Published in IET Image Processing
Received on 2nd September 2008
doi: 10.1049/iet-ipr.2009.0056

ISSN 1751-9659

Distance measures for reduced ordering-based vector filters

M. Emre Celebi

Department of Computer Science, Louisiana State University, Shreveport, LA, USA
E-mail: ecelebi@lsus.edu

Abstract: Reduced ordering-based vector filters have proved successful in removing long-tailed noise from colour images while preserving edges and fine image details. These filters commonly utilise variants of the Minkowski distance to order the colour vectors with the aim of distinguishing between noisy and noise-free vectors. In this study, the authors review various alternative distance measures and evaluate their performance on a large and diverse set of images using several effectiveness and efficiency criteria. The results demonstrate that there are in fact strong alternatives to the popular Minkowski metrics.

1 Introduction

Colour images are often contaminated with noise, which is introduced during acquisition (sensor noise) and/or transmission (channel noise). Sensor noise is inherent to all electronic image sensors and is often modelled using Gaussian or Poisson distributions [1]. Channel noise is caused by various sources including man made phenomena (car ignition systems, industrial machines in the vicinity of the receiver, switching transients in power lines, various unprotected switches etc.) and natural causes (lightning in the atmosphere, ice cracking in the Antarctic region etc.) [2]. Channel noise is often characterised by impulsive sequences, which occur in the form of short-duration, high-energy spikes attaining large amplitudes with probability higher than that predicted by the Gaussian density model [3]. The introduction of such noise into an image not only lowers its perceptual quality, but also makes subsequent tasks such as edge detection and segmentation more difficult. Therefore noise removal is often an essential preprocessing step in many colour image processing applications.

Numerous filters have been proposed for the removal of long-tailed, in particular impulsive, noise from colour images [1–6]. Among these, non-linear vector filters have proved successful in removing the noise while preserving the edges and fine image details. The early approaches to non-linear filtering of colour images often involved the application of a scalar filter to each colour channel independently. However, since separate processing ignores the inherent correlation between the colour channels, these methods often introduce colour artefacts to which the human visual system is very sensitive. Therefore vector filtering techniques that treat the colour image as a vector field and process colour pixels as vectors are more appropriate.

An important class of non-linear vector filters is the one based on robust order-statistics. These filters involve the ordering of a set of input vectors inside a predefined sliding window. The lowest ranked input vector in this window is often taken as the output vector. The ordering is commonly achieved using variants of the Minkowski distance. Recently, fuzzy similarity measures [7–16] have been proposed as alternatives to the more traditional distance measures. Since it is often trivial to convert a similarity measure to a dissimilarity measure, throughout this paper we will use the term 'distance measure' to refer to a function that quantifies the distance or similarity between two vectors.

In this paper, we review various distance measures that can be used to order colour vectors in reduced ordering-based vector filters. We evaluate the performance of these measures on a large set of images that cover a variety of domains using several effectiveness and efficiency criteria. The rest of the paper is organised as follows. Section 2 presents an overview of order-statistics based vector filters and distance measures that can be utilised to order colour

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

vectors. Section 3 describes the noise model, filtering performance criteria and the experimental setup. Finally, Section 4 gives the conclusions and the future work.

2 Distance measures for reduced ordering of colour vectors

Consider an $M \times N$ red–green–blue (RGB) input image $\boldsymbol{X}$ that represents a two-dimensional (2D) array of three-component vectors $\boldsymbol{x}(r, c) = [x_1(r, c), x_2(r, c), x_3(r, c)]$ occupying the spatial location (r, c), with the row and column indices $r = \{1, \ldots, M\}$ and $c = \{1, \ldots, N\}$, respectively. In the pixel $\boldsymbol{x}(r, c)$, the $x_k(r, c)$ values denote the red $(k = 1)$, green $(k = 2)$ and blue $(k = 3)$ components. Most images are non-stationary in nature; therefore filters often operate on the assumption that the local image features can be extracted from a small image region called a sliding window. The size and shape of the window influence the properties and efficiency of the filtering operation and are therefore application dependent [1]. A square window $W(r, c)$ of size $\sqrt{n} \times \sqrt{n}$ pixels centred on $\boldsymbol{x}(r, c)$ is commonly used because of its versatility and good performance [4]. The window slides over the entire image $\boldsymbol{X}$ in a raster fashion and the procedure replaces the input vector $\boldsymbol{x}(r, c)$ with the output $\boldsymbol{y}(r, c) = F(W(r, c))$ of a filter function $F(\cdot)$ that operates over the samples inside $W(r, c)$. Repeating the procedure for each pair (r, c), with $r = \{1, \ldots, M\}$ and $c = \{1, \ldots, N\}$, produces the output vectors $\boldsymbol{y}(r, c)$ of the $M \times N$ filtered image $\boldsymbol{Y}$. For notational simplicity, the input vectors inside $W(r, c)$ are re-indexed as a set, that is $W(r, c) = \{\boldsymbol{x}_i : i = 1, \ldots, n\}$ (see Fig. 1), as commonly seen in the related literature [1–6]. In this notation, the centre pixel in W is given by $\boldsymbol{x}_{(n+1)/2}$ and in the vector $\boldsymbol{x}_i = [x_{i1}, x_{i2}, x_{i3}]$ with components x_{ik}, the $i \in \{1, \ldots, n\}$ and $k \in \{1, 2, 3\}$ indices denote the position of the vector inside the window and the colour channel, respectively.

Order-statistics based vector filters operate by ordering (ranking) the vectors inside the sliding window. The purpose of this ordering is to distinguish between the noisy and noise-free vectors. However, in contrast to scalar data, there is no universally accepted method for ordering multivariate data. Widely known multivariate ordering methods include [17]:

- *Marginal ordering (M-ordering):* The vectors are ordered in each component independently. This scheme produces a set of ordered output vectors that is usually not the same as the set of input vectors and consequently results in colour artefacts when applied to multichannel image data.

- *Conditional ordering (C-ordering):* The vectors are ordered based on the marginal ordering of one of the components. This scheme disregards the vectorial nature of the multichannel image data.

- *Partial ordering (P-ordering):* The vectors are partitioned into smaller groups that are then ordered. Despite its theoretical appeal, this scheme is computationally demanding [18].

- *Reduced (aggregate) ordering (R-ordering):* The vectors are first reduced to scalar representatives using a suitable distance measure. The ordering of these scalars is then taken as the ordering of the corresponding vectors. This is the most common ordering scheme in the literature [1–6].

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \\ \mathbf{X}_4 & \mathbf{X}_5 & \mathbf{X}_6 \\ \mathbf{X}_7 & \mathbf{X}_8 & \mathbf{X}_9 \end{bmatrix}$$

Figure 1 *Indexing convention inside a 3 × 3 window*

To order the vectors inside the sliding window $W(r, c) = \{\boldsymbol{x}_i : i = 1, \ldots, n\}$, R-ordering-based vector filters first calculate the aggregate (cumulative) distance D_i for each vector $\boldsymbol{x}_i$

$$D_i = \sum_{j=1}^{n} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{1}$$

where $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes a suitable distance measure that quantifies the dissimilarity between vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The ordered sequence of scalars $D_{(1)} \leq D_{(2)} \leq \cdots \leq D_{(n)}$ implies the same ordering in the corresponding vectors $\boldsymbol{x}_{(1)} \leq \boldsymbol{x}_{(2)} \leq \cdots \leq \boldsymbol{x}_{(n)}$. Many filters define the lowest ranked input vector (lowest order-statistic), that is $\boldsymbol{x}_{(1)}$, as the output vector. This is because vectors with lower ranks are typical (representative) vectors in their neighbourhoods, whereas those with higher ranks are often outliers (noisy vectors) [19]. More explicitly, the output vector at location (r, c) is given by

$$\boldsymbol{y}(r, c) = \underset{\boldsymbol{x}_i \in W(r,c)}{\operatorname{argmin}} D_i \tag{2}$$

Note that if a similarity measure $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is used instead of $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in (1), the argmin operator should be replaced with argmax.

The vector median filter (VMF) [20], which is the most well known R-ordering-based vector filter, utilises the Minkowski distance $d_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\sum_{k=1}^{3} |x_{ik} - x_{jk}|^p)^{1/p}$ as its ordering criterion. Three special cases of d_p are of particular interest: d_1 (city-block distance), d_2 (Euclidean distance) and d_∞ (chessboard distance). Given the general Minkowski form, d_1 and d_2 can be defined in a straightforward fashion, whereas d_∞ is defined as $d_\infty(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max_{1 \leq k \leq 3} |x_{ik} - x_{jk}|$. Among these three forms, d_2 seems to be preferred in the colour image filtering literature due to its isotropic (rotation invariant) nature and good performance [21]. The squared form of d_2, that is d_2^2, can also be used as a computationally efficient alternative [22].

© The Institution of Engineering and Technology 2009

doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

Table 1 lists some of the traditional distance measures that can be used for the reduced ordering of colour vectors. Note that the Bray–Curtis, Canberra, Soergel and Ware–Hedges measures are normalised forms of the city-block distance, whereas the divergence coefficient measure is a normalised form of the Euclidean distance. Some of the measures given in Table 1 have already been used in the design of various R-ordering-based vector filters. For example, the Cosine distance was used in the design of the basic vector directional filter (BVDF) [23]. The Goude distance was used in the content-based rank filter [24]. The rationale behind this measure is that the similarity between two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be expressed as the ratio of some function of what they share (commonality) to what they comprise (totality). In particular, the numerator and denominator correspond to the vector difference and the vector sum, respectively.

In [9], the authors introduced a fuzzy magnitude similarity measure given by

$$s^{K}_{\text{fms}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{k=1}^{3} \frac{\min(x_{ik}, x_{jk}) + K}{\max(x_{ik}, x_{jk}) + K} \tag{3}$$

where K is a parameter of the measure.

The analogous fuzzy directional similarity measure between vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be obtained by first normalising the vectors to unit length, that is $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i / \|\boldsymbol{x}_i\|$ and $\hat{\boldsymbol{x}}_j = \boldsymbol{x}_j / \|\boldsymbol{x}_j\|$, and then calculating the fuzzy magnitude similarity between them, that is $s^{K}_{\text{fds}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = s^{K}_{\text{fms}}(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j)$ [11]. Furthermore, since both fuzzy magnitude similarity and fuzzy directional similarity values fall into the (0, 1] interval, they can easily be combined to obtain a fuzzy magnitude-directional similarity measure as follows: $s^{K_1,K_2}_{\text{fmds}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = s^{K_1}_{\text{fms}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot s^{K_2}_{\text{fds}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ [11].

In [10], the authors introduced a combined fuzzy similarity measure given by

$$s^{C,t}_{\text{cfs}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{C}{C + d_2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \cdot \frac{t}{t + \max(|r_i - r_j|, |c_i - c_j|)} \tag{4}$$

where (r_i, c_i) and (r_j, c_j) are the spatial coordinates of the vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively, and C and t are the parameters of the measure. It can be seen that $s^{C,t}_{\text{cfs}}$ combines colour similarity (first term) and spatial proximity (second term). The maximisation of the aggregate combined fuzzy similarity in a particular neighbourhood is equivalent to finding the vector that is centrally located and is a good representative of the neighbourhood in terms of chromatic content.

Table 1 Distance measures for R-ordering of colour vectors ($\|\boldsymbol{x}\|$: Euclidean norm of $\boldsymbol{x}$)

Bray–Curtis distance [26]	$d_{\text{bray}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \dfrac{\sum_{k=1}^{3} \lvert x_{ik} - x_{jk} \rvert}{\sum_{k=1}^{3} x_{ik} + x_{jk}}$
Canberra distance [27]	$d_{\text{canberra}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{3} \dfrac{\lvert x_{ik} - x_{jk} \rvert}{x_{ik} + x_{jk}}$
Chord distance [28]	$d_{\text{chord}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{k=1}^{3} \left(\sqrt{x_{ik}} - \sqrt{x_{jk}} \right)^2 \right)^{1/2}$
Cosine distance [23]	$d_{\text{cosine}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta = \arccos \left(\dfrac{\sum_{k=1}^{3} x_{ik} x_{jk}}{\|\boldsymbol{x}_i\| \, \|\boldsymbol{x}_j\|} \right)$
divergence coefficient [29]	$d_{\text{divergence}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\sum_{k=1}^{3} \dfrac{(x_{ik} - x_{jk})^2}{x_{ik} + x_{jk}} \right)^{1/2}$
Goude distance [30]	$d_{\text{goude}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\dfrac{\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_j\|^2 - 2 \sum_{k=1}^{3} x_{ik} x_{jk}}{\|\boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_j\|^2 + 2 \sum_{k=1}^{3} x_{ik} x_{jk}} \right)^{1/2}$
Soergel distance [31]	$d_{\text{soergel}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \dfrac{\sum_{k=1}^{3} \lvert x_{ik} - x_{jk} \rvert}{\sum_{k=1}^{3} \max(x_{ik}, x_{jk})}$
Ware–Hedges distance [32]	$d_{\text{ware}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{k=1}^{3} \dfrac{\lvert x_{ik} - x_{jk} \rvert}{\max(x_{ik}, x_{jk})}$

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

The ordering criterion used in the directional-distance filter (DDF) [25] combines the Minkowski and Cosine distances. In particular, the aggregate distance associated with pixel $\boldsymbol{x}_i$ is given by

$$D_i = \left(\sum_{j=1}^{n} d_p(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)\left(\sum_{j=1}^{n} d_{\text{cosine}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) \quad (5)$$

Note that the idea of combining directional and magnitude processing also underlies the design of s_{fmds}. However, while s_{fmds} is a proper similarity measure, that is it quantifies the similarity of two vectors, the ordering criterion given in (5) is only defined in a particular neighbourhood.

A function $f: S \times S \to \mathbb{R}$ is called a metric on a set S if it satisfies the following conditions for all $x, y, z \in S$ [33]:

1. $f(x, y) \geq 0$ (non-negativity)
2. $f(x, y) = 0$ if and only if $x = y$ (self-identity)
3. $f(x, y) = f(y, x)$ (symmetry)
4. $f(x, y) \leq f(x, z) + f(y, z)$ (triangle inequality)

Among the distance measures presented in this section, the squared euclidean, Bray–Curtis and Goude measures are not metrics on $\mathbb{R}$ since they violate the triangle inequality. The Cosine distance is a semi (pseudo) metric on $\mathbb{R}$ because it satisfies all of the conditions except for the self-identity condition. On the other hand, it can be shown that [9–11] the fuzzy similarity measures s_{fms}, s_{fds}, s_{fmds} and s_{cfs} are stationary F-bounded fuzzy metrics in George and Veeramani's sense [34].

The operation counts for each distance measure are given in Table 2 (ABS: absolute value; COMP: comparison; ADD: addition; SUB: subtraction; MULT: multiplication; DIV: division; SQRT: square root; ARCCOS: inverse cosine). Note that these operations are listed in ascending order according to their computational costs on a typical processor architecture, that is cost(ABS) $\leq$ cost(COMP) $\leq \cdots \leq$ cost(SQRT) $\leq$ cost(ARCCOS). This may not hold on certain architectures, but it nevertheless permits cost comparisons between the measures to a certain extent. The actual cost of each operation is highly dependent on the implementation platform (the processor and the compiler).

It should be noted that, even if it is mathematically simple, the use of a particular distance measure in an R-ordering-based vector filter can be computationally demanding. This

Table 2 Operation counts for the distance measures

Measure	ABS	COMP	ADD/SUB	MULT	DIV	SQRT	ARCCOS
d_1	3	–	5	–	–	–	–
d_2	–	–	5	3	–	1	–
d_2^2	–	–	5	3	–	–	–
d_∞	3	2	3	–	–	–	
d_{bray}	3	–	10	–	1	–	–
d_{canberra}	3	–	8	–	3	–	–
s_{cfs}	2	1	9	4	2	1	–
d_{chord}	–	–	5	3	–	7	–
d_{cosine}	–	–	6	10	1	1	1
$d_{\text{divergence}}$	–	–	8	3	3	1	–
s_{fds}	–	3	10	8	9	2	–
s_{fms}	–	3	6	2	3	–	–
s_{fmds}	–	6	16	11	12	2	–
d_{goude}	–	–	9	10	1	1	–
d_{soergel}	3	3	7	–	1	–	–
d_{ware}	3	3	5	–	3	–	–

© The Institution of Engineering and Technology 2009
IET Image Process., 2009, Vol. 3, Iss. 5, pp. 249–260
doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

is because during the filtering procedure the aggregate distance value (1) is calculated for every vector $\boldsymbol{x}_i$ in each neighbourhood $W(r, c)$. This means that the determination of the output vector $\boldsymbol{y}(r, c)$ requires $n(n-1)/2$ distance evaluations. For an $M \times N$ image this amounts to $MNn(n-1)/2$ evaluations. As an example, filtering a 512 $\times$ 512 image using a 3 $\times$ 3 window requires over 9.4 million evaluations. Fortunately, the computation of certain distance measures can be accelerated using table lookups (see Section 3).

3 Experimental results and discussion

In order to evaluate the performance of the presented distance measures, a set of 100 high quality RGB images was collected from the Internet. The set included images of people, animals, plants, buildings, aerial maps, man-made objects, natural scenery, paintings, sketches, as well as scientific, biomedical, synthetic and test images commonly used in the colour image processing literature. The corruption in the images was simulated by the widely used correlated impulsive noise model [35]

$$\boldsymbol{x} = \begin{cases} \boldsymbol{o} & \text{with probability } 1-\varphi \\ \{r_1, o_2, o_3\} & \text{with probability } \varphi_1 \cdot \varphi \\ \{o_1, r_2, o_3\} & \text{with probability } \varphi_2 \cdot \varphi \\ \{o_1, o_2, r_3\} & \text{with probability } \varphi_3 \cdot \varphi \\ \{r_1, r_2, r_3\} & \text{with probability } (1-(\varphi_1+\varphi_2+\varphi_3)) \cdot \varphi \end{cases} \tag{6}$$

where $\boldsymbol{o} = \{o_1, o_2, o_3\}$ and $\boldsymbol{x} = \{x_1, x_2, x_3\}$ represent the original and noisy colour vectors, respectively, $\boldsymbol{r} = \{r_1, r_2, r_3\}$ is a random vector that represents the impulsive noise, φ is the sample corruption probability and φ_1, φ_2, and φ_3 are the corruption probabilities for the red, green and blue channels, respectively. In the experiments, the channel corruption probabilities were set to 0.25.

Filtering performance was evaluated by the following effectiveness criteria [6]:

1. *Mean absolute error:* $\mathrm{MAE}(\boldsymbol{X}, \boldsymbol{Y}) = 1/3MN \sum_{r=1}^{M} \sum_{c=1}^{N} d_1(\boldsymbol{x}(r, c), \boldsymbol{y}(r, c))$ where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote, respectively,

Table 3 Average effectiveness ranking of the distance measures

Measure	MAE			MSE			NCD			Mean
	10%	20%	30%	10%	20%	30%	10%	20%	30%	
s_{cfs}	0.03	0.00	0.00	0.00	0.25	2.67	0.00	0.00	0.04	0.33
s_{fmds}	2.44	2.00	2.23	3.42	2.42	2.01	1.47	1.21	1.30	2.06
s_{fms}	2.20	1.65	1.30	2.07	1.79	1.62	6.49	3.21	2.53	2.54
d_1	3.45	2.86	2.63	1.99	2.42	2.61	8.13	4.90	4.06	3.67
$d_{divergence}$	6.23	6.08	4.85	4.67	3.74	2.94	3.38	3.25	3.21	4.26
d_2	7.61	7.98	7.24	5.97	5.01	4.18	4.12	5.46	5.34	5.88
d_{chord}	9.78	9.69	8.78	7.86	6.58	5.53	6.92	7.66	6.76	7.73
$d_{canberra}$	6.10	5.24	6.03	6.48	8.35	10.43	10.32	8.97	10.46	8.04
ddf	9.53	9.99	9.13	11.00	9.05	7.28	4.95	5.85	5.84	8.07
d_{bray}	6.03	6.40	8.26	6.32	9.19	11.24	10.16	10.05	11.14	8.75
d_{goude}	9.30	11.08	11.61	8.66	9.43	8.94	5.84	8.85	8.99	9.19
$d_{soergel}$	9.02	7.64	7.21	10.25	9.82	10.46	10.90	8.89	8.80	9.22
d_∞	12.19	12.59	12.53	11.11	11.16	9.25	6.15	10.81	10.01	10.64
d_{ware}	10.49	8.35	9.39	11.67	12.36	13.51	12.95	12.24	13.12	11.56
s_{fds}	14.75	14.12	14.09	14.90	14.73	14.72	13.93	13.81	13.65	14.30
d_2^2	15.90	15.73	15.64	14.11	14.03	12.76	15.83	15.88	15.80	15.08
d_{cosine}	15.45	15.12	15.16	15.52	15.67	15.85	14.73	14.96	14.95	15.27
none	12.50	16.48	16.92	17.00	17.00	17.00	16.73	17.00	17.00	16.40

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

the $M \times N$ original and filtered images in the RGB colour space. MAE measures the detail preservation capability of a filter.

2. Mean squared error: $\mathrm{MSE}(\boldsymbol{X}, \boldsymbol{Y}) = 1/3MN \sum_{r=1}^{M} \sum_{c=1}^{N} d_2^2(\boldsymbol{x}(r, c), \boldsymbol{y}(r, c))$ MSE measures the noise suppression capability of a filter.

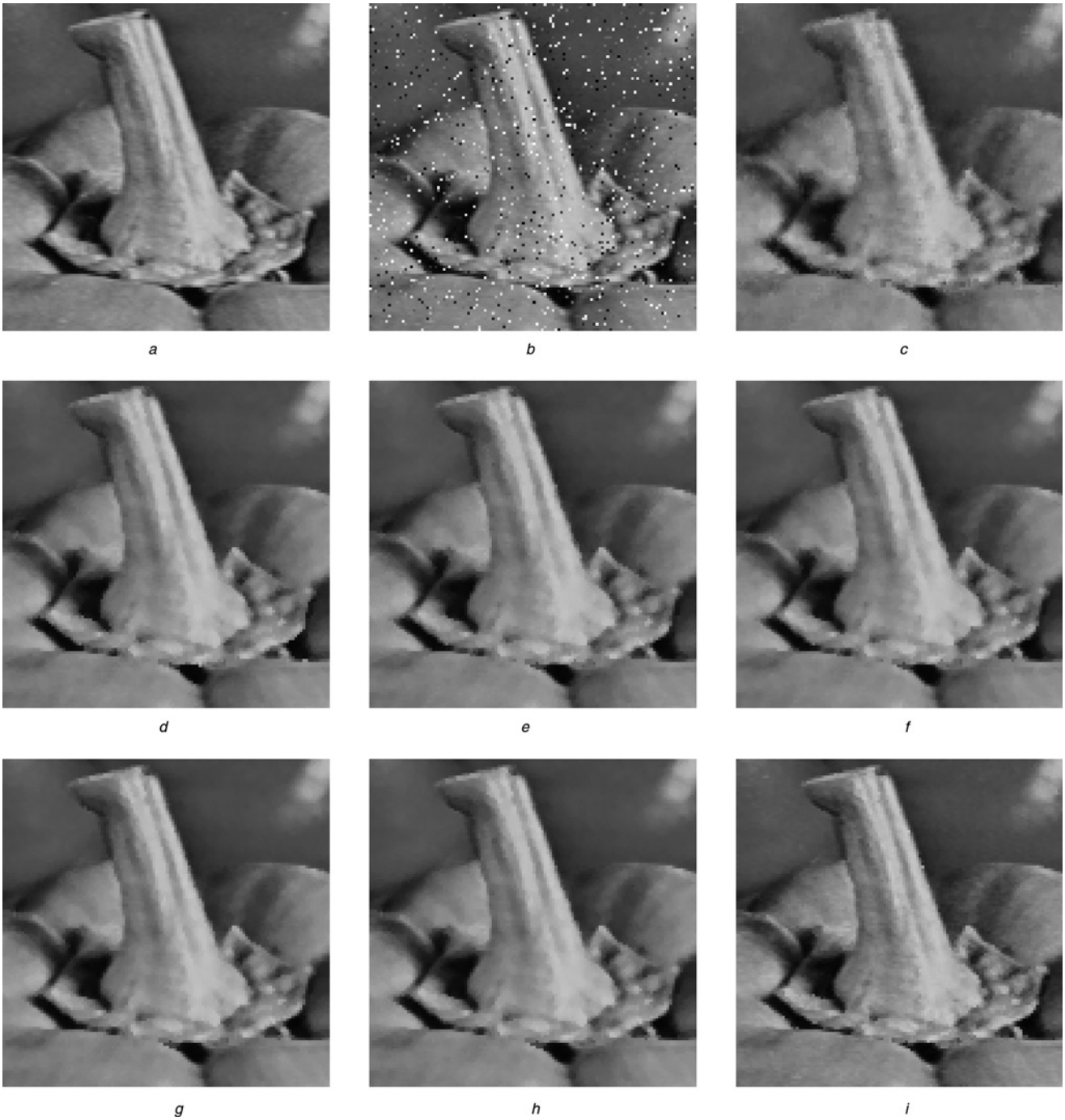

Figure 2 *Filtering results for the Peppers image corrupted with 10% noise*

a Original
b 10% noisy (MAE:6.34, MSE:1037.19)
c d_2^2 (MAE:3.72, MSE:42.88)
d d_{canberra} (MAE:2.35, MSE:26.36)
e $d_{\mathrm{divergence}}$ (MAE:2.29, MSE:20.71)
f d_2 (MAE:2.29, MSE:20.77)
g d_1 (MAE:2.23, MSE:20.70)
h s_{fms} (MAE:2.22, MSE:20.42)
i s_{cfs} (MAE:0.49, MSE:11.06)

IET Image Process., 2009, Vol. 3, Iss. 5, pp. 249–260
© The Institution of Engineering and Technology 2009
doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

3. Normalised colour difference: $\mathrm{NCD}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\sum_{r=1}^{M}\sum_{c=1}^{N} d_2(\boldsymbol{x}^{\mathrm{Lab}}(r,c), \boldsymbol{y}^{\mathrm{Lab}}(r,c))}{\sum_{r=1}^{M}\sum_{c=1}^{N} \|\boldsymbol{x}^{\mathrm{Lab}}(r,c)\|}$ where $\boldsymbol{x}^{\mathrm{Lab}}(r, c)$ and $\boldsymbol{y}^{\mathrm{Lab}}(r, c)$ denote the CIEL*a*b* coordinates of the pixel (r, c) in the original and filtered images, respectively. NCD measures the colour preservation capability of a filter.

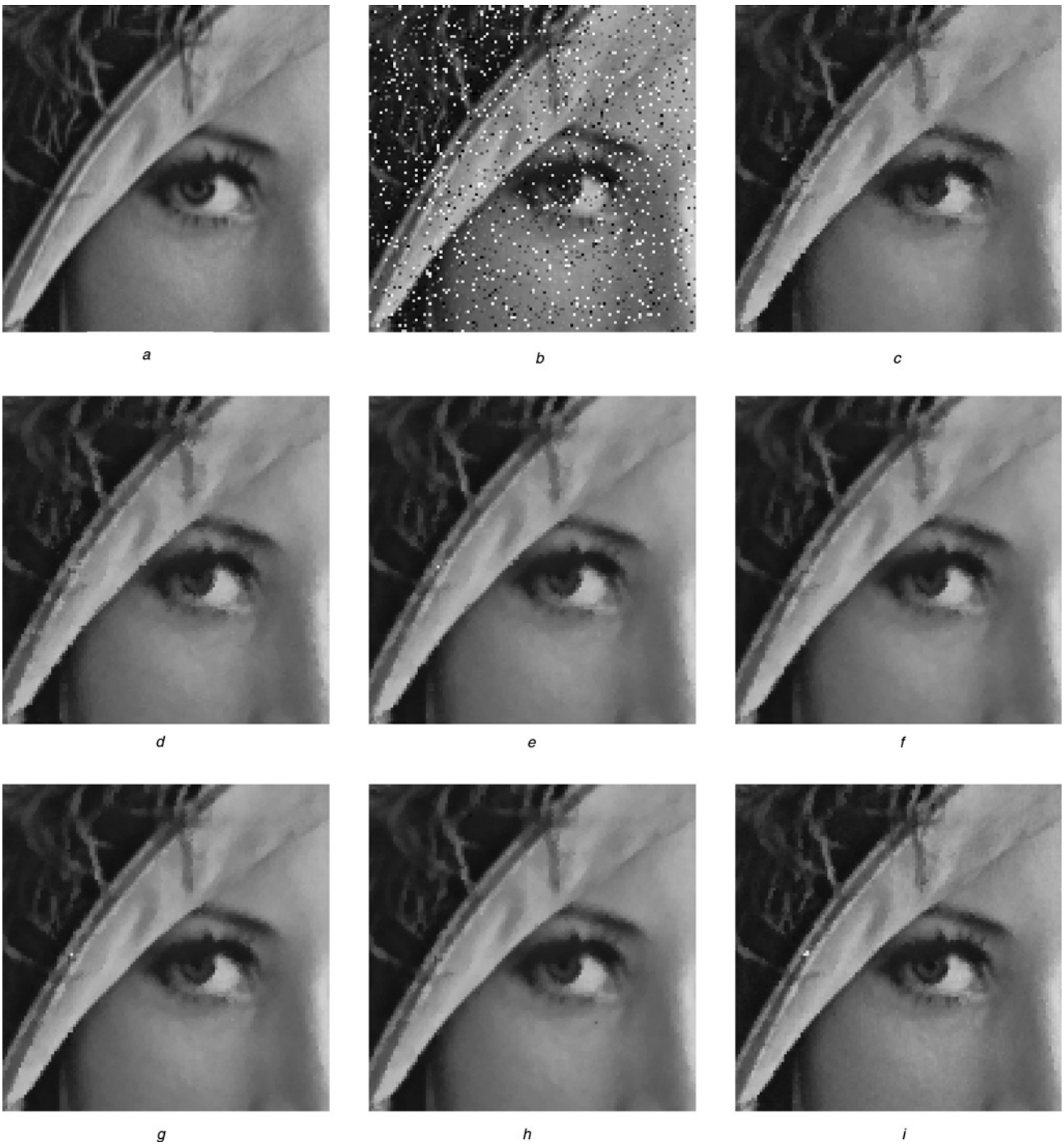

Figure 3 *Filtering results for the Lenna image corrupted with 20% noise*

a Original
b 20% noisy (MAE:12.67, MSE:1960.48)
c d_{cosine} (MAE:4.23, MSE:55.06)
d d_{∞} (MAE:4.01, MSE:48.72)
e d_{goude} (MAE:3.75, MSE:42.51)
f d_2 (MAE:3.70, MSE:40.71)
g s_{fmds} (MAE:3.54, MSE:37.48)
h d_1 (MAE:3.49, MSE:37.72)
i s_{cfs} (MAE:1.26, MSE:23.24)

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

Computational efficiency was measured by CPU time in milliseconds. All of the programs were implemented in the C language, compiled with the gcc 4.2.4 compiler and executed on an Intel®Core^{TM}2 Quad Q6700 2.66 GHz machine. The time measurements were averaged over ten identical runs. Although relative efficiency judgments can

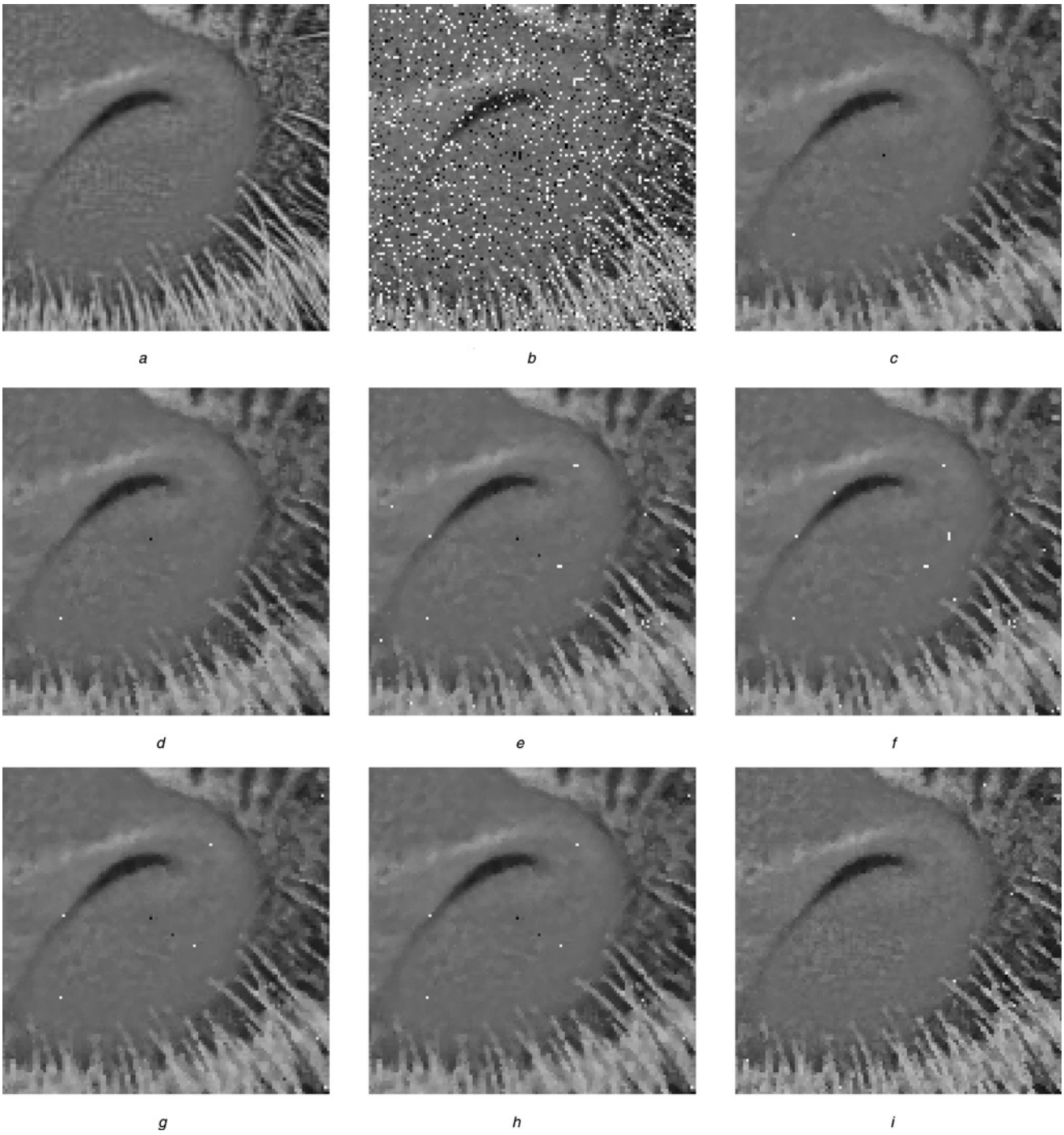

Figure 4 *Filtering results for the Baboon image corrupted with 30% noise*

a Original
b 30% noisy (MAE:19.07, MSE:2894.43)
c d_{chord} (MAE:11.86, MSE:383.68)
d d_2 (MAE:11.76, MSE:375.17)
e d_{ware} (MAE:11.66, MSE:436.98)
f $d_{soergel}$ (MAE:11.61, MSE:410.25)
g d_1 (MAE:11.29, MSE:363.30)
h s_{fms} (MAE:11.22, MSE:361.53)
i s_{cfs} (MAE:7.72, MSE:337.35)

IET Image Process., 2009, Vol. 3, Iss. 5, pp. 249–260
© The Institution of Engineering and Technology 2009
doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

be made using the information given in Table 2, the actual execution times were measured for two main reasons. First, the cost of the basic operations is not uniform (for example, ARCCOS is computationally very expensive when compared to the others). Second, the use of lookup tables (LUTs) can significantly accelerate the computation of certain distance measures.

Eighteen distance measures were evaluated in the experiments. These included the four variants of the Minkowski distance (d_1, d_2, d_∞ and d_2^2), the eight measures listed in Table 1, and the four fuzzy similarity measures discussed in Section 2. Each of these measures was plugged into (1) and the performance of the resulting filter was evaluated using the abovementioned criteria. A square window of size 3×3 ($n = 9$) is used in all of the filters. Although the ordering criterion used in DDF (5) is not a distance measure, because of its widespread use in the literature, it was included in the experiments. Furthermore, the identity filter (indicated by the label 'none'), which performs no filtering was included as a baseline. The K parameters of s_{fms} and s_{fds} were set to 1024 and 4 [11], respectively. The C and t parameters of s_{cfs} were set to 150 and 4 [10], respectively.

Implementation details for the distance measures are given below ($\vec{0} = [0, 0, 0]$):

1. d_2^2: Instead of calculating $n(n-1)/2$ distance values in each neighbourhood $W(r, c)$, first the mean vector $\boldsymbol{x} = 1/n \sum_{i=1}^{n} \boldsymbol{x}_i$ is calculated and then the output vector is determined to be the input vector that has the minimum d_2^2 distance to the mean vector, that is $\boldsymbol{y}(r, c) = \text{argmin}_{\boldsymbol{x}_i \in W(r,c)} d_2^2(\boldsymbol{x}_i, \boldsymbol{x})$ [17].

2. d_{bray}, d_{goude}, d_{soergel}: If $\boldsymbol{x}_i = \boldsymbol{x}_j = \vec{0}$, the distance between the two vectors is taken as 0.

3. d_{canberra}, $d_{\text{divergence}}$, d_{ware}: If $x_{ik} = x_{jk} = 0$ for any component $k \in \{1, 2, 3\}$, the contribution of that component to the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is ignored. The value of the term in the summation is precomputed and stored in a 2D LUT of size 256×256.

4. s_{cfs}: The spatial proximity term is precomputed and stored in a 2D LUT of size $n \times n$.

5. d_{chord}: The value of each term in the summation is precomputed and stored in a 1D LUT of size 256.

6. d_{cosine}: If either $\boldsymbol{x}_i = \vec{0}$ or $\boldsymbol{x}_j = \vec{0}$, the distance between the two vectors is taken as π (maximum possible angular distance). If $\boldsymbol{x}_i = \boldsymbol{x}_j = \vec{0}$, the distance between them is taken as 0 (minimum possible angular distance).

7. s_{fds}: If either $\boldsymbol{x}_i = \vec{0}$ or $\boldsymbol{x}_j = \vec{0}$, the similarity between the two vectors is taken as 0 (minimum possible similarity). If $\boldsymbol{x}_i = \boldsymbol{x}_j = \vec{0}$, the similarity between them is taken as 1 (maximum possible similarity).

8. s_{fms}: The value of the term in the product is precomputed and stored in a 2D LUT of size 256×256.

Table 3 shows the average effectiveness ranking of the distance measures over the entire image set. Note that the ranks start from 0 and the smaller the rank value, the better the corresponding distance measure. For example, with respect to the MAE criterion, the s_{fmds} measure has an average rank of 2.00 at 20% noise level. In other words, with respect to detail preservation, the R-ordering-based vector filter that utilises s_{fmds} ranks, on the average, in top 3 among 18 filters. The last column shows the overall mean effectiveness rank for each distance measure. It can be seen that, on the average, s_{cfs}, s_{fmds} and s_{fms} perform the best. Interestingly, these are all fuzzy metrics, which indicates that fuzzy logic might be better suited for outlier detection when compared to traditional distance measures. Among these metrics, s_{cfs} outperforms the other two by a large margin as evidenced by its impressive 0.33 average rank. The success of s_{cfs} is most likely due to its consideration of spatial proximity, an idea that seems to be overlooked in the literature. It should also be noted that the top two filters are both hybrid in

Table 4 Average efficiency ranking of the distance measures

Measure	Time			Mean
	10%	20%	30%	
none	0.00	0.00	0.00	0.00
d_2^2	1.00	1.00	1.00	1.00
d_1	2.00	2.00	2.00	2.00
d_{ware}	3.48	3.42	3.47	3.46
s_{fms}	3.52	3.58	3.54	3.55
d_{canberra}	5.00	5.00	4.99	5.00
d_∞	6.00	6.01	6.00	6.00
d_{bray}	7.00	6.99	7.00	7.00
d_{chord}	8.61	8.30	8.09	8.33
d_{soergel}	8.41	8.71	8.92	8.68
d_2	10.02	10.01	10.08	10.04
$d_{\text{divergence}}$	10.98	11.00	10.93	10.97
s_{cfs}	12.06	12.05	12.06	12.06
d_{goude}	12.96	12.97	12.96	12.96
s_{fds}	13.97	13.97	13.97	13.97
s_{fmds}	14.99	14.99	14.99	14.99
d_{cosine}	16.00	16.00	16.00	16.00
ddf	17.00	17.00	17.00	17.00

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

Table 5 Time requirements of the distance measures on the Peppers image

Measure	Actual time	Relative time
none	0.000	0T
d_2^2	0.063	1T
d_1	0.153	2T
s_{fms}	0.181	3T
d_{ware}	0.182	3T
$d_{canberra}$	0.183	3T
d_∞	0.240	4T
d_{bray}	0.256	4T
$d_{soergel}$	0.346	6T
d_{chord}	0.356	6T
d_2	0.366	6T
$d_{divergence}$	0.396	6T
s_{cfs}	0.508	8T
d_{goude}	0.533	8T
s_{fds}	0.623	10T
s_{fmds}	0.694	11T
d_{cosine}	1.424	23T
ddf	1.684	27T

nature: s_{cfs} combines magnitude similarity with spatial proximity, whereas s_{fmds} combines magnitude similarity with directional similarity. In addition to the fuzzy metrics, d_1 and $d_{divergence}$ also perform better than d_2, which is the most commonly used distance measure in the literature [1–6]. Despite the fact that it differs from d_2 only by its lack of a square root operation, d_2^2 performs among the worst. This is not surprising given the fact that the R-ordering-based vector filter that utilises d_2^2 essentially approximates the average filter [22], which is known to blur the fine details. d_{cosine} and s_{fds} also perform poorly, which shows that directional information alone is not sufficient to determine the most representative vector in a neighbourhood. Interestingly, the identity filter ('none'), which leaves the noisy image unchanged preserves the fine details better than s_{fds}, d_2^2 and d_{cosine} at 10% noise level.

It should be noted that the performance of the fuzzy metrics can be enhanced by tuning their parameters adaptively according to the image characteristics and the level of noise, see, for example, [10].

Fig. 2 shows sample filtering results for a close-up part of the Peppers image contaminated with 10% noise. It can be seen that d_2^2 blurs the fine details severely. $d_{divergence}$, d_2, d_1 and s_{fms} are fairly close in terms of detail preservation and noise removal performance. On the other hand, s_{cfs} performs significantly better than the others, producing an output image that closely resembles the original image.

Fig. 3 shows sample filtering results for a close-up part of the Lenna image contaminated with 20% noise. It can be seen that d_{cosine} not only fails to remove a significant amount of noise, but also smears the fine details. Even though d_{goude}, d_2, s_{fmds} and d_1 suppress the noise well, this comes at the expense of the blurring of the fine details, for example the boa fur and the eye lashes. In contrast, s_{cfs} achieves a remarkable balance between detail preservation and noise removal.

Fig. 4 shows sample filtering results for a close-up part of the Baboon image contaminated with 30% noise. This image presents a challenging case not only because of its high level of noise, but also because of its complex structural content, for example the whiskers. It can be seen that d_{chord} and d_2 fail to preserve the fine details, producing output images with blurry and/or broken whiskers. d_{ware}, $d_{soergel}$, d_1 and s_{fms} preserve the details better, while leaving many noisy pixels intact. As before, s_{cfs} outperforms the others in terms of both detail preservation and noise removal.

Table 4 shows the average efficiency ranking of the distance measures. Obviously, the identity filter consistently ranks first since it involves no actual filtering. As expected, d_2^2 is the most efficient distance measure since, in contrast to the other measures, it does not involve $n(n-1)/2$ distance evaluations in each neighbourhood. d_1 is the second most efficient measure as it involves only the simplest mathematical operations, that is absolute value and addition/subtraction. The computation of d_1 can be further accelerated using the method described in [36]. d_{ware}, s_{fms} and $d_{canberra}$ are also among the most efficient since they can be calculated rapidly using LUTs. Except for s_{fms}, the fuzzy metrics are not particularly efficient. Among these, s_{cfs} and s_{fmds} partially benefit from the use of LUTs. However, the calculation of the colour similarity term in s_{cfs} and the directional similarity term in s_{fmds} increases the computational requirements for these filters. Finally, d_{cosine} and ddf are the least efficient of all, which is due to their use of the computationally expensive ARCCOS operation.

Table 5 shows the average computational time requirements of the distance measures on the 512 × 512 Peppers image over 100 identical runs. The last two columns give the actual (measured in seconds) and relative (indicated in units of T) computational times for each distance measure. It can be seen that the actual time differences among many of the measures are rather negligible. However, d_{cosine} and ddf are still relatively time consuming when compared to the others.

4 Conclusions and future work

In this paper, we reviewed various distance measures that can be used to order colour vectors in R-ordering-based vector

© The Institution of Engineering and Technology 2009
IET Image Process., 2009, Vol. 3, Iss. 5, pp. 249–260
doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

filters. We evaluated the performance of these measures on a large and diverse set of images using several effectiveness and efficiency criteria. The results demonstrated that the fuzzy similarity metrics s_{cfs}, s_{fmds} and s_{fms} significantly outperform the commonly used d_1 and d_2 metrics in terms of detail preservation, noise removal and colour preservation. Furthermore, these fuzzy metrics are parametrised, which permits the tuning of their parameters according to the image characteristics and the level of noise.

The use of the presented distance measures is not limited to the basic R-ordering-based vector filters such as VMF, BVDF and DDF. Each of these basic filters can in turn be used as a back-end in the implementation of more advanced switching filters, that is those filters that selectively remove the noise by utilising impulse detectors [1, 5, 6]. However, a distance measure that performs well in a non-switching noise removal filter may not necessarily perform as well when combined with an (imperfect) impulse detector. In other words, the complex interaction between the front-end (impulse detector) and back-end (noise removal filter) in a switching filter will likely to have a non-negligible influence on the ultimate performance of a distance measure. This issue will be investigated in a future study.

5 Acknowledgments

This publication was made possible by a grant from The Louisiana Board of Regents (LEQSF2008-11-RD-A-12). The author is grateful to the anonymous reviewers for their insightful suggestions and constructive comments that improved the quality and presentation of this paper.

6 References

[1] SMOLKA B., VENETSANOPOULOS A.N.: 'Color image processing: methods and applications', 'Chapter Noise reduction and edge detection in color images' (CRC Press/Taylor & Francis, 2006), pp. 75–102

[2] PLATANIOTIS K.N., VENETSANOPOULOS A.N.: 'Color image processing and applications' (Springer-Verlag, 2000)

[3] SMOLKA B., PLATANIOTIS K.N., VENETSANOPOULOS A.N.: 'Nonlinear signal and image processing: theory, methods, and applications', 'Chapter Nonlinear techniques for color image processing' (CRC Press, 2004), pp. 445–505

[4] LUKAC R., SMOLKA B., MARTIN K., PLATANIOTIS K.N., VENETSANOPOULOS A.N.: 'Vector filtering for color imaging', *IEEE Signal Process. Mag.*, 2005, **22**, (1), pp. 74–86

[5] LUKAC R., PLATANIOTIS K.N.: 'Advances in imaging & electron physics', 'Chapter A taxonomy of color image filtering and enhancement solutions' (Academic Press, 2006), pp. 187–264

[6] CELEBI M.E., KINGRAVI H.A., ASLANDOGAN Y.A.: 'Nonlinear vector filtering for impulsive noise removal from color images', *J. Electron. Imag.*, 2007, **16**, (3), p. 033008

[7] SHEN Y., BARNER K.E.: 'Fuzzy vector median based surface smoothing', *IEEE Trans. Vis. Comput. Graph.*, 2004, **10**, (3), pp. 252–265

[8] MORILLAS S., GREGORI V., PERIS-FAJARŃES G., LATORRE P.: 'A new vector median filter based on fuzzy metrics', in KAMEL M.S., CAMPILHO A.C. (EDS.): Proc. Second Int. Conf. Image Analysis and Recognition (ICIAR'05), 2005, (*Lecture Notes in Computer Science*, **3656**), pp. 81–90

[9] MORILLAS S., GREGORI V., PERIS-FAJARŃES G., LATORRE P.: 'A fast impulsive noise color image filter using fuzzy metrics', *Real-Time Imag.*, 2005, **11**, (5/6), pp. 417–428

[10] MORILLAS S., GREGORI V., PERIS-FAJARŃES G., SAPENA A.: 'New adaptive vector filter using fuzzy metrics', *J. Electron. Imag.*, 2007, **16**, (3), p. 033007

[11] MORILLAS S., GREGORI V., RIQUELME J., DEFEZ B., PERIS-FAJARŃES G.: 'Fuzzy directional distance vector filter'. Proc. Seventh Int. Workshop on Fuzzy Logic and Applications: Applications of Fuzzy Sets Theory, 2007, pp. 355–361

[12] SCHULTE S., DE WITTE V., NACHTEGAEL M., VAN DER WEKEN D., KERRE E.E.: 'Histogram-based fuzzy colour filter for image restoration', *Image Vis. Comput.*, 2007, **25**, (9), pp. 1377–1390

[13] SCHULTE S., MORILLAS S., GREGORI V., KERRE E.E.: 'A new fuzzy color correlated impulse noise reduction method', *IEEE Trans. Image Process.*, 2007, **16**, (10), pp. 2565–2575

[14] CAMARENA J.-G., GREGORI V., MORILLAS S., SAPENA A.: 'Fast detection and removal of impulsive noise using peer groups', *J. Vis. Commun. Image Represent.*, 2008, **19**, (1), pp. 20–29

[15] MORILLAS S., GREGORI V., PERIS-FAJARŃES G.: 'Isolating impulsive noise pixels in color images by peer group techniques', *Comput. Vis. Image Underst.*, 2008, **110**, (1), pp. 102–116

[16] MORILLAS S., GREGORI V., PERIS-FAJARŃES G., SAPENA A.: 'Local self-adaptive fuzzy filter for impulsive noise removal in color images', *Signal Process.*, 2008, **88**, (2), pp. 390–398

[17] BARNETT V.: 'The ordering of multivariate data', *J. Roy. Statist. Soc. Ser. A*, 1976, **139**, (3), pp. 318–355

[18] TANG K., ASTOLA J., NEUVO Y.: 'Nonlinear multivariate image filtering techniques', *IEEE Trans. Image Process.*, 1995, **4**, (6), pp. 788–798

[19] PITAS I., TSAKALIDES P.: 'Multivariate ordering in color image filtering', *IEEE Trans. Circuits Syst. Video Technol.*, 1991, **1**, (3), pp. 247–259

© The Institution of Engineering and Technology 2009

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.

[20] ASTOLA J., HAAVISTO P., NEUVO Y.: 'Vector median filters', *Proc. IEEE*, 1990, **78**, (4), pp. 678–689

[21] BARNI M., BUTI F., BARTOLINI F., CAPPELLINI V.: 'A quasi-euclidean norm to speed up vector median filtering', *IEEE Trans. Image Process.*, 2000, **9**, (10), pp. 1704–1709

[22] BARNI M., CAPPELLINI V., MECOCCI A.: 'The use of different metrics in vector median filtering'. Proc. Sixth European Signal Processing Conf. (EUSIPCO'92), 1992, pp. 1485–1488

[23] TRAHANIAS P.E., VENETSANOPOULOS A.N.: 'Vector directional filters: a new class of multichannel image processing filters', *IEEE Trans. Image Process.*, 1993, **2**, (4), pp. 528–534

[24] PLATANIOTIS K.N., ANDROUTSOS D., VENETSANOPOULOS A.N.: 'Content-based color image filters', *Electron. Lett.*, 1997, **33**, (3), pp. 202–203

[25] KARAKOS D., TRAHANIAS P.E.: 'Generalized multichannel image filtering structures', *IEEE Trans. Image Process.*, 1997, **6**, (7), pp. 1038–1045

[26] BRAY J.R., CURTIS J.T.: 'An ordination of the upland forest communities of Southern wisconsin', *Ecol. Monogr.*, 1957, **27**, (4), pp. 325–349

[27] LANCE G.N., WILLIAMS W.T.: 'Computer programs for classification'. Proc. Australian National Committee on Computing and Automatic Control Conf., 1966, pp. 304–306

[28] BORG I., GROENEN P.J.F.: 'Modern Multidimensional Scaling: Theory and Applications' (Springer, 2005, 2nd edn.)

[29] CLARK P.J.: 'An extension of the coefficient of divergence for use with multiple characters', *Copeia*, 1952, **1952**, (2), pp. 61–64

[30] GOUDE G.A.: 'A multidimensional scaling approach to the perception of art', *Scand. J. Psychol.*, 1972, **13**, (1), pp. 258–271

[31] LEACH A.R., GILLET V.J.: 'An introduction to chemoinformatics' (Springer, 2003)

[32] WARE G.C., HEDGES A.J.: 'A case for proportional similarity in numerical taxonomy', *J. Gen. Microbiol.*, 1978, **104**, pp. 335–336

[33] KORN G.A., KORN T.M.: 'Mathematical handbook for scientists and engineers: definitions, theorems, and formulas for reference and review' (Dover Publications Inc., 2000, 2nd edn.)

[34] GEORGE A., VEERAMANI P.: 'On some results in fuzzy metric spaces', *Fuzzy Sets Syst.*, 1994, **64**, (3), pp. 395–399

[35] VIERO T., OISTAMO K., NEUVO Y.: 'Three-dimensional median-related filters for color image sequence filtering', *IEEE Trans. Circuits Syst. Video Technol.*, 1994, **4**, (2), pp. 129–142

[36] BARNI M.: 'A fast algorithm for 1-norm vector median filtering', *IEEE Trans. Image Process.*, 1997, **6**, (10), pp. 1452–1455

© The Institution of Engineering and Technology 2009
doi: 10.1049/iet-ipr.2009.0056

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on October 12, 2009 at 01:14 from IEEE Xplore. Restrictions apply.