11 Color image processing: problems, progress, and perspectives

E. R. Davies and D. Charles

11.1. Introduction

Image processing has evolved over a period of some forty years, and it has now attained a fair measure of maturity. It has spawned a good many branches, such as image enhancement, image restoration, image matching, image compression, and image recognition, not to mention the closely allied subjects of image analysis, computer vision, and automated visual inspection—together with some apparently less related subjects such as virtual reality.

In many ways, image processing is a narrow subject, its basic definition being the study of how one image may be converted into another: traditionally, this means suppression of noise and artefacts, enhancement to improve clarity, or "deconvolution" to eliminate mathematically modeled distortions. Effectively, this definition means applying a transform to convert an image, pixel by pixel, into an improved image of the same sort (though here we should not exclude changing the resolution to match the immediate purpose for which the images will be used). This definition of image processing is motivated by the need to present human operators with improved images, which will be easier for them to examine: for example, doctors may require X-radiographs to be enhanced and to have lower levels of noise so that diagnosis will be made easier.

However, a separate motivation for image processing is to improve or adapt images so that the computer will itself be able to analyze them more easily and effectively. Image processing can then be considered as a preprocessing stage on the way to analysis and recognition. A further aspect is that image processing can be used to present a completely transformed image exhibiting edge signal instead of image intensity, or to present a corner- or other feature-enhanced image, which (to go one stage further) need only to be thresholded to indicate points that will be of immediate relevance for matching, recognition, 3D disparity measurement, and so on.

There are also higher level possibilities, such as transforming an image to a different viewpoint in 3D, taking several images, and reconstructing whole scenes,

or in some way, automatically removing undesired objects from photographs (a recent case in point is automatically removing the effects of blooming in digital cameras by deconvolution using inverses of rather complex blooming functions [1]).

In this chapter, we largely ignore this last set of complexities, and indeed all complexities arising from 3D aspects of the real world. Instead we concentrate on fairly general image processing aspects pertaining to preprocessing and the needs of subsequent image analysis; and we develop this topic to cover color processing as well as standard gray-scale processing. Color has recently been increasing in importance, with the advent of digital television, digitally generated films, and ubiquitous digital cameras—which have also become very much the sine qua non of the latest generation of mobile phones. In addition, we must not forget the huge increase in the use of surveillance—not just from earth-circling satellites, but from cameras above motorways and in urban environments including supermarkets, shops, and railway stations. Indeed, the digital camera and the webcam have fallen in cost so far that price has become almost negligible relative to the cost of computers and computer software, and, furthermore, it is actually becoming difficult to find cameras that present black and white rather than color images. Added to this, color is almost always preferred by humans (whether acting as amateurs or as professional operators), and it clearly gives far more information that may be crucial for matching and recognition tasks. Note also that the arrival of color in a convenient form and at competitive prices is very recent relative to the time span of the whole subject area, and to a certain extent we have been caught unprepared for its being provided so conveniently without behest.

It used to be possible to argue in many cases that color was a waste of resources. For example, in a simple inspection application, a gray-scale image could be thresholded to give a binary image that provided all the shape information needed to inspect metal flanges or other flat objects, and to check that all holes, corners, and other features were intact. That was a time when color seemed to add little, and indeed, one of the few inspection applications where color seemed to be crucial was in the inspection of fruit for ripeness and freedom from bruises [2].

On the other hand, we have now moved into the era of surveillance and crime prevention and detection, and it has become necessary to use full-color images to ascertain exactly what is going on in an outdoor environment: coincidentally, the September 11 scenario happened at a time when it was becoming increasingly possible to detect criminal and terrorist activity from surveillance cameras. This chapter will pay special attention to the color question, and will examine the extent to which color image processing has progressed in comparison with its gray-scale counterpart.

The aspects of color processing that will be covered in this chapter include how to manage the considerable additional information it brings; the concept of color bleeding—how color bleeding arises and how to cope with it; the contrast between linear and nonlinear filtering; the mode filter—its value and its implementation; the role of modern switched noise suppression filters; filters with adjustable parameters; image distortions produced by grayscale and color filters; and

a general review of recent color work. Further extensions and clarifications will appear in the immediately following chapter, which covers nonlinear edge detection in color images.

11.2. The color problem

Gray-scale images reflect a systematic change away from binary images, partly in respect of logic values giving way to intensity values, but also in respect of 1-bit arithmetic precision giving way to 8-bit (or even 12-bit) precision. In the latter case, we are merely moving along a scale of precision that can proceed as far as necessary, and the change is monotonic in nature. However, the change from gray-scale to color is distinctly different: three separate channels (or more in the case of multispectral, e.g., satellite, data) are involved, so the data characteristics have undergone a quantum leap rather than a monotonic change. Furthermore, it is not always obvious how to handle and combine the information from the different independent channels. If the images arose in a fixed application, a single channel could perhaps be produced by a simple linear weighting and addition operation applied to the three intensities, so, for example, apples could be graded on a single green-red scale. More complicated nonlinear functions could employ a lookup table to convert from color to gray-scale for the same basic purpose. However, the problem is that in a good many applications, the color is necessary, and the means of processing it to produce optimal meaningful results are likely to be complex and nontrivial. One standard approach [3] is to apply principal components analysis (PCA) to find the main information-carrying channels, and move from there to more systematic informed processing of the data. In fact, these aspects will not be broached in this chapter, as our aim here is to concentrate on the basic image processing: but the fact that there will be three independent channels rather than a single one will form a vibrant theme in the analysis.

11.3. Linear versus nonlinear processing

Image processing is commonly carried out in small neighborhoods or "windows," and in the past, 3×3 pixels was the most commonly used, though windows as large as 21×21 pixels are not uncommon now that computers have become more powerful. Linear processing is simple, useful, and straightforward to analyze and understand mathematically. When a linear operation is applied uniformly across the whole image, the mathematical process is actually a convolution—a very well understood process, and one that is exceptionally powerful. Convolution forms the basis of correlation-type template matching, and this provides maximum sensitivity for detection of features via the (spatial) matched filtering process, which came into prominence in the development of radar [4]. Thus it is no surprise that convolution forms the basis of most types of edge detector,[1] and has also been employed for corner detection, line segment detection, and sometimes detection of whole

[1]This also applies in the case of color; see the account given in the following chapter.

objects. The overall procedure is to use convolutions to produce an image whose pixels present the local signal for the feature in question, and then to apply nonmaximum suppression and/or thresholding—or more sophisticated methods—to perform the actual feature detection. For example, the Sobel edge detector and the Plessey corner detector apply convolutions that differentiate the image locally in the x and y directions, giving g_x and g_y differential gradient images, from which the following formulae lead to edge and corner signal values [5, 6]:

$$\begin{aligned} g &= \left(g_x^2 + g_y^2\right)^{1/2}, \\ c &= \frac{\langle g_x^2\rangle\langle g_y^2\rangle - \langle g_x g_y\rangle^2}{\langle g_x^2\rangle + \langle g_y^2\rangle}, \end{aligned} \tag{11.1}$$

the latter being deduced from the rotationally invariant determinant and trace of the locally defined matrix:

$$A = \begin{bmatrix} \langle g_x^2\rangle & \langle g_x g_y\rangle \\ \langle g_x g_y\rangle & \langle g_y^2\rangle \end{bmatrix}. \tag{11.2}$$

Several other corner detectors employ second-order local derivatives, g_{xx}, g_{xy}, g_{yy}—again derived by convolution masks—and these are applied within more complex formulae to estimate the local corner signal [5].

Lastly, we note that mean and Gaussian image smoothing filters also apply convolution operators, typified by the following 3×3 convolution masks:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \tag{11.3}$$

Overall, considering that, even in a 3×3 window, the general convolution operator has nine adjustable parameters, its high range of utility and power is undeniable. Nevertheless, its power is limited. First, two convolutions applied one after another remain a convolution, not a more powerful operator (though an $n \times n$ and an $m \times m$ convolutions applied in sequence will be equivalent to a larger $p \times p$ convolution, where $p = n + m - 1$), that is, it is still linear. Second, several of the linear processes mentioned above are normally followed by a highly nonlinear operation such as thresholding to actually detect the features. Third, the image smoothing operators are well known to have the disadvantage of smoothing the signal as well as any noise, so it is generally impossible to use them for this purpose. (However, Gaussian smoothing can be used for the special purpose of progressively cutting down image detail to create a hierarchical image structure or data pyramid, leading on to scale-space representations of images—though we will not pursue this possibility further here.)

These considerations show that in general we cannot do without nonlinear processing of the image. In fact, a more important question is not whether *linear*

processing is powerful enough to be used on its own but whether *nonlinear* processing is powerful enough to be applied in this way. However, even this question is misguided. Why constrain image processing in any such way? In fact, we should use a method that is appropriate for the particular type of data and the particular task to be carried out. Again, in image smoothing, it is well known that mean filtering is optimal for images that are corrupted by Gaussian noise, while median filtering is optimal for suppressing impulse noise. (Mathematically, median filtering is optimal for double exponential noise, which has wide wings matching well the incidence of outliers [5, 7].)

11.4. Color filtering

The remarks in the previous section were restricted to gray-scale processing. However, it is now necessary to enquire how color processing differs from gray-scale processing. We start by taking the well-known example of the median filter. In fact, this provides an immediate problem, as the median is only applicable to a single channel (such as gray-scale) whose entities are placed in numerical order and the median element in the ordered set is selected as the output value. In the multichannel case, there is no intrinsic ordering, and the median is hence undefined. Much work has been carried out to allow a suitable ordering to be applied in the multichannel (vectorial signal) case, but the problem has no universal solution, and again, any solutions that are found will tend to be of limited applicability: as remarked in the previous section (in the case of linear and nonlinear processing), the particular choice must depend on the data and the task.

In spite of these remarks, we can deal with the median filter problem by an alternative approach that does not require an ordering procedure [7]. We merely apply an alternative formulation—that the generalized median is the result of forming the following minimum sum of distances in color space:[2]

$$M = \min_i \sum_j |d_{ij}| \quad (i \neq j), \tag{11.4}$$

where

$$d_{ij} = \left[\sum_{k=1}^{3} (I_{i,k} - I_{j,k})^2 \right]^{1/2}, \tag{11.5}$$

$\mathbf{I}_i$, $\mathbf{I}_j$ being RGB vectors, and $I_{i,k}, I_{j,k}$ $(k = 1, \ldots, 3)$ being the color components. Thus the generalized median vector is the one that has the smallest total distance to all the other vectors. Clearly, an outlier would have a rather large total distance

[2] A more precise statement is that $\mathbf{I}_i = \arg(\min_i \sum_j |d_{ij}|)$, the value of i for which the minimum occurs being used to point back to the appropriate input vector. (It may be helpful to note that the mathematical expression $\arg\min_i$ is intended to be read "the sample vector giving the minimum value of ....")

to all the other vectors, and this total distance would become smaller as the chosen vector became more central: so this is an intuitively correct solution. More particularly, in one dimension, the formulation gives an identical result to that for the standard median: there will be equal numbers of points on either side of the chosen value. The only problem in the 1D case arises when there is an even number of points. This particular formulation asserts that the proper answer is the midpoint between the two central points, whereas that was only a convenient convention in the normal median case (it was inconvenient to say that there were two medians or alternatively none, so workers chose to say that the true median was the midpoint: but the new formulation forces us to make the median the midpoint).

Finally, the generalized median is one of the original vectors, and this corresponds to the median being one of the original data points in the 1D case. From now on, we follow normal practice in calling the generalized median, defined in this way, the vector median (though we will sometimes revert to calling it the median when it is understood that we are dealing with a particular 3D manifold).

Before proceeding further, it is worth noting that the generalized mean in color space is readily defined using the same color space metric (e.g., the Euclidean distance) as for the median. There is no necessity for the result to be one of the initial data points, and indeed very little likelihood that the mean would be the same as one of the initial points, either in 1D or in the multichannel case.

So far, there has been no indication whether or not it is desirable for the output signal to be identical, in any dimensionality, to one of the input signals. It seemed that in the median case, it had to be identical, whereas in the mean case it was most unlikely to be identical. Yet in the latter case, restricting the result to one of the input values would significantly limit the accuracy of the final result. Indeed, Davies [8] has shown that this situation also applies for the single channel median filter. We will consider the situation in more detail below.

11.5. Color bleeding

As has already been remarked, one of the advantages of median filtering is its capability for eliminating outliers. However, when an outlier occurs near an edge, the effect can be to pull the edge closer to the outlier as well as eliminating the outlier. This is seen from the 1D examples shown in Figure 11.1, for all of which a 3-element median has been applied to the data.

This behavior could be serious in the case of multichannel data. For example, in a 3-color RGB image, if we applied the median separately to the three channels and an outlier occurred in just one of them, the result would be that the outlier would be eliminated, but the edge would be shifted in one channel, and this would result in an erroneous change of color at that location [7] (Figure 11.2). The effect might sometimes be more serious than the improvement resulting from elimination of the outlier—though "seriousness" may be at least partly a human subjective response governed by the prominence of any new color that is introduced.

There is also the opportunity for erroneous colors to arise in other situations. While edges tend to demarcate pairs of regions in an image, there will be places

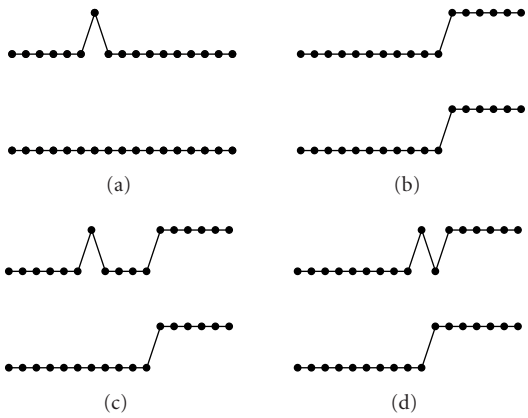


FIGURE 11.1. Application of a 1D median filter. Here a 3-element median filter is applied to four 1D signals: (a) an isolated noise point, (b) a single edge, (c) a noise point far from an edge, (d) a noise point near an edge. In (d), notice how the edge has been shifted left by one pixel.

FIGURE 11.2. Effect of scalar component-wise filtering. This figure shows the effect of mode filtering on a brightly colored image with the filter applied to each color channel independently. Color bleeding is most noticeable as isolated pink pixels and a number of green pixels around the yellow sweets. (Reproduced from D. Charles and E. R. Davies, "Mode filters and their effectiveness for processing color images," Imaging Science Journal © 2004 RPS.)

where three regions come together, and at these points there will often be three totally different colors, each of which can be regarded as an outlier to the other two.[3] Depending on the particular geometry existing at the joining regions, and the local variabilities of the channel intensities, substantial "bleeding" of one region into another can occur. The problem is exacerbated by the numbers of different colors that can result when different intensities of each color channel are brought in.

[3]It is important to remember that outliers may arise not only as a result of noise but also in the form of background clutter.

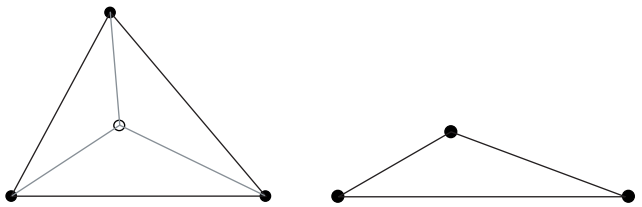


FIGURE 11.3. Geometry of three color data points in 2D. In (a), the minimum total distance point (MTDP) lies near the centroid of the triangle. In (b), the MTDP is the uppermost data point.

The value of the vector median approach now becomes particularly apparent: the output pixel vector is severely restricted by the fact that it must be the same as that of a particular pixel in the window. Thus no new color combination is generated, and the output pixel vector will in the case of a 3×3 window be identical to that of one of the nine pixels [7]. In larger windows, the restriction will not be so marked. However, in a $p \times p$ window, the maximum number of available colors will be p^2, compared with $(p^2)^3 = p^6$ when the input to any color channel may be combined with the input of any other two color channels in 3D color space.

It should next be questioned whether any disadvantage can result from application of the vector median procedure. Here we consider just one problem. We illustrate the problem by reference to a 2D color space containing three data points (the case of two data points is essentially a 1D problem). Let us find the minimum total distance point (MTDP) given by (11.4) for several sets of 2D vector data. Figure 11.3 suggests that the MTDP will, in most cases, occur in the space between the data points. Exceptions arise when the points are arranged in a rather squat triangle. In fact, for nonsquat triangles, the MTDP will lie near the centroid of the space, where the lines joining the MTDP to the three data points are at angles of 120° to each other. (An intuitive proof of this result involves first considering the points at the corners of an equilateral triangle, and then noticing that moving them, arbitrary distances from the MTDP in the same directions would not produce any movement of the MTDP.)

For four data points in 2D, there are again two possibilities [9]: one is that the MTDP is at the space near the centroid of the points; the other is that the MTDP is situated at one of the points. In the first of these cases, the actual MTDP is on the crossing of the diagonals of the quadrilateral formed from the four data points (Figure 11.4): to prove this, we merely note that moving in any direction from the crossing point changes one or other diagonal, or both, to two sides of a triangle, which must have greater total length. The second case corresponds to a concave quadrilateral (i.e., one point lies inside the triangle formed from the other three points): to prove that the central point C has smaller total distance to the other points, first move to a point T a short distance in the direction of the point P (Figure 11.5); this will not change the sum of the distances to C and to P, but will increase the distances to the other two points Q and R; the MTDP is at C.

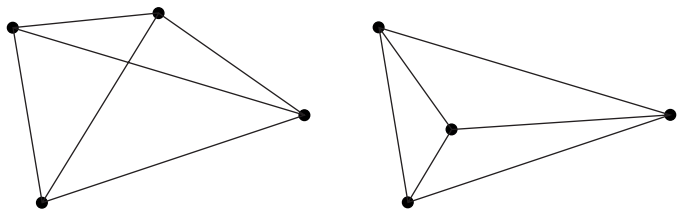


FIGURE 11.4. Geometry of four color data points in 2D. Case where (a) none and (b) one of the points lie within the triangle formed from the other three points. In (a), the MTDP is at the crossing of the diagonals, whereas in (b), it is at the central data point.

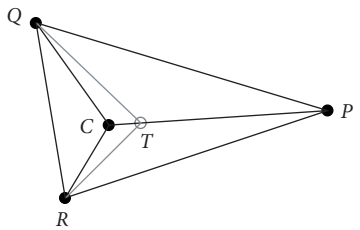


FIGURE 11.5. Geometry for proving the position of the MTDP for four color data points in 2D. In this case, the MTDP for data points P, Q, R, C cannot be at T, because this would extend the distances from Q and R, but not the sum of the distances from P and C.

Coupled with the 1D case, these two examples show that there is something like 50% probability that the MTDP will lie in free space, and not at one of the initial data points—and this situation does not seem likely to be different in a 3D color space, or in higher dimensional spaces [9]. Thus the basic mathematical equivalent of the median ordering formulation, (11.4), would appear to state that for minimum error, we must always use the MTDP: it does not insist that the median be taken to be the data point that is closest to the MTDP. Contrariwise, restricting the output multichannel median to one of the input vector samples seems likely to introduce substantial error.

Understanding the situation is not trivial. One view is that if a purely numeric solution is needed, the MTDP must give the correct solution. On the other hand, if a true color interpretation or color recognition process is to ensue, then the vector median solution must be used, as misinterpretation of the color would constitute a major error. Another factor is that if the noise is Gaussian, the MTDP will be better (in the sense of being numerically more accurate), whereas if the noise is impulse noise, the vector median will be better (in the sense of not being open to misinterpretation). In any case, it should always be borne in mind that using the vector median procedure may not represent the most accurate solution, and that the decision about whether to use it must be task dependent.

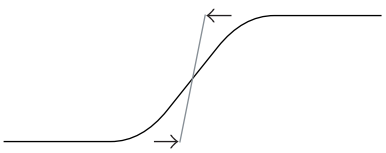


FIGURE 11.6. Action of the mode filter. Here the mode filter is applied to a 1D edge intensity profile. The filter outputs the majority value of the intensities within each window and thus pushes this intensity towards the edge boundary from each side. The result is a profile more like that indicated by the gray line.

11.6. The mode filter

As indicated above, median filters are widely used for suppressing noise in images, and are particularly important in view of the fact that they do not at the same time blur the image signals. In fact, there are many other types of image filter. One other type of filter based on a commonly known statistic is the mode filter. This takes the main mode of the local intensity distribution (in the case of gray-scale images) and uses it to provide the output value for each pixel. There is much sense in using this particular statistic, as its value ought to represent the most probable intensity in the local intensity distribution. However, the result is not exactly what might a priori have been expected—for the mode filter *enhances* the image rather than merely eliminating noise (though it does achieve a considerable degree of noise reduction, as the mode is generally well away from the extremes of the distribution and thus it is good at eliminating outliers). The enhancement property is achieved since the filter ignores minor modes and therefore gives a majority view in the vicinity of edge points: just inside an edge it gives the inside view, ignoring the outside view, and just outside an edge it gives the outside view, ignoring the inside view. The result is that it pushes the local majority intensity inwards from both sides of the edge—thereby giving a much more crisp boundary to the edge [10] (Figure 11.6).

Unfortunately, there are distinct problems in estimating the mode, not merely because of noise, but also because the local intensity distribution is sparse; as a result, while the highest point is technically the mode, it is not necessarily the underlying mode, which is the one that needs to be used as the filter output. To overcome this problem, Algorithm 11.1 was devised (see also Figure 11.7) [10].

The rationale of this algorithm is the following: if we knew the position of the underlying mode U, and located the nearer end E of the distribution, and then moved from E to U and an equal distance further on, this would be a reasonable position to truncate the distribution. Next, noting that the median of the distribution is normally on the far side of the underlying mode relative to E, using the median as an estimator of U is safe in the sense that it will never truncate more than the ideal proportion of distribution (Figure 11.7). While iterating the procedure can in principle give a better estimate of U, it has been found that this is not necessary in practice. Indeed, the mode filter has been found to produce very good

(1) Find the median M of the distribution.
(2) Find the end E of the distribution that is nearer to M.
(3) Measure the distance d from E to M.
(4) Move a further distance d from M, to point T.
(5) Truncate the distribution from T to the other end F of the distribution.
(6) Find the median U of the remaining portion of the distribution.
(7) Use the median value U as an estimator of the underlying mode.

Algorithm 11.1. Mode filter algorithm for gray-scale images.

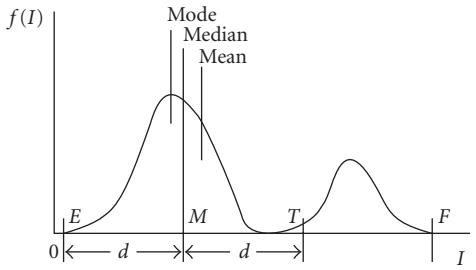


Figure 11.7. Method of truncating the local distribution. The diagram shows the ordering of the three means for a bimodal distribution. It also shows how the intensity histogram is truncated at a specified distance from the median. The rationale for this procedure is indicated by the idealized situation that would exist if the position of the mode was known initially. (Reproduced from D. Charles and E. R. Davies, "Mode filters and their effectiveness for processing color images," Imaging Science Journal © 2004 RPS.)

results with a minimum of algorithmic complexity and to be reasonably resistant to the effects of sparsity of the local intensity distribution [10]. One final aspect that is of interest here: in stage 7 of the algorithm, the median can be replaced by the mean with little effect on performance, and without introducing blurring. This is because blurring is normally introduced by a mean filter as a result of mixing together the intensities from parts of the window inside and outside the edge, but here this is prevented as the smaller of these parts has been eliminated by truncating the distribution.

11.6.1. The situation in color images

When applying the mode filter to color images, all the basic components of the algorithm can be replicated without problem. Specifically, the median can be replaced by the vector median, and the truncation can be carried out without difficulty. Nevertheless, determining exactly where to perform the truncation involves slight difficulties and the gray-scale algorithm has to be adapted carefully.

The first requirement is to find the vector median of the sample points in the 3D color space. Then we need to consider where the closest part on the boundary of the distribution might be. However, this is difficult to do robustly, as it is necessary to distinguish the closest points on the boundary from those internal to

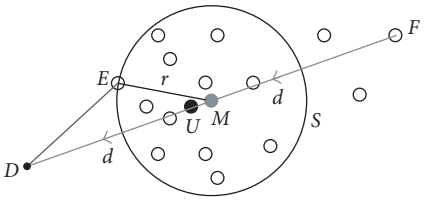


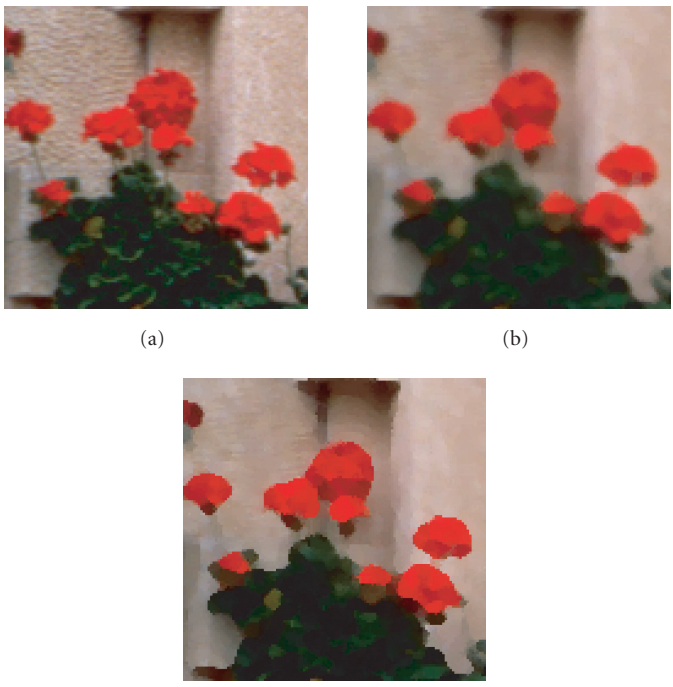
FIGURE 11.8. Method of truncating the local distribution in 2D, extendible to 3D color spaces. The circles are samples, the shaded circle is the vector median M of the input distribution, and the black circle U is the vector median of the truncated distribution. The dot D is not a sample: it is the location in color space found by moving from sample F to sample M and then the same vector distance **d** further on. (Reproduced from D. Charles and E. R. Davies, "Properties of the mode filter when applied to color images," IEE Conference Publication, © 2003 IEE.)

(1) Find the vector median M of the distribution.
(2) Find the end F of the distribution that is furthest from M.
(3) Find the position D that is (relative to M) diametrically opposite the end F.
(4) Find the point E that is closest to D.
(5) Construct a sphere S centered at M and passing through E.
(6) Truncate the parts of the distribution lying outside S.
(7) Find the vector median U of the remaining portion of the distribution.
(8) Use the vector median U as an estimator of the underlying mode.

ALGORITHM 11.2. Mode filter algorithm for color images.

the distribution. In fact, it is easiest to start by determining the *furthest* part of the boundary, that is, the position and direction of the furthest outlier F. Once these have been located, we move to the median and then an equal vector distance **d** further on, to D; next, we find the sample that is closest to D [11, 12] (Figure 11.8). The resulting algorithm is shown in Algorithm 11.2: it is essentially the same as in the gray-scale case. However, it is important to note that stage 8 now has to employ the vector median rather than the mean or the MTDP, as the final position selected must be one of the original data points, so that color bleeding will be suppressed.

An interesting aspect of mode filters in color images is that in addition to their enhancement properties (Figure 11.9), they turn out to have remarkable capabilities for eliminating large amounts of impulse noise [12]. In fact, up to 70% of randomly corrupted pixels containing random colors (each color channel having 70% of the intensities uniformly distributed over the entire range 0 to 255) were largely eliminated with very little distortion of the final images (Figure 11.10). This capability for elimination of high levels of noise was not matched by the median filter in spite of the fact that the median filter ought to be more resistant to outliers, as theory and earlier work on gray-scale images had suggested. It is likely that this degree of improvement resulted from using larger windows than used to be customary with these relatively computation intensive filters, thereby allowing the



(a) (b)

(c)

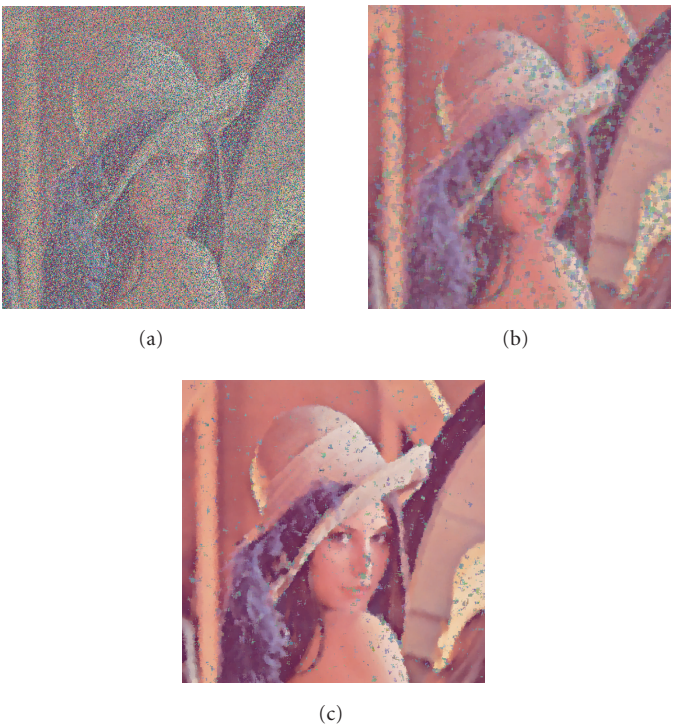
FIGURE 11.9. Effect of the mode filter on a color image. (a) Original image. (b) Effect of vector median filtering. (c) Effect of vector mode filtering. Both of the filters are applied in a 5×5 window. (Reproduced from D. Charles and E. R. Davies, "Mode filters and their effectiveness for processing color images," Imaging Science Journal © 2004 RPS.)

mode filter to lock on to and make full use of the remaining 30% of unaffected pixels. Another factor is the considerable discriminating power available when color is taken into account.

11.7. Modern "switched" noise suppression filters

In the past few years, several filters that are substantially better than the median at eliminating noise have been devised (see, e.g., [13–16]), and some comment on the mode filter's noise performance vis-à-vis these new filters is appropriate.

Typically, these new filters obtain their improvements by analyzing the image contents more carefully than the median does, and acting accordingly. Rather than merely sorting the pixel intensity values and taking no account of where in the window each intensity originated, the new filters adopt cleverer strategies. In particular, they have several means of calculating the output intensity, and select amongst (or "switch" between) these values according to deductions made about the situation at and around each pixel. For instance, they retain the option of using the output of a standard median filter or leaving the pixel intensity unchanged



(a) (b)

(c)

FIGURE 11.10. Filtering in the presence of large amounts of impulse noise. (a) Lena image with 70% random impulse noise. (b) Effect of vector median filtering. (c) Effect of vector mode filtering. Both of the filters are applied in a 9×9 window. (Reproduced from D. Charles and E. R. Davies, "Mode filters and their effectiveness for processing color images," Imaging Science Journal © 2004 RPS.)

if it is judged not to be modified by impulse noise. In the case of the Eng and Ma [13] noise adaptive soft-switching median (NASM) and "ideal" filters,[4] selection of output values is also made on the basis of any local edges that are detected in the image and whether more than one noise impulse is found at or near a given pixel. These particular filters also consider the optimum size of the filter window, larger windows being selected when the noise density is greater.

Tests made by Charles and Davies [12] confirmed that the two filters proposed by Eng and Ma were very effective at suppressing high levels of impulse noise. These tests employed gray-scale images, as the Eng and Ma filters only provided this capability. In comparison with the median and mode filters, the peak signal-to-noise ratios achieved for an impulse noise density of 70% were

(i) median 17 dB,
(ii) mode 24 dB,
(iii) NASM 28 dB,
(iv) ideal 32 dB.

[4]The "ideal" filter is a simulated idealized filter in which the positions of the noise impulses are taken to be known, and to this extent it does not provide a realistic comparison.

Clearly, then, the mode filter is not as effective as the NASM filter at suppressing high densities of impulse noise, though in this respect it is much better than the median filter. What is interesting is how well the mode filter performs despite its simplicity (the C++ code is only about 30 lines long) and lack of specific mechanisms for adapting to the local conditions.

As remarked above, the two switched types of filter outlined above were applied only to gray-scale images, and it is of interest here whether they could be applied successfully to color images. The critical aspects of these filters include the use of a median filter and switching its output on and off depending on local conditions in the image. None of these aspects precludes their use with color images, the only proviso being that a vector median filter needs to be used in place of a median filter. Further developments in this area are therefore to be expected in the near future.

11.8. Filters with adjustable parameters

Many image filters such as the median filter have no adjustable parameters other than the crude one of deciding the size of window in which they are to be applied. Hence there is virtually no scope for optimization to the conditions occurring in different images—such as varying levels of noise. Thus there has been a trend towards the inclusion of adjustable parameters within filter algorithms. As we will see below, these parameters often take the form of weights.

The concept of weighted median filters is now well established in the literature, and a number of workers have given much attention to the development of weighted filters for gray and color image noise removal and enhancement (Sun and Neuvo [17]; Fotopoulos et al. [18]; Sangwine and Horne [19]; Marshall [20]). In one definition (Sun and Neuvo [17]), the weighting is achieved by duplication of samples inside the filter, such that with input sample set $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and corresponding weights $\mathbf{w} = [w_1, w_2, \ldots, w_n]$, the output y of the filter is

$$y = \text{median}[w_1 \diamondsuit x_1, w_2 \diamondsuit x_2, \ldots, w_n \diamondsuit x_n], \tag{11.6}$$

where $\diamondsuit$ denotes weighting by using each x_i data point w_i times. In this case, the w_i are necessarily integral. In contrast, the minimum weighted distance function defined by Fotopoulos et al. [18] (leading to the result $\mathbf{x}_i$) is

$$M = \min_i w_i \sum_j |d_{ij}| \quad (i \neq j), \tag{11.7}$$

where the weights w_i need not be integral.

In an alternative approach, Charles and Davies [21] introduced a filter that incorporates a window distance weighting function. In particular, (11.4) is modified by introducing a weighting function $f(r_i)$ dependent on the Euclidean distance r_i

between pixel i and the center of the window:

$$M = \min_i f(r_i) \sum_j |d_{ij}| \quad (i \neq j), \tag{11.8}$$

where

$$r_i = \left[(x_i - x_0)^2 + (y_i - y_0)^2 \right]^{1/2}. \tag{11.9}$$

Here x_0 and y_0 represent the location of the center pixel and $f(r)$ is the weighting function chosen according to the application.

The weighting functions $f(r)$ are necessarily increasing functions which are largest at large distances. This is useful for continuity reasons, so that the limitations of the window support region do not have a too great effect on the properties of the operation: indeed, if the function value went to infinity at the boundary of the support region, increasing the size of the window would give no observable effect. (It is of course necessary to remember that the most relevant function values are those for small r, as (11.8) takes the minimum value as the output.)

The value of this approach is that one or two crucial carefully selected parameters can be employed instead of a plethora of arbitrary weights. In fact, in [21], only a single parameter α reflecting the radial extent of the function was used:

$$f(r) = 1 + \frac{r}{\alpha}. \tag{11.10}$$

Clearly, the weighting function and the amount of weighting applied will vary with the type of application and desired results, but even in its basic form, this type of weighted median filter moves continuously from the detail-preserving properties of a small operator to the greater noise-averaging properties of a large operator by adjustment of the single parameter α.

Figure 11.11 shows the shape of the normalized mean-square error (NMSE) optimization curve when applying the distance-weighted filter for a range of parameter values. (In such cases, tests are made by adding known amounts of noise and then measuring how much of the noise is eliminated by the algorithm.) Note that the horizontal asymptote shows the situation when a standard vector median filter is employed—that is, this is a limiting, suboptimal case of the distance-weighted filter.

11.9. Distortions produced by median and other filters

The previous sections of this chapter have discussed image processing and filtering, and have gone to some trouble to consider how color modifies the algorithms and to assess their effectiveness. At this point it is useful to consider the extent to which they may produce distortions in the images as a by-product of their desired action, whether this be noise elimination or some other effect such as enhancement.

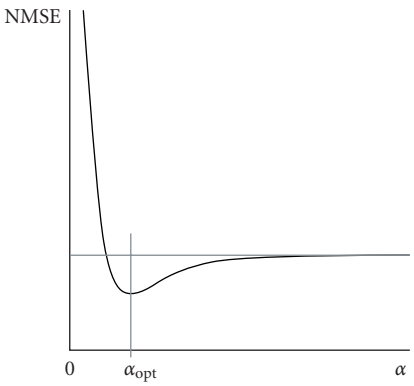


FIGURE 11.11. Shape of the normalized mean-square error (NMSE) optimization curve when applying the distance weighted filter for a range of parameter values. The horizontal gray line shows the NMSE for the standard vector median filtered image.

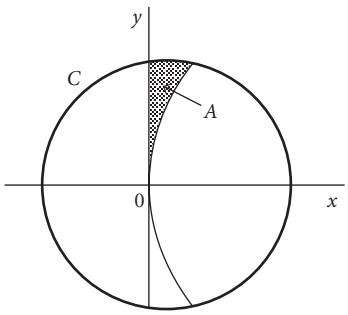


FIGURE 11.12. Geometry for calculation of contour shifts using the median filter. The area *A* is shown shaded. The origin on the median contour (which bisects the area of the circular window *C*) is not at the center of *C*. (Reproduced from E. R. Davies, "Image distortions produced by mean, median, and mode filters," IEE Proceedings-Vision Image and Signal Processing, © 1999 IEE.)

In fact, median and other filters produce distinct edge shifts, but only for curved edges. For straight edges, symmetry dictates that there will be no shifts (any straight edge will be shifted laterally by the same amount in both directions). To discuss the situation for curved edges, consider first the case of binary images—or, equivalently, gray-scale images containing step edges. Here the median contour in a window will be the one with equal areas (equal numbers of pixels) on either side of the contour. The same applies when mean or mode images are being produced. Interestingly, the same result applies again in the case of the median filter when the edge profiles present linear rather than step variations. We will return later to certain other possibilities.

Fortunately, it is quite simple to see what happens when a circular boundary divides the region in a circular window into equal areas [22]. Examining Figure 11.12, we can see that the circular boundary cannot pass through the center of the window, so when its intensity is placed at the output of the filter, it will appear at

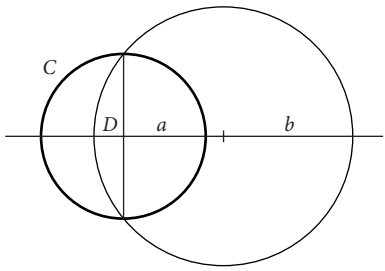


FIGURE 11.13. Geometry for the calculation of contour shifts using the mode filter. (Reproduced from E. R. Davies, "Image distortions produced by mean, median, and mode filters," IEE Proceedings-Vision Image and Signal Processing, © 1999 IEE.)

the center position in the output image—that is, there will be a distinct shift inwards. We can find an approximate value for the shift by taking a circular contour, centered at $(b, 0)$ and passing through the origin (Figure 11.12)

$$(x - b)^2 + y^2 = b^2. \tag{11.11}$$

After simplifying and approximating to large b, we find

$$x \approx \frac{y^2}{2b}. \tag{11.12}$$

The part of the area in the positive quadrant lying to the left of the contour is

$$A \approx \int_0^a x\,dy = \frac{1}{2b}\int_0^a y^2 dy = \frac{1}{2b}\left[\frac{y^3}{3}\right]_0^a = \frac{1}{6b}a^3 = \frac{1}{6}\kappa a^3, \tag{11.13}$$

where a is the radius of the circular window; the result is an approximation which is only valid if the curvature $\kappa(= 1/b)$ is small and applies because A is actually bounded by the curved boundary of the window rather than a horizontal line distant a from the x-axis. For a median contour, A must be equal to aD. Hence

$$D = \frac{1}{6}\kappa a^2. \tag{11.14}$$

This proves the result for a median filter for all shapes of edge profile, and for the mean and mode filters when there is a step edge profile. Moving on to the mean filter when there is a linear edge profile, we find that we have a similar calculation to perform with similar approximations, except that in this case we have to find the mean intensity in the window using a polar coordinate representation, and then deduce for what contour the mean intensity would arise. This leads to the formula

$$D = \frac{1}{8}\kappa a^2. \tag{11.15}$$

For details of the calculation, the reader is referred to [23, 24].

TABLE 11.1. Summary of shifts predicted on continuum model. (Reproduced from E. R. Davies, "Image distortions produced by mean, median, and mode filters," IEE Proceedings-Vision Image and Signal Processing, © 1999 IEE.)

Edge type	Mean	Median	Mode
Step	$\frac{1}{6}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{6}\kappa a^2$
Intermediate	$\sim \frac{1}{7}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{2}\kappa a^2$
Linear	$\frac{1}{8}\kappa a^2$	$\frac{1}{6}\kappa a^2$	$\frac{1}{2}\kappa a^2$

In the case of the mode filter when there is a step edge profile, the situation turns out to be far simpler and no integration is required [24, 25]. This is because the contour with the mode intensity must be the longest one, and is therefore the one which spans the longest diameter of the window (Figure 11.13). Thus we merely have to ask how far this contour is from the center of the window. The answer is given by setting $x = D$ and $y = a$ in (11.11) and (11.12), leading to the formula

$$D = \frac{1}{2}\kappa a^2. \tag{11.16}$$

All the above results are summarized in Table 11.1. Table 11.1 also includes a set of estimates of shifts for intermediate intensity patterns—namely those whose profiles lie between a linear edge and a step edge. Note, however, that a full explanation of the variations in the shifts for the mode filter is quite complex [24].

11.9.1. Using a discrete model to explain median shifts

The explanation of edge shifts given above was completely analog in nature, as it effectively dealt with spatial densities of pixels in various parts of the filter window. While it leads to good estimates of the shifts for large windows, it is inaccurate for small windows and thus a new discrete model had to be developed to make accurate predictions of the shifts [22, 26].

To produce a discrete model, we need to recognize explicitly the positions of the pixels within an $n \times n$ window. We approximate by assuming that the intensity of any pixel is the mean intensity over the whole pixel and is represented by a sample positioned at the center of the pixel. We start by examining the case of a 3×3 window, and proceed by taking the underlying analog intensity variation to have contours of curvature κ.

First, zero shift occurs for $\kappa = 0$. Next, if κ is even minutely greater than zero, the center pixel will not necessarily be the median pixel. Consider the case when the circular median intensity contour passes close to the center of the window at a small angle θ to the positive x-axis (Figures 11.14 and 11.15(a)). In that case, the

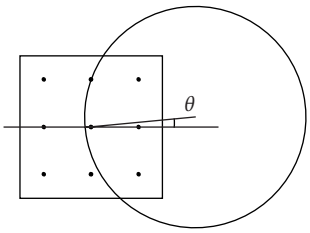


FIGURE 11.14. Geometry for the calculation of median shifts on the discrete model. (Reproduced from E. R. Davies, "Image distortions produced by mean, median, and mode filters," IEE Proceedings-Vision Image and Signal Processing, © 1999 IEE.)

filter will produce a definite shift, whose value is

$$D_\theta \approx \frac{1}{2}\kappa a_0^2 - a_0\theta = \frac{1}{2}\kappa - \theta, \tag{11.17}$$

where a_0 is the interpixel separation, which is taken as unity in this chapter. (The factor 1/2 in the first term arises following (11.12) when y is replaced by a_0.) If θ is close to 45°, the shift will have the value

$$D_\phi \approx \frac{1}{2}\kappa(\sqrt{2}a_0)^2 - (\sqrt{2}a_0)\phi = \kappa - \sqrt{2}\phi, \tag{11.18}$$

where

$$\phi = \frac{\pi}{4} - \theta \tag{11.19}$$

and the relevant pixel separation is $\sqrt{2}a_0$ rather than a_0 (see Figure 11.15(c)).

As indicated above, (11.17) and (11.18) are approximate. However, the above derivations and solutions provide a useful insight into the situation. In particular, (11.17) and (11.18) show that at the ends of the range $0 \le \theta \le \pi/4$, D_θ varies in proportion to $\Delta\theta$. The next problem is understanding what happens when D_θ falls to zero at intermediate values of θ. In fact, D_θ remains at zero in this range, the reason being that the median contour reverts to passing through the central pixel in the window (Figure 11.15(b)). The resulting approximately piecewise-linear variation in D_θ is far from what would be expected on the continuum model. To make a realistic comparison, we must average over all θ. In that case we obtain the result

$$\begin{aligned} D &\approx \frac{\left[\int_0^{\kappa/2}(\kappa/2-\theta)d\theta + \int_0^{\kappa/\sqrt{2}}(\kappa-\sqrt{2}\phi)d\phi\right]}{(\pi/4)} \\ &= \frac{4}{\pi}\left\{\left[-\frac{1}{2}\left(\frac{\kappa}{2}-\theta\right)^2\right]_0^{\kappa/2} + \left[-\frac{1}{2\sqrt{2}}(\kappa-\sqrt{2}\phi)^2\right]_0^{\kappa/\sqrt{2}}\right\} \\ &= \frac{1+2\sqrt{2}}{2\pi}\kappa^2 \approx 0.61\kappa^2. \end{aligned} \tag{11.20}$$

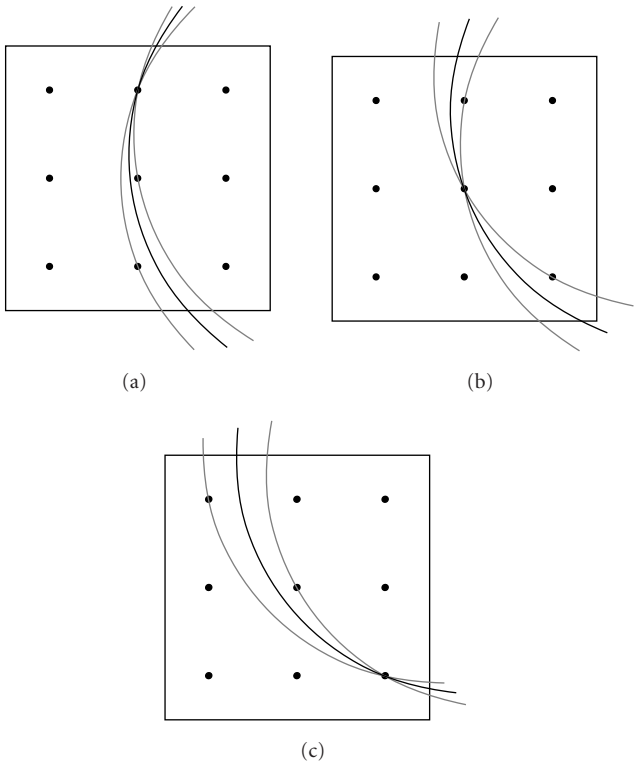


FIGURE 11.15. Geometry for the calculation of median shifts at low κ. These three diagrams show the positions of the median pixels and the ranges of orientations of circular intensity contours for which they apply: (a) for low θ, (b) for intermediate θ, and (c) for high θ. (Reproduced from E. R. Davies, "Image distortions produced by mean, median, and mode filters," IEE Proceedings-Vision Image and Signal Processing, © 1999 IEE.)

This shows that D follows a square law rather than a linear law at low values of κ, unlike the situation for the continuum model. However, for increasing values of κ, the variation reverts ever more closely to a linear model: this occurs when κ reaches such high values that the range of values of θ for which $D_\theta = 0$ falls to zero.

11.9.2. The situation for color images

Interestingly, the continuum and discrete shift models given above apply without change for color images. It is clear that they apply for the individual color channels. However, why should they apply when the colors are considered together? The reason is that the models are rather special in considering single idealized circular edges, with no complicating factors arising from other edge fragments (such as might arise from shadows or image clutter) or from noise. The models were very

carefully set up to consider what happened to the underlying signal. Other work had already been carried out on the effects of noise and how noise can introduce edge shifts (as has already been considered in Section 11.5) [27, 28]. Thus the idea of the two types of the model was to provide a framework on which noise aspects could be considered properly.

As a check on the two models and the theories of edge shifts they gave rise to, careful experimental measurements were made by Charles and Davies [12] for median filters: no detectable difference between the color and gray-scale shifts was found. Exactly the same situation occurred for color and gray-scale mode filters (though at certain curvature values the mode shifts suddenly rose sharply relative to the median shifts—a result for which a reasonable explanation was found [12]).

Overall, intrinsic shifts have been found to occur for binary, gray-scale, and color images when median, mean, and mode filters are applied. It has been found possible to explain these shifts using continuum models and, to a limited extent, (but where this is possible, more accurately) using discrete models.[5] As the shifts are intrinsic rather than being due to noise, they are unavoidable, and in practical applications, they can be dealt with in the following ways:

(1) by avoiding the use of filters wherever possible;
(2) by applying special types of filter such as the hybrid median filter [29] or the neural network (trained) filter [30];
(3) by correcting for the shifts using the models outlined in this section.

11.10. Review of other color work

In the preceding sections, color arose in a natural way as the simultaneous representation and use of three color channels—normally RGB—and no interaction between them was envisaged. However, it was noted that the results of processing using obvious types of filter could appear unnatural to the human eye. It is also the case that image analysis needs to make best use of the image colors if objects such as faces and trees are to be segmented or recognized easily. Indeed, one does not have to look far in the literature of color vision (human or machine) before questions emerge about the most appropriate color spaces.

If color filtering was solely an abstract process, it would make little difference which of many orthogonal rotations were made in a color space from the original RGB axes. However, human perceptual problems, and the complex task of machine recognition, place a substantial gloss on this question. But the situation is far more complex than this. Spaces such as HSI (hue, saturation, intensity) reexpress the relevant variables nonlinearly, in such a way that hue is an angle and saturation is the argument of the hue, to which varying degrees of whiteness are added. Thus hue is periodic; furthermore, its value is meaningless when $S = 0$, a situation that

[5]Note that the mathematical analysis of discrete models is not trivial, and so far has only been achieved for median filters. Hence continuum models have by no means been superseded: this is why both types of model have been examined in the present chapter.

arises both when $I = 0$ and when $I = 255$;[6] this reflects the "double-cone" structure of color space (the vertices of the cones lie at $(0, 0, 0)$ and $(255, 255, 255)$).

The motivation for the HSI representation is supposedly that of theoretical relevance, but the factors just mentioned make it difficult to work with. As a result, many workers have searched for more appropriate representations within their own spheres of relevance. Normalized RGB (or rgb) is a useful representation, and is also useful as a paradigm for comparison:

$$\begin{aligned} r &= \frac{R}{(R+G+B)}, \\ g &= \frac{G}{(R+G+B)}, \\ b &= \frac{B}{(R+G+B)}. \end{aligned} \tag{11.21}$$

It will be impossible here to do justice to the plethora of color spaces that have been developed. Instead it will be useful to outline a few choices workers have recently made in different areas of application. For further examples and insights into color spaces, see Chapter 12.

11.10.1. Recognizing human skin tones

When finding human skin regions, Chen and Grecos [31] found that skin-tone color pixels are most compact (and therefore most easily segmented) in normalized RGB space, and recalled previous work showing that normalized RG color space performs almost as well as RGB space. Working with this space, they cut down computation and also avoided a color affine transformation. Hsu et al. [32] tackled the same problem, stating that normalized RG space is not the best choice for face detection, and adopting YC_bC_r space since it is "perceptually uniform" and also gives compact skin clusters in color space.

11.10.2. Extraction of multicolored objects

Bucha et al. [33], working on extraction of multicolored cartographic objects, stated that it is difficult to design color image processing algorithms in RGB space because of the high correlation between channels, and asserted that HSI allows more robust measures of the differences between two colors to be made; they then reiterated the difficulties with HSI mentioned earlier. They therefore combined RGB and HSI to obtain more reliable distance measures, following earlier work by Gevers and Smeulders [34]. The latter presented several new color models and gave theoretical justification for choice of color spaces depending on whether the

[6] When $S \approx 0$, a similar situation arises, especially if noise is present.

illumination is controlled, whether highlights are present, and whether the objects to be recognized are multicolored.

11.10.3. Mathematical morphology in color spaces

Hanbury and Serra [35] approach the problem of mathematical morphology in color spaces. In fact, color morphology suffers from exactly the same problems as median (and other) filtering, in that an ordering of the pixels within the window is needed, and in color images it is a priori undefined. Thus an ordering has to be imposed. Lexicographic ordering (applying a dictionary type of ordering, such as ordering by R, then by G and then by B—effectively ordering by the value of $256^2R + 256G + B$) is an obvious one, but is arbitrary: in fact, any such ordering may be appropriate in some situations but poor in others. The chapter aimed to find a theoretically more suitable ordering: specifically, it examined how the HLS space could best be adapted for the purpose (a priori, hue should be taken first, then saturation and finally luminance). They found that because HLS space has a double-cone shape, it is far better to weight hue by saturation, and only then to prioritize using hue, saturation, and luminance (note that hue had to be applied on its own to avoid introducing false colors). Another aspect of this work was the need to choose an origin of hue, as this permitted ordering on the hue unit circle—hue being a periodic function. However, this imparted a degree of arbitrariness to the methodology, which would make it less general but at the same time more adaptable to different recognition tasks. Discussions of some alternative schemes are given by Hanbury and Serra [36].

J. Li and Y. Li [37] took a rather different approach to ordering in color space by concentrating on principal component analysis (PCA) to provide a suitable ordering procedure. As a result, the method is able to preserve colors in the same way as the vector median filter.

11.10.4. Effect of varying illumination and shadows

All the cases listed so far provide fixed color spaces with perhaps one or two parameters that are adjusted to optimize for any given application. The approach made by Finlayson et al. [38, 39] looks instead at achieving color constancy—that is, resistance to natural variations in illumination on objects. In this work, two images of a scene are taken, in one of which a color filter is placed in front of the camera. This gives additional information which allows the color to be normalized. Specifically, the additional information is of such a type that it can be used to cancel out misleading information that arises as the background illumination changes. The work is founded on rigorous theory and is practically impressive; for details, the reader is referred to [38, 39]. In related work, Finlayson et al. have found how to eliminate shadows from color images [40]: only a single image is needed once the camera has been calibrated. Both of these sets of work can be called "filtering," though they are some way away from conventional filtering as described in the literature as, to some degree, they embody global understanding of image content.

11.11. Summary

This chapter has presented a short survey of image processing, color filtering, and their inter-relationship. There is clearly much more to be said on this topic, given its marked overlap with image analysis and recognition. One point that has emerged strongly is that image processing does not happen in a vacuum: it is needed as an aid both for human interaction and for computer interpretation. In either, case divergences and preferences exist. What humans regard as divisive color bleeding may be largely irrelevant to a computer recognition algorithm, though in the latter case it could introduce inaccuracies of measurement.

At first, color bleeding appeared to be an accidental occurrence in image filtering. However, its incidence is deeper, and it applies equally to mathematical morphology, which is a widely used, rather generalized approach to the processing and analysis of images. Intrinsic to bleeding is the need to order image data points, and in a multichannel context, this is fraught with difficulty. Some approaches to multichannel ordering, such as lexicographic ordering, are obviously arbitrary, and there is no reason why they would work well in practice (except with restricted data sets and algorithm settings). More systematic approaches aim to identify relevant color representations, but these will often be applicable only to restricted types of data. Methods based on PCA have more right to be called general, but (as in many other applications of PCA) they may still be suboptimal and at this stage they do not appear to be fully proven experimentally in the color context.

Color is a complex topic: while it is perceptually simple for humans, it is difficult to program and equally difficult to be sure of optimality. In fact, the computer vision community has not had anything like as much experience with color as with gray-scale images, but with the advent of very cheap color cameras, this situation seems certain to change quite quickly. Nevertheless, the multichannel ordering problem is a hard one, and much more work is required before the theoretical backcloth is fully prepared for guaranteeing optimal algorithms. The following chapter confirms that the problems recounted in the present chapter are not parochial, but are manifest for a good many aspects of color processing. For further reading, a number of useful articles on color image processing appear in [41]: in particular, the article by Lukac et al. on vector filtering contains valuable additional topics such as vector directional filters which could not be covered adequately in the present chapter [42].

Acknowledgments

Professor E. R. Davies is grateful to Research Councils UK for Basic Technology Grant GR/R87642/02. Table 11.1, Figures 11.8, 11.12–11.15, and some of the text are reproduced from [11, 22] with permission from the IEE; Figures 11.2, 11.7, 11.9, 11.10, and some of the text are reproduced from [12] with permission from the Royal Photographic Society.

Bibliography

[1] G. Ward, "High dynamic range imaging," in *Invited Paper at the IEE International Conference on Visual Information Engineering (VIE '05)*, Glasgow, Scotland, UK, April 2005.

[2] P. H. Heinemann, Z. A. Varghese, C. T. Morrow, H. J. Sommer III, and R. M. Crassweller, "Machine vision inspection of 'Golden Delicious' apples," *Applied Engineering in Agriculture*, vol. 11, no. 6, pp. 901–906, 1995.

[3] Y. Ohta, *Knowledge-Based Interpretation of Outdoor Natural Color Scenes*, Pitman, Marshfield, Mass, USA, 1985.

[4] E. R. Davies, *Electronics, Noise and Signal Recovery*, Academic Press, London, UK, 1993.

[5] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Morgan Kaufmann, San Francisco, Calif, USA, 3rd edition, 2005.

[6] E. R. Davies, "Using an edge-based model of the Plessey operator to determine localisation properties," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '05)*, pp. 385–391, Glasgow, Scotland, UK, April 2005.

[7] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proceedings of IEEE*, vol. 78, no. 4, pp. 678–689, 1990.

[8] E. R. Davies, "A remanent noise problem with the median filter," in *Proceedings of 11th IAPR International Conference on Pattern Recognition (ICPR '92)*, vol. 3, pp. 505–508, The Hague, The Netherlands, August–September 1992.

[9] E. R. Davies, "Accuracy of multichannel median filter," *Electronics Letters*, vol. 36, no. 25, pp. 2068–2069, 2000.

[10] E. R. Davies, "On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters," *Pattern Recognition Letters*, vol. 7, no. 2, pp. 87–97, 1988.

[11] D. Charles and E. R. Davies, "Properties of the mode filter when applied to colour images," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '03)*, IEE Conference Publication, no. 495, pp. 101–104, Guildford, UK, July 2003.

[12] D. Charles and E. R. Davies, "Mode filters and their effectiveness for enhancing colour images," *Imaging Science Journal*, vol. 52, no. 1, pp. 3–25, 2004.

[13] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 242–251, 2001.

[14] T. Chen and H. R. Wu, "Application of partition-based median type filters for suppressing noise in images," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 829–836, 2001.

[15] R. Lukac, "Adaptive vector median filtering," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1889–1899, 2003.

[16] X. Xu, E. L. Miller, D. Chen, and M. Sarhadi, "Adaptive two-pass rank order filter to remove impulse noise in highly corrupted images," *IEEE Transactions on Image Processing*, vol. 13, no. 2, pp. 238–247, 2004.

[17] T. Sun and Y. Neuvo, "A simple synthesis method for weighted median filters," in *Proceedings of 6th European Signal Processing Conference (EUSIPCO '92)*, pp. 1405–1408, Brussels, Belgium, August 1992.

[18] S. Fotopoulos, D. Sindoukas, N. Laskaris, and G. Economou, "Image enhancement using color and spatial information," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, vol. 4, pp. 2581–2584, Munich, Germany, April 1997.

[19] S. J. Sangwine and R. E. N. Horne, Eds., *The Colour Image Processing Handbook*, Chapman & Hall, London, UK, 1998.

[20] S. Marshall, "New direct design method for weighted order statistic filters," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 151, no. 1, pp. 1–8, 2004.

[21] D. Charles and E. R. Davies, "Distance-weighted median filters and their application to colour images," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '03)*, IEE Conference Publication, no. 495, pp. 117–120, Guildford, UK, July 2003.

[22] E. R. Davies, "Image distortions produced by mean, median and mode filters," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 146, no. 5, pp. 279–285, 1999.

[23] E. R. Davies, "Median and mean filters produce similar shifts on curved boundaries," *Electronics Letters*, vol. 27, no. 10, pp. 826–828, 1991.

[24] E. R. Davies, "An analysis of the geometric distortions produced by median and related image processing filters," *Advances in Imaging and Electron Physics*, vol. 126, pp. 93–193, 2003.

[25] E. R. Davies, "Shifts produced by mode filters on curved intensity contours," *Electronics Letters*, vol. 33, no. 5, pp. 381–382, 1997.

[26] E. R. Davies, "Formulation of an accurate discrete theory of median shifts," *Signal Processing*, vol. 83, no. 3, pp. 531–544, 2003.

[27] G. J. Yang and T. S. Huang, "The effect of median filtering on edge location estimation," *Computer Graphics and Image Processing*, vol. 15, no. 3, pp. 224–245, 1981.

[28] A. C. Bovik, T. S. Huang, and D. C. Munson Jr., "The effect of median filtering on edge estimation and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 181–194, 1987.

[29] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 74–90, 1987.

[30] D. Greenhill and E. R. Davies, "Relative effectiveness of neural networks for image noise suppression," in *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, E. S. Gelsema and L. N. Kanal, Eds., pp. 367–378, Elsevier Science, Amsterdam, The Netherlands, 1994.

[31] L. Chen and C. Grecos, "A fast skin region detector for colour images," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '05)*, pp. 195–201, Glasgow, Scotland, UK, April 2005.

[32] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[33] V. Bucha, S. Ableameyko, and T. Pridmore, "Semi-automatic extraction and vectorisation of multicoloured cartographic objects," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '05)*, pp. 115–120, Glasgow, Scotland, UK, April 2005.

[34] T. Gevers and A. W. M. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[35] A. G. Hanbury and J. Serra, "Mathematical morphology in the HLS colour space," in *Proceedings of 12th British Machine Vision Conference (BMVC '01)*, vol. 2, pp. 451–460, Manchester, UK, September 2001.

[36] A. G. Hanbury and J. Serra, "Morphological operators on the unit circle," *IEEE Transactions on Image Processing*, vol. 10, no. 12, pp. 1842–1850, 2001.

[37] J. Li and Y. Li, "Multivariate mathematical morphology based on principal component analysis: initial results in building extraction," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. B7, pp. 1168–1173, 2004.

[38] G. D. Finlayson and P. Morovic, "Human visual processing: beyond 3 sensors," in *Proceedings of IEE International Conference on Visual Information Engineering (VIE '05)*, pp. 1–7, Glasgow, Scotland, UK, April 2005.

[39] G. D. Finlayson, S. D. Hordley, and P. Morovic, "Colour constancy using the chromagenic constraint," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 1079–1086, San Diego, Calif, USA, June 2005.

[40] G. D. Finlayson, S. D. Hordley, and M. S. Drew, "Removing shadows from images," in *Proceedings of 7th European Conference on Computer Vision-Part IV (ECCV '02)*, pp. 823–836, Copenhagen, Denmark, May 2002.

[41] H. J. Trussell, E. Saber, and M. Vrhel, "Color image processing [basics and special issue overview]," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 14–22, 2005.

[42] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 74–86, 2005.

E. R. Davies: Machine Vision Group, Department of Physics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK

Email: e.r.davies@rhul.ac.uk

D. Charles: Machine Vision Group, Department of Physics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK
Current address: Fuji Film Electronic Imaging, Rickmansworth WD3 1RE, UK

Email: derekcharles@hotmail.com