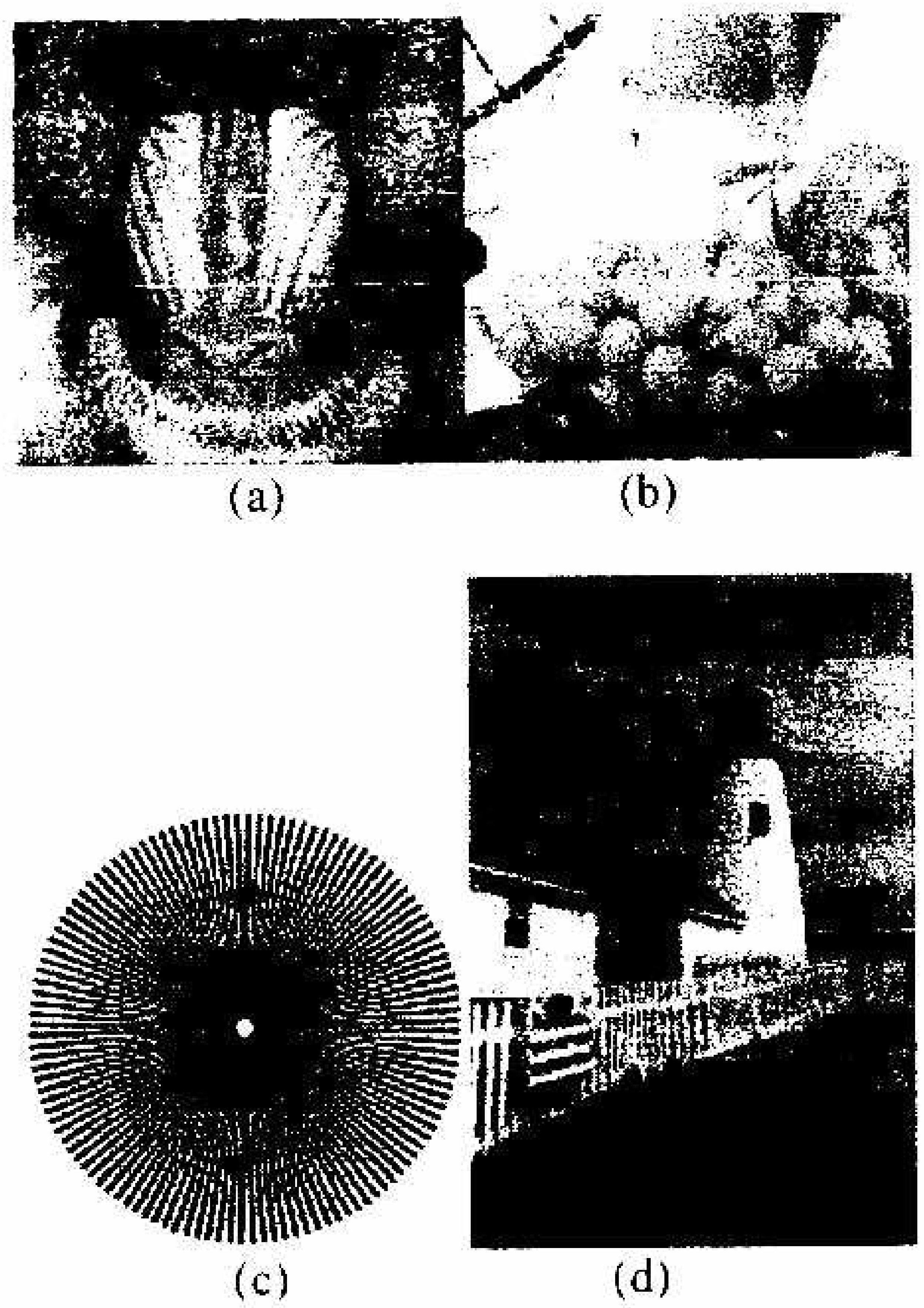

Fig. (4) Images used in testing. (These images are referred to as Image1 to Image8 in the paper, enumerated from left-to-right, and top-to-bottom)

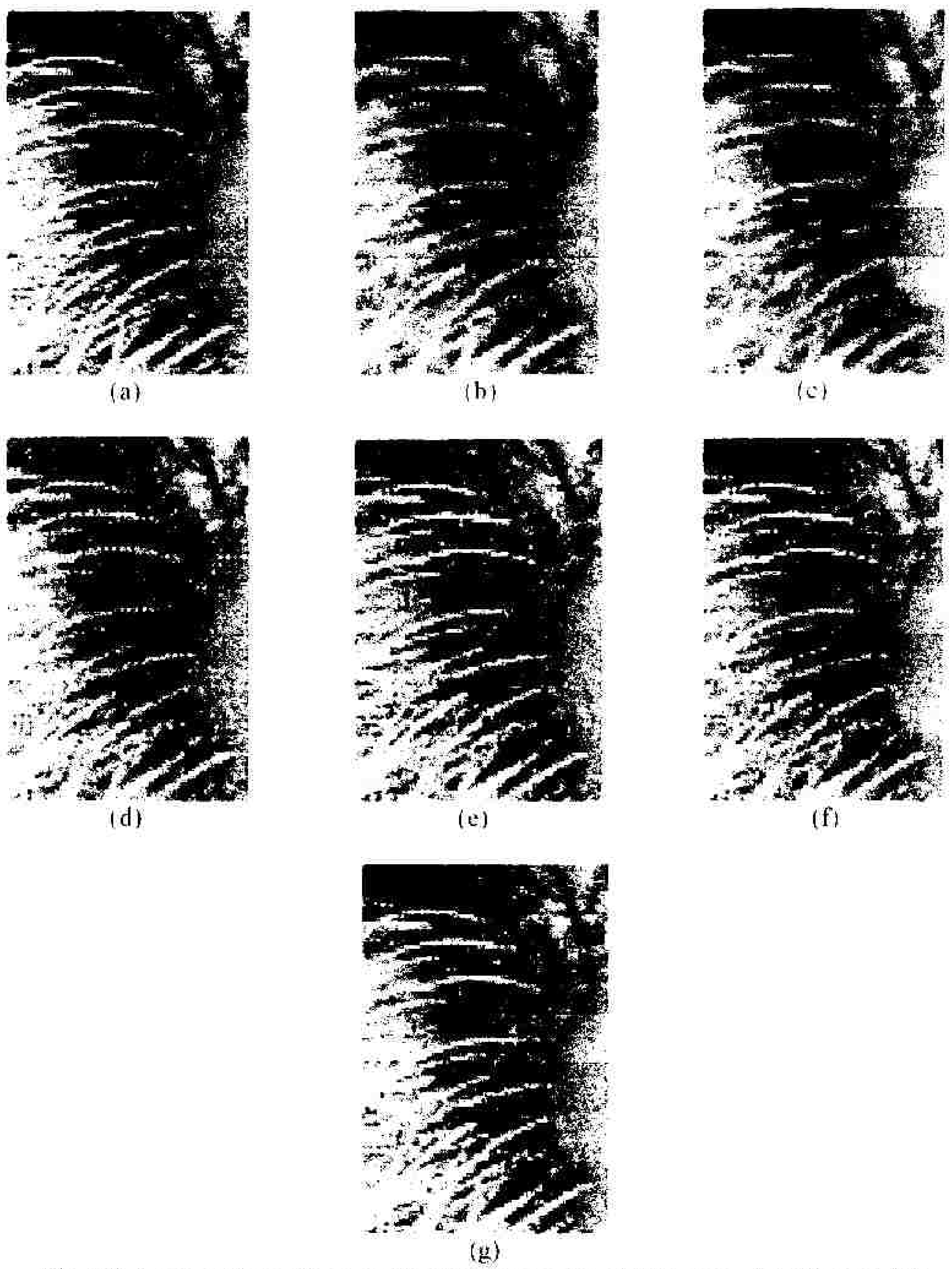

Fig. (5) A comparison of the methods visual quality for Image 1. (a) Original. (b) Method 1. (c) Method 2. (d) Method 3. (e)Method 4. (f) Method 5 (g) Method 6.

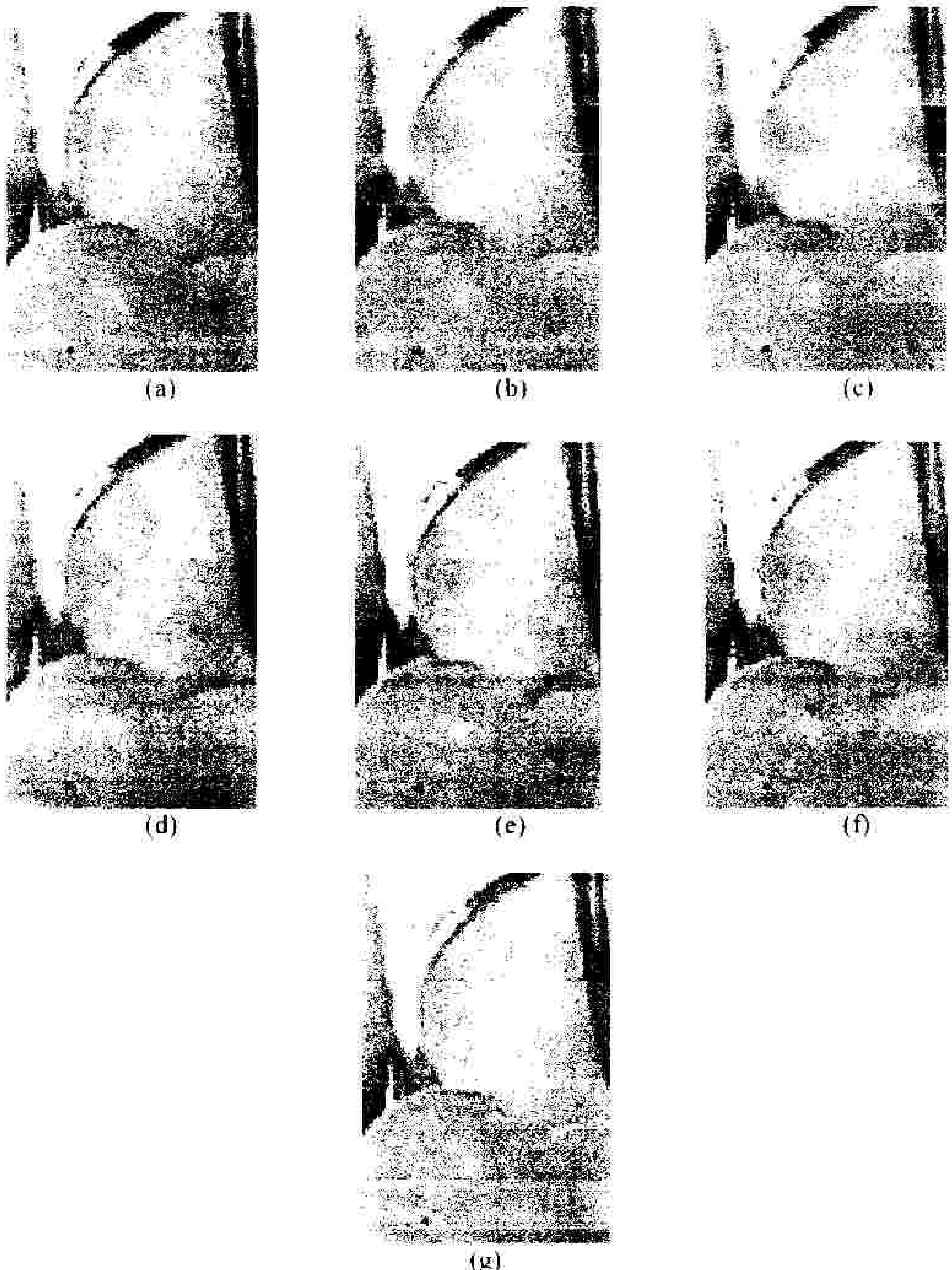

Fig. (6) A comparison of the methods visual quality for Image 2. (a) Original. (b) Method 1. (c) Method 2. (d) Method 3. (e)Method 4. (f) Method 5 (g) Method 6.

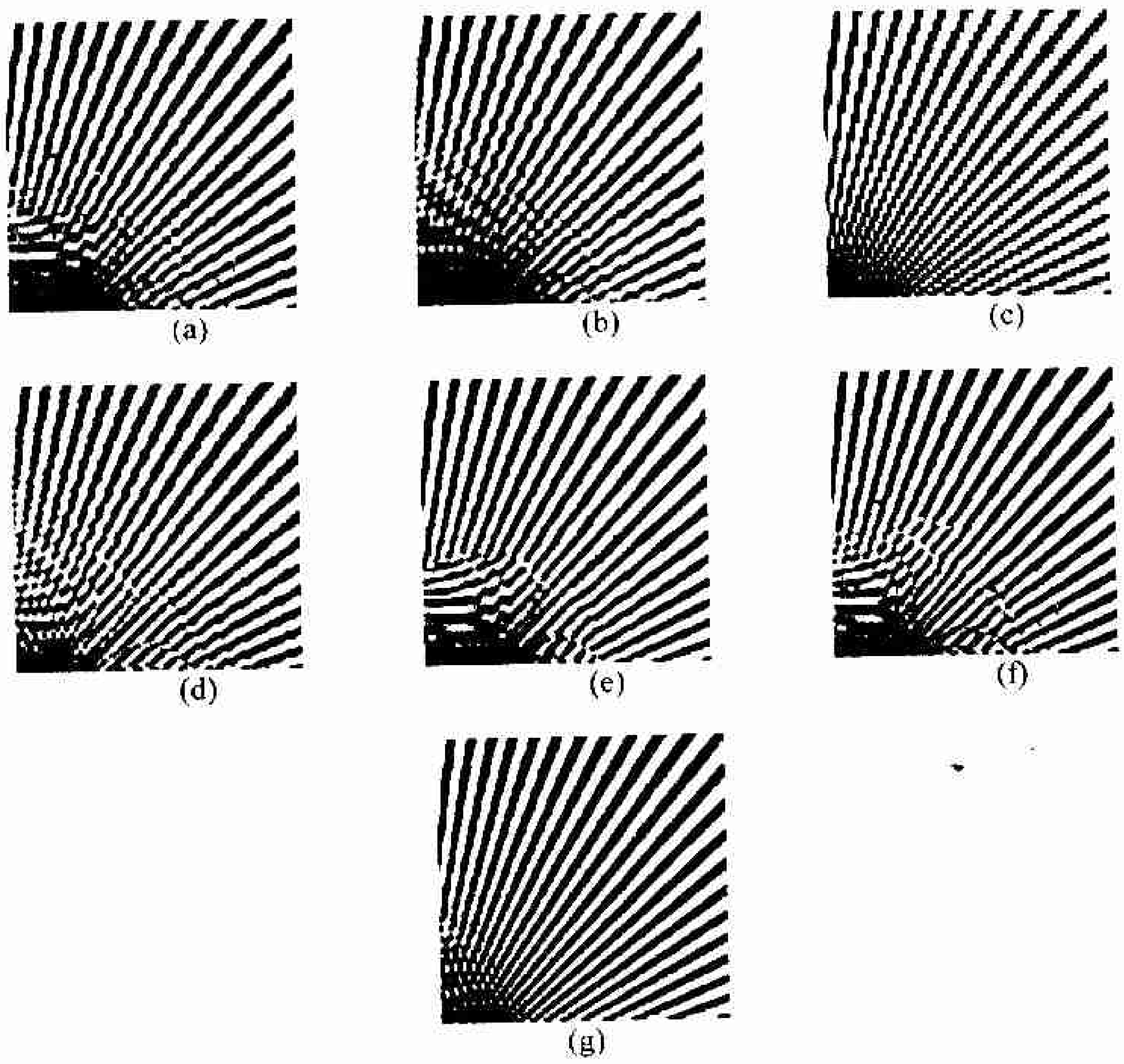

Fig. (7) A comparison of the methods visual quality for Image 6. (a) Original. (b) Method 1 (c) Method 2 (d) Method 3 (e)Method 4 (f) Method 5 (g) Method 6

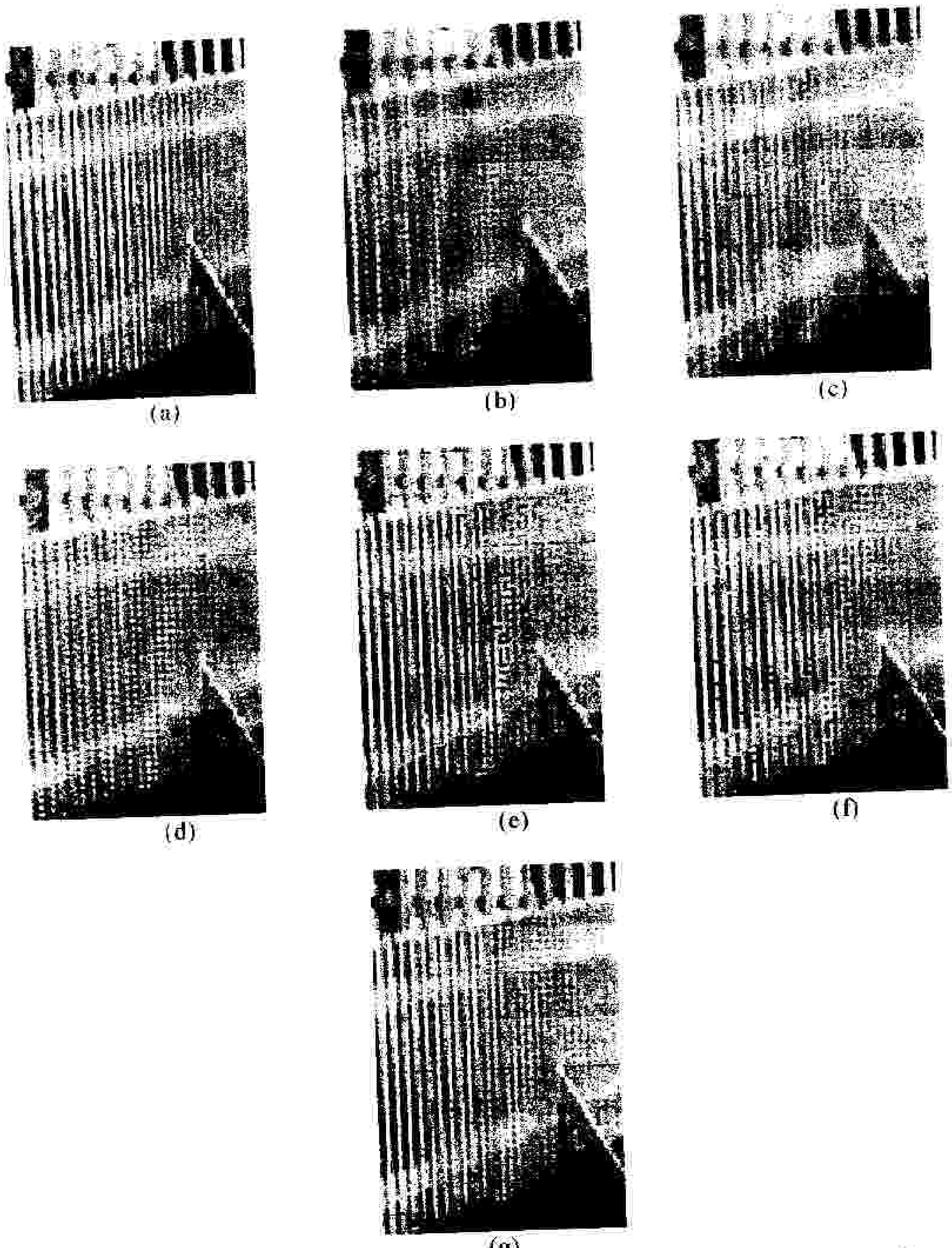

Fig. (8) A comparison of the methods visual quality for Image 8. (a) Original. (b) Method 1 (c) Method 2 (d) Method 3. (e)Method 4 (f) Method 5 (g) Method 6

دراسة المقارنه لأزالــة موزائـيـك مصفوفة مرشـح بايـــر

دوســـر ثابــت نــعــمان
قسم علوم الحاسبات ، كلية التربيه أبن الهيثم، جامعة بغداد

الخلاصه

ان عملية التقاط الصور في اغلب الكاميرات الرقمية للصور الـساكنة والمتحركـة تتضمن إستخدام جهاز شاحن مزدوج (charge coupled device) مغطى بمـصفوفة مـن قطعة واحدة من مرشحات الألوان بحيث ان كل بيكسل يأخذ قيمته من حزمة الـوان واحـدة. لذلك فأن التوليد الداخلي ضروري لاستعادة عينات الألوان المفقودة في هذا الناتج الموزائيكي وهذه العملية تدعى بإزالة الموزائيك (demosaicking).

هذا البحث يعطي نظرة شاملة عن إزالة الموزائيك، ويقدم عدداً من الطرائق المقترحة سابقاً، كما يحتوي على مقارنة نوعية لهذه الطرائق لمجموعة من الصور.