Fast median filtering for phase or orientation data

Martin Storath, Andreas Weinmann

Abstract—Median filtering is among the most utilized tools for smoothing real-valued data, as it is robust, edge-preserving, value-preserving, and yet can be computed efficiently. For data living on the unit circle, such as phase data or orientation data, a filter with similar properties is desirable. For these data, there is no unique means to define a median; so we discuss various possibilities. The arc distance median turns out to be the only variant which leads to robust, edge-preserving and value-preserving smoothing. However, there are no efficient algorithms for filtering based on the arc distance median. Here, we propose fast algorithms for filtering of signals and images with values on the unit circle based on the arc distance median. For non-quantized data, we develop an algorithm that scales linearly with the filter size. The runtime of our reference implementation is only moderately higher than the Matlab implementation of the classical median filter for real-valued data. For quantized data, we obtain an algorithm of constant complexity w.r.t. the filter size. We demonstrate the performance of our algorithms for real life data sets: phase images from interferometric synthetic aperture radar, planar flow fields from optical flow, and time series of wind directions.

Index Terms—Median filter, circle-median, phase data, orientation data, circle-valued data, manifold-valued data.

1 Introduction

Median filtering is a frequently used basic tool for smoothing signals and images. Its main advantages over linear filtering are the preservation of edges, its robustness to outliers, and the preservation of the values of the data. Yet, the median filter can be computed exactly and efficiently for real-valued data [1, 2]. However, there are many signals and images that do not take their values in the real numbers but on the unit circle $\mathbb{T} = \mathbb{S}^1$. Circle-valued data appear naturally as orientation data as, for example, in the rotation in the bacterial flagellar motor [3] or as wind directions [4]. They appear as phase signals that are defined modulo 2π, as for example in interferometric synthetic aperture radar [5]. Or, they appear when considering the orientation component of planar flow fields as for instance in optical flow [6] or in connection with wind fields [7].

For data with values on the circle, a filter with similar properties as in the real-valued case is desirable. There are various means of defining a circle median, and, thus, a circle median filter. An early definition is a bisecting circle median [8] which generalizes the concept that a median is a data point bisecting the data values into two groups of equal size. Also extrinsic concepts can be used to define a circle median: a median of the ambient vector space is projected to the circle; see for instance [9]. Another way is to generalize the energy minimizing property of the real-valued median to the circle [8, 10]. Then, for a circle-valued image $y \in \mathbb{T}^{M\times N}$, the resulting arc distance median filter is given by

$$u_{mn} = \arg\min_{a\in\mathbb{T}} \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(a, y_{m+i,n+j}), \qquad (1)$$

Martin Storath is with the Image Analysis and Learning Group, Heidelberg Collaboratory for Image Processing, Universität Heidelberg, Germany
Andreas Weinmann is with the Institute of Computational Biology, Helmholtz Zentrum München, and with the Department of Mathematics and Natural Sciences, Hochschule Darmstadt, Germany.

where $d(a, b)$ denotes the (shortest) arc length distance between a and b; see [11]. The arc distance median is a special instance of the general concept of a geometric median for Riemannian manifolds studied by Fletcher et al. [12]. Most importantly, among the above variants, the arc distance median filter is the only concept that shares the desirable properties with the real-valued median filter mentioned above, i.e., preservation of edges, robustness, and the preservation of the values. But as the functional in (1) is not convex it is more involved to compute than the real-valued median filter. Currently, there are no fast algorithms for computing the arc distance median filter.

1.1 Prior and related work

For real-valued data, medians are a very well studied object in robust statistics. For median filtering of real-valued data, there are two types of algorithms: those that work with quantized data and those for non-quantized data. For quantized data, an early method with linear scaling in the side length of the filter mask, i.e., it scales with $\mathcal{O}(R)$ for an $R \times R$ filter mask, has been proposed by Huang et al. [13]. This has been improved to logarithmic complexity [14], and even to constant complexity [2, 15]. For non-quantized real-valued data, Gil and Werman [1] have proposed an algorithm of squared logarithmic complexity, which is close to a theoretical logarithmic lower bound established by the same authors. However, as the algorithm uses quite involved data structures often linearly scaling algorithms are used in practice; an example is the Matlab implementation of the median filter. Median filtering is still an active field of research even for real-valued data; for example, recent advances are fast algorithms for weighted median filters [16, 17].

From a statistical perspective, it is desirable to have a robust estimate of the preferred direction as well. Therefore,

there is quite some statistical literature on circle valued medians. We exemplarily refer to the books [8, 18, 19], the articles [20–27], and to the references therein. There, the notions bisecting median and arc distance median appear. The bisecting median is mostly used in the context of sample medians, whereas the arc distance median is mostly used in the (more theoretical) context of population medians. We could not find a clarification of the relation between the two definitions in the literature. The properties of circle medians in a wider sense are studied in [20]. It has been observed in [11] that the arc distance median filter has edge preserving properties similar to the real-valued median filter, and it has been applied for detection of edges in angular images. Hanbury and Serra [28] have studied further morphological operators such as the morphological gradient and top-hat operator on the unit circle. Circle-valued data can also be processed on the basis of the circle mean value. It is the maximum likelihood estimator for the Von Mises distribution [19] which can be seen as an analogue of the normal distribution on the unit circle. As for real-valued data, mean value filtering is suitable when the noise follows this type of distribution whereas median filtering is preferable for impulsive noise. We refer to [11] for a comparison of circle mean and circle median filters.

Variational models for circle-valued data, in particular total variation (TV) regularization, have been investigated recently in a series of papers [29–37]. These more advanced global energy minimization methods are typically computationally more demanding than local filtering approaches. Although it has has been shown that TV problems can be solved efficiently for 1D signals with values on the unit circle [35], the solvers proposed there unfortunately cannot be generalized to 2D images; the strategies proposed for TV regularization for 2D circle-valued images are computationally more demanding [33, 38].

1.2 Contribution

The goal of this work is to develop efficient algorithms for median filtering of images with values on the unit circle. At first, we clarify the relation between different concepts of defining a median on the circle. In particular, we discuss the relations between the arc distance circle median and the bisecting circle median. We show that the arc distance median filter with odd filter size gives a unique result for almost all input data. The main contributions are fast algorithms for arc distance median filtering: one for quantized data (e.g., data quantized to degrees or minutes) and one for non-quantized data. The complexity of our algorithm for quantized data is constant w.r.t. the size of filter mask. This is optimal, and it matches the complexity of the corresponding median filter for real-valued data [2]. We propose an algorithm that deals with non-quantized data as well. It scales linearly w.r.t. the size of the filter mask. This does not reach the squared logarithmic scaling of the real-valued median filter with the lowest complexity [1], but – as the defining functional is non-convex and the bisecting property is not sufficient for being an arc distance median – we conjecture that this cannot be improved. (We refer to the end of the paper for a discussion.) Our reference implementation is in practice only moderately more expensive than the real-valued median filter algorithm implemented in Matlab. Furthermore, our method is much faster than filtering using the (extrinsic) normalized L_1 median, which is based on the vector median in $\mathbb{R}^2$. Let us also mention that, for small masks and quantized data, the runtime is even lower than the runtime of the proposed algorithm for quantized data. We illustrate the performance of the proposed methods for smoothing of orientation data, phase data, and flow fields. In particular, we consider time series of wind directions, interferometric synthetic aperture radar images, and optical flow images.

1.3 Organization of the paper

In Section 2, we clarify the relation between different means of defining a median on the circle, and we derive important properties of the arc distance median. In Section 3, we derive an algorithm for arc distance median filtering for non-quantized data. In Section 4, we develop an algorithm for quantized data. In Section 5, we drive numerical experiments.

2 Medians on the unit circle

We first discuss different ways of defining a median on the unit circle. Recall that the classical real-valued median $\mu_{\mathbb{R}}$ of real-valued data $y \in \mathbb{R}^N$ has two important characterizing properties: *(i)* a minimality property, i.e., it is the minimizer of the total absolute deviation meaning that $\mu_{\mathbb{R}} \in \arg\min_{a\in\mathbb{R}} \sum_i |a - y_i|$; *(ii)* a bisecting property stating that at most half of the elements of y are above and below $\mu_{\mathbb{R}}$. Remarkably, the analogues of these properties on the unit circle lead to non-equivalent definitions. In the following, we give precise definitions and discuss their relations. Then, we also discuss extrinsic medians, i.e., quantities that are derived from medians of the ambient vector space $\mathbb{R}^2$. Eventually we compare the different definitions.

2.1 Circle-median based on the minimization of the arc distance

We consider signals and images with values on the unit circle, denoted by $\mathbb{T}$ or $\mathbb{S}^1$. If not stated differently, we use the angular representation $a \in (-\pi, \pi]$ for an element of the unit circle, and y denotes data on the unit circle.

A natural metric on the unit circle is given by the arc length distance d which is defined for $a, b \in \mathbb{T}$ by

$$d(a,b) = \begin{cases} |a-b|, & \text{if } |a-b| \le \pi, \\ 2\pi - |a-b|, & \text{else.} \end{cases}$$

This distance just measures the smallest angle between a, b. In analogy to the minimization property of the real-valued median, the median for the unit circle can be defined as

$$\operatorname{med}(y) = \arg\min_{a\in\mathbb{T}} \sum_i d(a, y_i). \tag{2}$$

As in [20], we say that $\mu \in \operatorname{med}(y)$ is an *arc distance (circle) median*.

Defining the median on the unit circle as energy minimizer can be traced back to Mardia [8] in 1972 in the context of population medians; see also [19]. For samples of

spherical data it has been considered by Fisher [10]. Therefore, (2) is also known as Mardia-Fisher median [39]. As mentioned in the introduction, the arc distance median is a specialization of the general concept of a geometric median for general Riemannian manifolds studied by Fletcher et al. [12].

2.2 Circle-median based on a bisecting property

The following definition of a (bisecting) median for N data points on the circle corresponds to the one of a sample median used by Mardia [8, 19]. We denote the antipodal point of a by $\tilde{a}$; that is, $\tilde{a}$ is the uniquely determined point such that $d(\tilde{a}, a) = \pi$. The definition employs the following bisecting property: $\rho \in \mathbb{T}$ is a bisecting point, if more than half of the data lies on either (closed) half-circle/hemisphere defined by ρ and its antipodal point $\tilde{\rho}$. Precisely, using the notation H_ρ^-, H_ρ^+ for the clockwise and the counterclockwise hemisphere, respectively, determined by $\rho, \tilde{\rho}$, we define

$$\rho \in \mathrm{bi}(y) \;:\Leftrightarrow\; |\{y_i \in H_\rho^-\}| \geq \tfrac{N}{2} \text{ and } |\{y_i \in H_\rho^+\}| \geq \tfrac{N}{2}. \tag{3}$$

With a point ρ, its antipodal point $\tilde{\rho}$ is a bisecting point for data y as well. A *bisecting (circle) median* is now the one of the candidates $\rho, \tilde{\rho}$ which has more data points on its side. More precisely, using the notation $S_\mu, S_{\tilde{\mu}}$ for the half circle centered at the points μ and $\tilde{\mu}$,

$$\mu \in \text{b-med}(y) \;:\Leftrightarrow\; \mu \in \mathrm{bi}(y) \text{ and } |\{y_i \in S_\mu\}| \geq |\{y_i \in S_{\tilde{\mu}}\}|. \tag{4}$$

This definition generalizes the characterization of the real-valued median by its bisecting property. The median defined this way is also called Mardia median nowadays. We here use the name bisecting median to avoid possible confusion with the denomination Mardia-Fisher median for (2).

2.3 Relations between the bisecting median and the arc distance circle-median

Since we did not find it in the literature, we here clarify the relation between arc distance circle medians defined by (2) and bisecting circle medians defined by (4). In the following, we illustrate the differences by several simple examples which are collectively visualized in Figure 1.

In contrast to the real-valued case, on the circle, bisecting medians need not be arc distance medians, and vice versa, as the next example shows.

Example 1. We consider the data $y = (-9\pi/16, -9\pi/16, 0, 9\pi/16, 9\pi/16)$. The unique arc distance median is 0. The points $0, -9\pi/16, 9\pi/16$ as well as their antipodal points have the bisecting property (3), and the bisecting medians are $\pi, 7\pi/16, -7\pi/16$. In particular, $\text{b-med}(y) \cap \text{med}(y) = \emptyset$.

The previous example tells us that it might even happen that the bisecting median and the arc distance median sets are disjoint. It also reveals that there might be no bisecting median which is a data point.

On the positive side, we have the following relation.

Proposition 1. *Every arc distance median μ has the bisecting property* (3)*.*

We note that we work rather formal in the following since the intuition might be sometimes misleading when working with circle-valued data. In order to show Proposition 1 we need the following lemma as a preparation. It states that, if a pair of a point and its antipodal is contained in the data, we may remove it without changing the median set. It will be helpful later on as well.

Lemma 2. *Consider data $y_1, \ldots, y_N$ which contains antipodal points y_k and $y_l = \tilde{y_k}$, $k < l$. Then the set of arc distance medians $\text{med}(y_1, \ldots, y_N)$ for data $y_1, \ldots, y_N$ equals the set of arc distance medians $\text{med}(y_1, \ldots, y_{k-1}, y_{k+1}, \ldots, y_{l-1}, , y_{l+1}, \ldots, y_N)$ with the data items y_k and y_l removed from the data.*

Proof. We note that, for all points a, a' on the circle, $d(a, y_k) + d(a, y_l) = d(a', y_k) + d(a', y_l)$. Hence, for all point a, a', we have that $\sum_{i=1}^N d(a, y_i) \leq \sum_{i=1}^N d(a', y_i)$ if and only if $\sum_{i=1, i\neq k,l}^N d(a, y_i) \leq \sum_{i=1, i\neq k,l}^N d(a', y_i)$. Hence μ minimizes the above sum w.r.t. all data items if and only if it minimizes the corresponding sum with the kth and lth summand removed. □

Proof of Proposition 1. We assume that μ is an arc distance median. Without loss of generality, we may, by Lemma 2, assume that no antipodal points of data points are in the data. We denote the number of data points in the (closed) clockwise and counterclockwise hemispheres H_ρ^-, H_ρ^+ by r and l, respectively. Further, we denote the number of data items which equal μ by n and those which equal $\tilde{\mu}$ by m. By our assumption, if the antipodal point $\tilde{\mu}$ is in the data, then μ is not in the data, and the other way round. Hence, one of the two numbers n, m always equals zero. We take a point μ' in one of the hemispheres H_ρ^-, H_ρ^+, which contain more data items, i.e., take $\mu' \in H := H_\rho^-$ if $r \geq l$ and $H := H_\rho^+$ else. Assume further that $\mu' \neq \mu$ is so close to μ that no data item or antipodal point of a data item is contained in the interval between μ' and μ. We now assume that $m \neq 0$. Then $d(\mu', y_i) < d(\mu, y_i)$ for all y_i with $y_i = \tilde{\mu}$. Furthermore, for $y_i \in H$, we have $d(\mu', y_i) = d(\mu, y_i) - d(\mu', \mu)$. For the y_i on the other hemisphere, we have $d(\mu', y_i) = d(\mu, y_i) + d(\mu', \mu)$. Since the number of points in the hemisphere H is at least as large as the number of points on the opposite hemisphere, this implies $f_\mathbb{T}(\mu') < f_\mathbb{T}(\mu)$, where $f_\mathbb{T}(a) = \sum_i d(a, y_i)$. This contradicts μ being a minimizer. Hence, it remains to consider the case $m = 0$. We show that $l \geq N/2$ and that $r \geq N/2$. Without loss of generality, we assume to the contrary that $l < N/2$. As above, we chose $\mu' \in H$ and similarly obtain that $f_\mathbb{T}(\mu') < f_\mathbb{T}(\mu)$ which contradicts μ being a minimizer. Hence, μ has the bisecting property (3). □

It is briefly mentioned in the introduction of [20] which deals with population medians that there is a one to one correspondence between bisecting medians and local minimizers of the mean absolute deviation w.r.t. the considered measure. In our case, the energy is given in (2), i.e.,

$$f_\mathbb{T}(a) = \sum_i d(a, y_i). \tag{5}$$

In our setup, a one to one correspondence between bisecting medians and local minimizers of (5) is not given in general, as the following example shows.

Example 2. We consider the data $-\pi/8, \pi/8, \pi, \pi$ which is of even length. The bisecting median set is given by the union of the intervals $[-7\pi/8, -\pi/2]$, $[-3\pi/8, -\pi/8]$, $[\pi/8, 3\pi/8]$, $[\pi/2, 7\pi/8]$, and the points 0 and π. The global minimizer of (5), i.e., the arc distance median, is π. The local minimizers ρ of (5), in the sense that there is a neighborhood on which $f_{\mathbb{T}}$ is not smaller than $f_{\mathbb{T}}(\rho)$, are given by the open intervals $(-7\pi/8, -\pi/8)$, $(\pi/8, 7\pi/8)$, and π. Hence, there are local minimizers of (5), e.g., $7\pi/16$, which are not bisecting medians. Further, the bisecting median 0 is not a local minimizer of (5); it is even the global maximum of $f_{\mathbb{T}}$.

However, for data of odd size, the set of local minimizers of (5) is characterized as the set of points having the bisecting property (3) and which are contained in the data.

Proposition 3. *Let data $y \in \mathbb{T}^N$ with N odd. The total absolute deviation functional $f_{\mathbb{T}}$ defined by* (5) *attains a local minimum in $\nu \in \mathbb{T}$ if and only if ν both has the bisecting property* (3) *and is contained in the data.*

Proof. By Lemma 2 we may assume that the data does not contain antipodal points. Inspecting the proof of Proposition 1, we see that ν being a local minimizer of (5) implies that ν has the bisecting property and that ν does not agree with an antipodal point of a data point. (Global minimality was not used in this part.) Since data is of odd size, the bisecting property implies that either ν or $\tilde{\nu}$ is contained in the data. Hence, ν is contained in the data.

For the converse direction, assume that ν is contained in the data and that has the bisecting property (3). We consider ν' in a small neighborhood of ν not containing any other data point or its antipodal. Similar to the proof of Proposition 1, we divide the data points y_i into those with $d(\mu', y_i) = d(\mu, y_i) - d(\mu', \mu)$ and those with $d(\mu', y_i) = d(\mu, y_i) + d(\mu', \mu)$. Then the bisecting property (3) implies that the number of data points fulfilling the first equality is smaller than the number of points fulfilling the second inequality. Hence, $f_{\mathbb{T}}(\mu) < f_{\mathbb{T}}(\mu')$ which shows that μ is a local minimizer. □

Concerning the non-uniqueness of the bisecting circle-median we consider the following example.

Example 3. Consider data $-3\pi/8, 0, 2\pi/3$. Then the data points 0 and $-3\pi/8$ as well as the antipodal point of $2\pi/3$ are bisecting medians. The unique arc distance median is 0. If we perturb the data slightly but arbitrarily, the bisecting median sets still consist of three elements. Therefore, the bisecting median remains non-unique on a (Lebesgue) nonzero set.

We record that the bisecting median can be non-unique for a (Lebesgue) nonzero set of data, even for data of odd size. Uniqueness properties of the arc distance median will be investigated in the following subsection.

Summing up this subsection, we have seen that the arc distance median and the bisecting median are quite different. The corresponding median sets can be even disjoint. It can also happen that bisecting medians are not contained in the data. On the positive side, we have seen that every arc distance median has the bisecting property (3), i.e. $\mathrm{med}(y) \subset \mathrm{bi}(y)$, for all circle valued data y (Proposition 1). Furthermore, the bisecting medians which are contained in the data and the local minimizers of the arc distance median defining functional (5) coincide for data of odd size (Proposition 3).

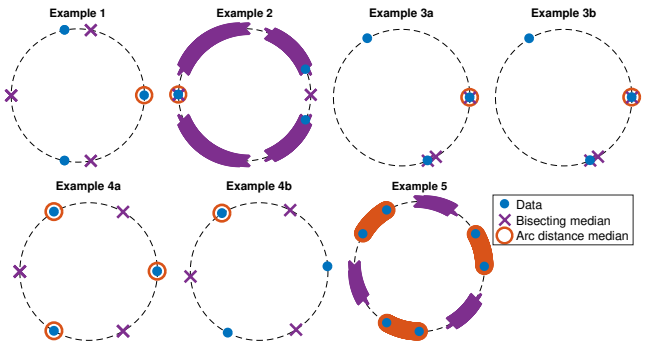


Fig. 1: Visualization of the toy examples 1 to 5 for comparison of the bisecting median and the arc distance median. The variants of example 3 and 4 correspond to exact data (a) and to slightly perturbed data (b).

2.4 Uniqueness properties of the arc distance median

We here discuss the uniqueness properties of the arc distance median. We recall that the real-valued median is unique for data of odd length and need not be unique for data of even length. In contrast, the arc distance median may be non-unique even for an odd number of elements as the following example shows.

Example 4. We consider a "Mercedes-Benz" type configuration $z = (0, 2\pi/3, -2\pi/3)$. Here, all elements of z are arc distance medians.

For an even number of data items, there are situations where every point is a median: e.g. when only pairs of antipodals are contained in the data. For example, for $z = (0, \pi/2, \pi, -\pi/2)$ every point on the circle is a median. For data of even length also rather arbitrary unions of disjoint intervals may appear as the following example shows.

Example 5. We consider a "doubled Mercedes-Benz" type configuration $z = (0, \pi/6, 2\pi/3, 5\pi/6, -2\pi/3, -3\pi/6)$. Here the circle intervals $[0, \pi/6]$, $[2\pi/3, 5\pi/6]$, and $[-3\pi/6, -2\pi/3]$ are the arc distance median sets.

As it is the case for real-valued data, for data of even length, uniqueness is not given even in the almost every setting. This can also be seen at the previous example. Varying the data points slightly but arbitrarily, still at least one interval remains a median interval which results in non-uniqueness on a non-zero set.

On the positive side we have the following result which includes uniqueness guarantees for data of odd size in an almost everywhere setting.

Theorem 4. *For data of odd size, the arc distance median is contained in the data and it is unique for almost all input data y.*

For showing that the median is contained in the data, we need the following preparation.

Lemma 5. *Let y be data with values on the unit circle. Then the data y contains an arc distance median of y.*

	Induced from ambient vector space		Intrinsic definitions	
	Separable median	Normalized L_1 median	Bisecting median	Arc distance median
Preservation of values	no	no	no	yes
Uniqueness (a.e. sense, odd number of elements)	yes	yes	no	yes
Invariant to rotation	no	yes	yes	yes
Exact solver	yes	no	yes	yes

TABLE 1: Comparison of medians for circle-valued data. Only the arc distance median provides uniqueness (in the almost everywhere sense for an odd number of elements), preservation of values, and invariance to rotations. Furthermore, it can be computed using a non-iterative scheme.

This statement is formulated as [22, Lemma 4.1] and proved there.

Lemma 6. *For data of odd length, the set of arc distance medians* $\operatorname{med}(y)$ *is contained in the data* y*. In particular, the median set consists of a finite number of points.*

Proof. An arc distance median is a global minimizer of (5), and therefore also a local minimizer. Hence, by Proposition 3, it is contained in the data. □

Proof of Theorem 4. The uniqueness part of the theorem is a special case of Theorem 7 which we prove in Section 3. By Lemma 6, arc distance medians are data points. □

We note that a statement analogous to Theorem 4 is not true for the bisecting median which may be non-unique for larger ranges of data of odd size (cf. Example 3).

2.5 Quantities induced by medians of the ambient vector space

A simple way to define a median type filter using the ambient vector space is as follows. We regard the unit circle as a subset of $\mathbb{R}^2$, and we apply real-valued median filtering to the components of the data viewed as living in $\mathbb{R}^2$; i.e., we compute vectors (b^*, c^*) of median type by

$$\begin{aligned} b^* &= \operatorname{med}_{\mathbb{R}}(\cos(y_1), \dots, \cos(y_N)), \\ c^* &= \operatorname{med}_{\mathbb{R}}(\sin(y_1), \dots, \sin(y_N)). \end{aligned} \tag{6}$$

The vector (b^*, c^*) is typically not an element of the unit circle, so it has to be renormalized by taking the angular component $a^* = \operatorname{atan2}(b^*, c^*)$. We refer to a^* as *separable median.*

Ducharme and Milasevic [9] have proposed to use *normalized L_1 medians*. Here, one considers the values on the unit circle as points in $\mathbb{R}^2$ and computes the classical vector median in $\mathbb{R}^2$,

$$(b^*, c^*) = \arg \min_{(b,c)\in\mathbb{R}^2} \sum_i \|(b, c) - (\cos(y_i), \sin(y_i))\|_2. \tag{7}$$

For the vector median in two dimensions, no closed form is available, and a standard method for its computation is the iterative Weiszfeld algorithm [40]. The vector median (7) is unique unless all points lie on a line, see [39]. As before, the result (b^*, c^*) is usually not an element of the unit circle. So it is projected to the unit circle by $a^* = \operatorname{atan2}(b^*, c^*)$ as well.

For both ambient space definitions considered, it can happen that $(b^*, c^*) = 0$. Then the closest point projection on the unit circle is not unique, and we might choose an arbitrary point on the circle. However this happens only for data of measure zero. So we have uniqueness in an almost everywhere sense, as for the arc distance median.

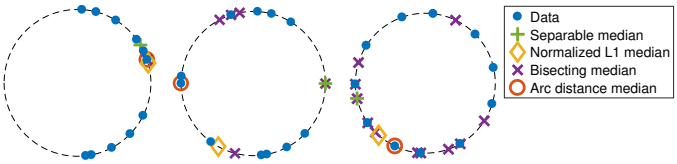


Fig. 2: Illustration of different notions of a median on the unit circle. If all points lie on one hemisphere, the bisecting median coincides with the arc distance median (left). In general, there are many possibilities for the bisecting median, and they are not necessarily data points (center, right). In all configurations, the medians induced by the ambient space $\mathbb{R}^2$ – separable and normalized L_1 median – do not coincide with a data point (or its antipodal point). The arc distance median is unique (almost surely) and it is contained in the data.

2.6 Comparison

We compare the properties of the different notions of medians on the unit circle. It is the main advantage of the definitions based on the ambient space – separable median and normalized L_1 median – that the algorithms for the vector space setting can be reused. On the flipside, the characteristic property that the median is contained in the data gets lost. It is a further disadvantage of the normalized L_1 median that its computation requires an iterative procedure. As the separable median minimizes a Manhattan distance, it is not rotation invariant. Compared with the normalized L_1 median of the ambient space, the bisecting median has the advantage that it can be computed exactly as it can be done for the separable median. Compared with the bisecting median, the arc distance median has the advantage that it preserves the values, i.e., there is always a median which is a data point. Further, it has stronger uniqueness properties.

Summing up, as only concept, the arc distance median satisfies the desirable properties of preservation of values, of uniqueness in an almost everywhere sense and of allowing for an exact solution. A summary relating the different concepts of a median and the desired properties is given in Table 1. Figure 2 illustrates the discussed quantities for some typical data configurations.

3 Arc Distance Median Filtering

Now we deal with filtering based on the arc distance median. We consider a circle-valued image $y \in \mathbb{T}^{M\times N}$, where $M, N \geq 1$. We always assume mirror boundary conditions, that is, for $n < 0$

$$y_{mn} = \begin{cases} y_{-m-1,-n-1}, & \text{if } m \leq 0, \\ y_{2M-m-1,-n-1}, & \text{if } m \geq M, \end{cases}$$

and analogously for $n > N-1$ by $y_{mn} = y_{-m-1,2N-n-1}$, if $m \leq 0$, and by $y_{mn} = y_{2M-m-1,2N-n-1}$, if $m \geq M$.

At first, we prove a uniqueness statement for the filter output.

Theorem 7. *For almost all signals or images, the output of the arc distance median filter* (1) *with a mask of odd size is unique.*

Proof. We first consider data y with the property that the data item y_{ij} does not coincide with another data item $y_{kl}, (k,l) \neq (i,j)$, or its antipodal point $\tilde{y}_{kl}, (k,l) \neq (i,j)$, i.e., $y_{ij} \notin V_{ij}$, for all i,j, where

$$V_{ij} = \{z \in \mathbb{T} : z = y_{kl} \text{ or } z = \tilde{y}_{kl}, \\ (0,0) \leq (k,l) \leq (N-1, M-1), (k,l) \neq (i,j)\}.$$

We gather the data with this property in the set P, i.e.,

$$P = \{y \in \mathbb{T}^{N \times M} : y_{ij} \notin V_{ij} \text{ for all } (i,j)\},$$

and start out to show that it is enough to prove the assertion for data in an arbitrary neighborhood of $y \in P$.

To this end, we consider, for $\delta > 0$, the sets P_δ defined by

$$P_\delta := \{y \in \mathbb{T}^{N \times M} : d(y_{ij}, V_{ij}) \geq \delta \text{ for all } (i,j)\}.$$

We note that the elements of the sets P_δ are well separated. We further notice the important fact that P_δ is compact. Hence it allows for a finite subcovering for every initial covering. We have that $P = \bigcup_{n \in \mathbb{N}} P_{1/n}$. We note that the complement of P is a Lebesgue zero set. Hence it is enough to show the assertion of the theorem for P. Since P is the countable union of the sets $P_{1/n}$, $n \in \mathbb{N}$, it is sufficient to show the assertion of the theorem for $P_\delta, \delta > 0$. Let us assume that, for every $y \in P$, the statement holds in a neighborhood U_y of y. Then we can cover P_δ with the sets U_y, $y \in P_\delta$, for every $\delta > 0$. By the compactness of P_δ, there are finitely many neighborhoods U_y covering P_δ, and in each such neighborhood the minimizer is a.e. unique. Therefore, the minimizer is a.e. unique in P_δ. Summing up, it is enough to show the assertion of the theorem in a neighborhood of U_y for $y \in P$.

As a next step, we see that it is enough to show the statement pointwise, i.e., the response of the circle median filter is a.e. unique at the pixel (k,l) for data in a neighborhood of $y \in P$. If we assume to the contrary that the filter output is non-unique on a Lebesgue nonzero set in a neighborhood of $y \in P$, then there is a pixel (k,l) such that the filter output at (k,l) is non-unique on a Lebesgue nonzero set in a neighborhood of $y \in P$. Hence, almost everywhere-uniqueness of the filter at a pixel (k,l) implies the corresponding statement for the whole filter response. Hence, for notational convenience, we consider the $(2r+1)(2t+1)$ data items of y which contribute to the values of the median filter at pixel (k,l), and rearrange them as the vector $(y_1, \ldots, y_N)$. Summing up, we have data $(y_1, \ldots, y_N)$, and we have to show that the circle median is a.e. unique for $y \in P$.

To this end, we have to unwrap the circle w.r.t. a base point b on the circle, i.e., we consider the mapping $\pi_b : \mathbb{T} \to (-\pi, \pi])$ which maps b to 0, x to $d(x,b)$ if x is on the counterclockwise hemisphere w.r.t. b and x to $-d(x,b)$ else. We need the function

$$f(a, y_1, \ldots, y_N) = \sum_{i=1}^{N} d(a, y_i). \tag{8}$$

By our definition of P, for every $b \in V$, there are sufficiently small neighborhoods U_y of the data y and of b, such that

$$f(a, y'_1, \ldots, y'_N) = \sum_{i=1}^{N} |\pi_b(a) - \pi_b(y'_i)| \tag{9}$$

for all y' in the neighborhood U_y of y, and all a in the neighborhood of b.

By Lemma 6, there is an arc distance median contained in the data y. Let v be such an arc distance median and consider v as a basepoint for unwrapping. For data y, $\pi_v(v)$ is a minimizer of the right-hand side of (9) w.r.t. data $\pi_b(y_i)$. Since the minimizer of the right-hand side of (9) (w.r.t. the first variable for fixed data $\pi_b(y'_i)$) is unique, we get a uniquely defined element $v(y')$ such that $f(v(y'), y'_1, \ldots, y'_N) > f(a, y'_1, \ldots, y'_N)$ for all $a \neq v(y')$ in a neighborhood N_v of v, and all $y' \in U_y$. We note that $v(y) = v$ and that, in particular, v is locally a unique minimizer. Also $v(y')$ is locally a unique minimizer. Furthermore, since f is continuous and $\mathbb{T}$ is compact there is a neighborhood U'_y of y such that, for all $y' \in U'_y$, the minimizers of $f(\cdot, y'_1, \ldots, y'_N)$ are located in the above neighborhoods N_v, where v is a minimizer w.r.t. data y. Then, for every y' in U'_y, we consider the number $n(y')$ of different minimizers for data y'. We have to show that $n(\cdot) = 1$ besides a Lebesgue zero-set in U'_y. We assume to the contrary that $n(y') > 1$ on a set $B \subset U'_y$ of positive Lebesgue measure. Then there is an open set $W \subset B$ and circle medians $v'(y')$ and $v''(y')$ for data $y' \in B$ such that v', v'' are continuous functions of y', such that $v'(y') \neq v''(y')$ for all $y' \in B$, and such that $f(v'(y'), y'_1, \ldots, y'_N) = f(v''(y'), y'_1, \ldots, y'_N)$ for all $y' \in W$. Now we use the right hand side of (9) to conclude the differentiability of the mappings $h(y') := f(v''(y'), y'_1, \ldots, y'_N)$: in fact, since we may assume that all data items $\pi_b(y'_j)$ are different (because y is assumed to be in P), the right hand side of (9) with $\pi_b(v'(y'))$ and $\pi_b(v''(y'))$ as functions of y' as first argument is differentiable w.r.t. the variable y'. Its partial derivative with respect to an arbitrary variable is given by either $1, -1$, or 0. Here 0 appears precisely once: we obtain zero when taking the derivative w.r.t. the variable which represents the median. Further, the number p_i of $+1$'s and the number n_i of -1's on the interval I_i are equal. For the real valued right hand side of (9), a partial derivative of $+1$ is equivalent to the corresponding data item being greater or equal to the median; a partial derivative of -1 is equivalent to the corresponding data item being smaller or equal to the median. Now we consider fixed data $y'' \in W$, and the corresponding different medians $v'(y'') \neq v''(y'')$. Since they are different, there is a data item y''_i such that $\pi_{v'(y'')}(y''_i) > 0$ and $\pi_{v''(y'')}(y''_i) < 0$, or reversed, i.e., $\pi_{v'(y'')}(y''_i) < 0$ and $\pi_{v''(y'')}(y''_i) > 0$. (Note that the median equals 0 since $\pi_z(z) = 0$, for all z.) In any case, the derivative of the right-hand side of (9) w.r.t $\pi_{v''(y'')}(y''_i)$ does not equal the derivative of the right-hand side of (9) w.r.t $\pi_{v'(y'')}(y''_i)$. This contradicts $f(v'(y'), y'_1, \ldots, y'_N) = f(v''(y'), y'_1, \ldots, y'_N)$ for all $y' \in W$. Hence, the number of different arc distance medians $n(\cdot) = 1$ almost everywhere in a neighborhood of data y, which we saw in the beginning, is enough to conclude the assertion of the theorem. □

Next we turn to algorithms for computing the arc distance median filter (1). For simplicity, we focus on filter

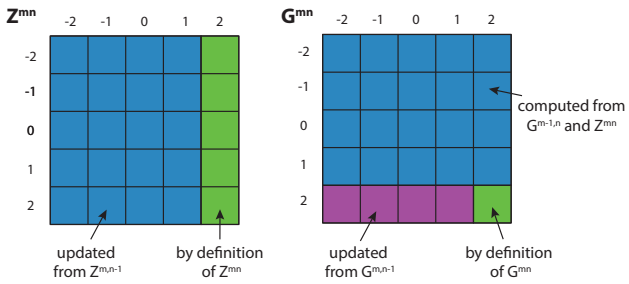


Fig. 3: Graphical illustration of the two-stage recurrence scheme at pixel (m, n) for a 5×5 filter mask.

masks of odd side lengths $R = 2r + 1$ and $T = 2t + 1$. (The modifications for even filter sizes are straightforward, but note that uniqueness gets lost.) We start with the following simple observation: Since an arc distance median is contained in the data (cf. Lemma 5), it is enough to search a minimizer among the data points in the mask; that is, we have to determine the sums in (1) for a being data points only. Note that a simple "brute-force" approach, i.e. evaluating (1) directly for all items in the mask, would lead to an $\mathcal{O}(R^2T^2)$ algorithm. An algorithm of complexity $\mathcal{O}(RT(\log R + \log T))$ can be devised by utilizing dynamically updated sorting of the elements in the filter mask. On the flipside, that approach requires a self-balancing binary search tree (e.g., a red-black tree) as underlying data structure which makes implementation more involved and leads to computational overhead. The algorithm we propose here has lower complexity and does not require dynamic data structures. The crucial point of the proposed approach is the observation that there are several recurrence relations between adjacent rows and columns which we may exploit to compute the sums in (1) efficiently. More specifically, if we perform rowwise filtering, we can compute the deviations for all elements in the filter mask as follows. The deviations with respect to the values in the first $R - 1$ rows of the mask obey a recurrence relation which depends on the last processed row and on a type of accumulator. The accumulator in turn satisfies another recurrence in row-direction. Then, for the first $T - 1$ elements in the last row of the mask, we use yet another recurrence in horizontal direction. The remaining element in the mask (Rth row, Tth column) is computed directly using the sum in (1). The detailed procedure is provided as pseudo-code in Algorithm 1. Next we prove its correctness along with the derivation of the recurrence relations. A graphical illustration of the computational scheme is given in Figure 3.

Theorem 8. *Algorithm 1 computes the arc distance median filter* (1) *of mask size $R \times T$ for non-quantized circle-valued data y in $\mathcal{O}(RT)$ time per element.*

Proof. For integers k, l, m, n with $-r \le k \le r$, $-t \le l \le t$, $0 \le m \le M - 1$, and $0 \le n \le N - 1$, we consider the quantities

$$G_{kl}^{mn} = \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(y_{m+k,n+l}; y_{m+i,n+j}). \tag{10}$$

We recall that we are looking for minimizing arguments k^*, l^* of G_{kl}^{mn}, since these minimizing arguments define the

Algorithm 1: Arc distance median filter for non-quantized data

Input: Data $y \in \mathbb{T}^{M \times N}$; filter mask side lengths $R = 2r + 1$ and $T = 2t + 1$, with $r, t \in \mathbb{N}_0$
Output: Arc distance median filtered data $u \in \mathbb{T}^{M \times N}$

```
/* Process first row                                                    */
for k ← −r to r do          // process first element (n = 0, m = 0)
    for l ← −t to t do
        G^{00}_{kl} ← Σ_{i=−r}^{r} Σ_{j=−t}^{t} d(y_{k,l}; y_{i,j});
    end
end
for n ← 1 to N − 1 do          // process rest of first row (m = 0)
    for k ← −r to r do
        for l ← −t to t − 1 do
            G^{0n}_{kl} ← G^{0,n−1}_{k,l+1} + Σ_{i=−r}^{r} d(y_{k,n+l}; y_{i,n+t})
                        − d(y_{k,n+l}; y_{i,n−t−1});
        end
        G^{0n}_{kt} = Σ_{i=−r}^{r} Σ_{j=−t}^{t} d(y_{k,n+t}; y_{i,n+j});
    end
end
Fill u_{0n} for n = 0, ..., N − 1 using equation (11);
/* Process rows m = 1, ..., M − 1                                       */
for m ← 1 to M − 1 do
    for k ← −r to r do                          // init auxiliary array Z
        for l ← −t to t do
            Z^{mn}_{kl} ← Σ_{j=−t}^{t} d(y_{m+k,n+l}; y_{m+r,n+j})
                        − d(y_{m+k,n+l}; y_{m−r−1,n+j});
        end
    end
    for l ← −t to t do                  // compute first element G^{m0}
        for k ← −r to r − 1 do
            G^{m0}_{kl} ← G^{m−1,0}_{k+1,l} + Z^{m0}_{kl};
        end
        G^{m0}_{rl} ← Σ_{i=−r}^{r} Σ_{j=−t}^{t} d(y_{m+r,l}; y_{m+i,j});
    end
    for n ← 1 to N − 1 do
        for k ← −r to r do                                  // update Z
            for l ← −t to t − 1 do
                Z^{mn}_{kl} ← −d(y_{m+k,n+l}; y_{m+r,n−t−1})
                            + d(y_{m+k,n+l}; y_{m−r−1,n−t−1})
                            + d(y_{m+k,n+l}; y_{m+r,n+t})
                            − d(y_{m+k,n+l}; y_{m−r−1,n+t}) + Z^{m,n−1}_{k,l+1};
            end
            Z^{mn}_{kt} ← Σ_{j=−t}^{t} d(y_{m+k,n+t}; y_{m+r,n+j})
                        − d(y_{m+k,n+t}; y_{m−r−1,n+j});
        end
        for l ← −t to t do          // compute first 2r rows of G^{mn}
            for k ← −r to r − 1 do
                G^{mn}_{kl} ← G^{m−1,n}_{k+1,l} + Z^{mn}_{kl};
            end
        end
        for l ← −t to t − 1 do          // compute last row of G^{mn}
            G^{mn}_{rl} ← Σ_{i=−r}^{r} d(y_{m+k,n+l}; y_{m+i,n+t})
                        − d(y_{m+k,n+l}; y_{m+i,n−t−1}) + G^{m,n−1}_{r,l+1};
        end
        G^{mn}_{rt} ← Σ_{i=r}^{r} Σ_{j=−t}^{t} d(y_{m+k,n+l}; y_{m+i,n+j});
    end
    Fill u_{mn} for n = 0, ..., N − 1 using equation (11);
end
return u;
```

arc distance median filter by

$$u_{mn} = y_{m+k^*,n+l^*}. \tag{11}$$

We first show the correctness of Algorithm 1 by showing the following recurrence formulae. We start out letting $m = 0$. For $n = 1, ..., N-1$, $k = -r, ..., r$, and $l = -t, ..., t-1$, we get that the values G_{kl}^{0n} fulfill the recurrence relation

$$\begin{aligned}
G_{kl}^{0n} &= \sum_{i=-r}^{r} \Big\{ d(y_{k,n+l}; y_{i,n+t}) - d(y_{k,n+l}; y_{i,n-t-1}) \\
&\quad + \sum_{j=-t-1}^{t-1} d(y_{k,n+l}; y_{i,n+j}) \Big\} \\
&= \sum_{i=-r}^{r} d(y_{k,n+l}; y_{i,n+t}) - d(y_{k,n+l}; y_{i,n-t-1}) \\
&\quad + \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(y_{k,n-1+l+1}; y_{i,n+j-1}) \\
&= \sum_{i=-r}^{r} d(y_{k,n+l}; y_{i,n+t}) - d(y_{k,n+l}; y_{i,n-t-1}) + G_{k,l+1}^{0,n-1}.
\end{aligned}$$

For $l = t$, we compute G_{kt}^{0n} simply by $G_{kt}^{0n} = \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(y_{k,n+t}; y_{i,n+j})$, for $k = -r, ..., r$.

Next, we consider general row indices $m = 1, ..., M-1$. For $k = -r, \ldots, r-1$, $l = -t, \ldots, t$ and $n = 1, ..., N-1$, we may express G_{kl}^{mn} by

$$\begin{aligned}
G_{kl}^{mn} &= \sum_{i=-r-1}^{r-1} \sum_{j=-t}^{t} d(y_{m+k,n+l}; y_{m+i,n+j}) \\
&\quad + \sum_{j=-t}^{t} d(y_{m+k,n+l}; y_{m+r,n+j}) \\
&\quad - \sum_{j=-t}^{t} d(y_{m+k,n+l}; y_{m-r-1,n+j}) \\
&= \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(y_{m-1+k+1,n+l}; y_{m+i-1,n+j}) + Z_{kl}^{mn} \\
&= G_{k+1,l}^{m-1,n} + Z_{kl}^{mn},
\end{aligned}$$

where the auxiliary array Z_{kl}^{mn} is defined by

$$\begin{aligned}
Z_{kl}^{mn} &= \sum_{j=-t}^{t} d(y_{m+k,n+l}; y_{m+r,n+j}) \\
&\quad - d(y_{m+k,n+l}; y_{m-r-1,n+j}).
\end{aligned}$$

Also, the auxiliary array Z fulfills a recurrence relation: for $k = -r, \ldots, r$ and $l = -t, \ldots, t-1$, $n = 1, ...N-1$, and $m = 1, ..., M-1$, we get that

$$\begin{aligned}
Z_{kl}^{mn} &= - d(y_{m+k,n+l}; y_{m+r,n-t-1}) \\
&\quad + d(y_{m+k,n+l}; y_{m-r-1,n-t-1}) \\
&\quad + d(y_{m+k,n+l}; y_{m+r,n+t}) \\
&\quad - d(y_{m+k,n+l}; y_{m-r-1,n+t}) \\
&\quad + \sum_{j=-t-1}^{t-1} d(y_{m+k,n+l}; y_{m+r,n+j}) \\
&\quad - d(y_{m+k,n+l}; y_{m-r-1,n+j}) \\
&= - d(y_{m+k,n+l}; y_{m+r,n-t-1}) \\
&\quad + d(y_{m+k,n+l}; y_{m-r-1,n-t-1}) \\
&\quad + d(y_{m+k,n+l}; y_{m+r,n+t}) \\
&\quad - d(y_{m+k,n+l}; y_{m-r-1,n+t}) + Z_{k,l+1}^{m,n-1}.
\end{aligned}$$

Using a similar argument, we get for $k = r$ and $l = -t, \ldots, t-1$, that the G_{kl}^{mn} are related via

$$\begin{aligned}
G_{kl}^{mn} &= G_{k,l+1}^{m,n-1} + \sum_{i=-r}^{r} d(y_{m+k,n+l}; y_{m+i,n+t}) \\
&\quad - d(y_{m+k,n+l}; y_{m+i,n-t-1}).
\end{aligned}$$

We further observe that we can discard the tables $G^{m-1,n}$ for $n = 0, \ldots, N-1$ when processing G^{mn}, and the same is true for Z^{mn}. This shows the correctness of the algorithm. Furthermore, we see that the extra memory consumption is in the order of N arrays of size $R \times T$.

It remains to prove the assertion on the time complexity. We first note that (11) has linear complexity in the size of the filter mask RT. For the computation of G^{mn} we need at most four nested for-loops, which are – since $R \leq M$ and $T \leq N$ – all in $\mathcal{O}(RTMN)$. We obtain the overall complexity $\mathcal{O}(RTMN)$, or $\mathcal{O}(RT)$ per element. □

4 A CONSTANT TIME ARC DISTANCE MEDIAN FILTER FOR QUANTIZED DATA

Next we derive a fast method for arc distance median filtering for quantized data. That is, the values of y are contained in a finite set $Q = \{q_1, \ldots, q_S\} \subset \mathbb{T}$, where $S \in \mathbb{N}$. In contrast to Algorithm 1, where we computed the deviations G_{kl}^{mn} for the data items in the mask, we now consider the set of possible data values Q as a candidate set. We derive recurrence relations for the corresponding deviations F^{mn} given by

$$F^{mn}(q) = \sum_{i=-r}^{r} \sum_{j=-t}^{t} d(q; y_{m+i,n+j}), \tag{12}$$

for $0 \leq m \leq M-1$, $0 \leq n \leq N-1$, and $q \in \mathbb{T}$, which are key for a fast computation. The algorithm is given as pseudo-code in Algorithm 2. We next state a theorem on the correctness and the complexity of the algorithm.

Theorem 9. *Let $Q \subset \mathbb{T}$ be a finite set. Algorithm 2 computes the arc distance median filter of size $R \times T$ for data y with values in Q in $\mathcal{O}(1)$ time per element.*

Proof. By Lemma 5, the search space for an arc distance median can be reduced to the set of values which, in the quantized situation, is contained in the set Q. Hence, we have

$$u_{mn} = \arg\min_{q \in Q} F^{mn}(q). \tag{13}$$

We compute the tabulations $F^{mn}(q)$ for all $q \in Q$. We start out with $m = 0$ and obtain the recurrence relation

$$\begin{aligned}
F^{0n}(q) &= \sum_{i=-r}^{r} \Big\{ \sum_{j=-t-1}^{t-1} d(q; y_{m+i,n+j}) \\
&\quad + d(q; y_{m+i,n+t}) - d(q; y_{m+i,n-t-1}) \Big\} \\
&= \sum_{i=-r}^{r} \Big\{ \sum_{j=-t}^{t} d(q; y_{m+i,n+j-1}) \\
&\quad + d(q; y_{m+i,n+t}) - d(q; y_{m+i,n-t-1}) \Big\} \\
&= \sum_{i=-r}^{r} d(q; y_{m+i,n+t}) - d(q; y_{m+i,n-t-1}) \\
&\quad + F^{0,n-1}(q).
\end{aligned}$$

Next we consider the remaining rows $m = 1, \dots, M-1$, and define the auxiliary quantity Z by

$$Z^{mn}(q) = \sum_{j=-t}^{t} d(q; y_{m+r,n+j}) - d(q; y_{m-r-1,n+j}).$$

For $n = 1, \dots, N-1$, we have that the Z^{mn} fulfill the relation

$$\begin{aligned} Z^{mn}(q) =& Z^{m,n-1}(q) + d(q; y_{m+r,n+t}) - d(q; y_{m-r-1,n+t}) \\ & - d(q; y_{m+r,n-t-1}) + d(q; y_{m-r-1,n-t-1}). \end{aligned}$$

It follows that, for all $m = 1, ..., M-1$ and $n = 0, \dots, N-1$, the F^{mn} are related via the Z^{mn} by the recurrence relation

$$\begin{aligned} F^{mn}(q) &= \sum_{j=-t}^{t} \Big\{ \sum_{i=-r-1}^{r-1} d(q; y_{m+i,n+j}) \\ &\quad + d(q; y_{m+r,n+j}) - d(q; y_{m-r-1,n+j}) \Big\} \\ &= \sum_{j=-t}^{t} \Big\{ \sum_{i=-r}^{r} d(q; y_{m+i-1,n+j}) \\ &\quad + d(q; y_{m+r,n+j}) - d(q; y_{m-r-1,n+j}) \Big\} \\ &= Z^{mn}(q) + F^{m-1,n}(q). \end{aligned}$$

Next we observe that we may overwrite the values of $F^{m-1,n}$ and $Z^{m-1,n}$ when processing the mth row. Furthermore, only the value of $Z^{m,n-1}$ is required when processing the element at m, n. This proves the correctness of Algorithm 2.

Regarding the computational costs, we first note that computing (13) has constant complexity in the size of the filter mask. As Q is fixed, the computational effort for the first elements of Z is $\mathcal{O}(MT)$, and for the other elements of Z is $\mathcal{O}(MN)$. Likewise, computations for F^{00}, F^{0n}, and F^{mn} are in $\mathcal{O}(RT)$, $\mathcal{O}(RN)$, and $\mathcal{O}(MN)$, respectively. As $R \leq M$ and $T \leq N$, the overall complexity is $\mathcal{O}(MN)$, or $\mathcal{O}(1)$ per element. □

Algorithm 2: Arc distance median filter for quantized data

Input: Data $y \in Q^{M \times N}$ with $Q \subset \mathbb{T}$; filter mask side lengths $R = 2r+1$ and $T = 2t+1$ with $r, t \in \mathbb{N}$
Output: Arc distance median filtered image $u \in Q^{M \times N}$

```
/* Process first row                                                    */
for q ∈ Q do
    F^0(q) ← Σ_{i=-r}^{r} Σ_{j=-t}^{t} d(q; y_{m+i,n+j});
    for n ← 1 to N − 1 do
        F^n(q) ← F^{n−1}(q) + Σ_{i=-r}^{r} d(q; y_{i,n+t}) − d(q; y_{i,n−t−1});
    end
end
for n ← 0 to N − 1 do
    u_{0n} ← arg min_q F^n(q);
end
/* Process rows m = 1, ..., M − 1                                       */
for m ← 1 to M − 1 do
    for q ∈ Q do
        Z(q) ← Σ_{j=-t}^{t} d(q; y_{m+r,j}) − d(q; y_{m−r−1,j}) ;
        F^0(q) ← Z(q) + F^0(q);
        for n ← 1 to N − 1 do
            Z(q) ← d(q; y_{m+r,n+t}) − d(q; y_{m−r−1,n+t})
            − d(q; y_{m+r,n−t−1}) + d(q; y_{m−r−1,n−t−1}) + Z(q);
            F^n(q) ← Z(q) + F^n(q);
        end
    end
    for n ← 0 to N − 1 do
        u_{mn} ← arg min_q F^n(q);
    end
end
```

5 Numerical results

We have implemented the proposed algorithms in C++ with Matlab interface using Mex files.[1] We compare our method with filters based on the separable median and on the normalized L_1 median. For the first one we use the function `medfilt2` shipped with Matlab 2016b. For the latter, we have implemented the Weiszfeld algorithm in C++; see [40]. We use the starting value $(0, 0)$ and the stopping criterion $\|\mu^k - \mu^{k+1}\|_1 \leq \tau \|\mu^{k+1}\|_1$ with $\tau = 10^{-9}$ as proposed in [41]. We also limit the maximum number of iterations to 100. All experiments were conducted on a laptop with 2.2 GHz Intel Core i7 and 16 GB RAM. We first compare the denoising performance and the runtimes of the different circle median filters on synthetic data. Then we apply the filters for smoothing real data.

1. Reference implementation provided at https://github.com/mstorath/CircleMedianFilter

5.1 Experimental results on synthetic data

First, we compare the denoising performance of the filters on synthetic data. To our knowledge, there is no established denoising benchmark for circle-valued images. To create a benchmark data set with diversified content, we utilize the hue values of the 200 training images of the Berkeley Segmentation Dataset BSDS500 [42, 43]. We corrupted these circle-valued images with impulsive type noise: we randomly chose a set of pixels (according to a given percentage), and then let those pixels be i.i.d. with respect to the uniform distribution on the unit circle. As in [11], we use the circular noise reduction index

$$\mathrm{NRI_c} = 10 \log_{10} \left(\frac{\sum_{ij} d(g_{ij}, y_{ij})}{\sum_{ij} d(g_{ij}, u_{ij})} \right) \mathrm{db}$$

for the quantitative evaluation. Here g denotes the groundtruth, y the data, and u the filtering result. A high value of $\mathrm{NRI_c}$ is desirable. The results for different noise levels (i.e., fractions of corrupted pixels) and different filter sizes are given in Table 2. The reported values are averages over all processed images. We observe that the normalized L_1 median filter and the arc distance median filter give consistently better results than the separable median filter. The normalized L_1 median filter gives slightly better results than the arc distance median filter for low noise levels in combination with small filter sizes only. In all other cases, including in particular higher noise levels, the arc distance median filter performs slightly better. In Figure 4, we illustrate the qualitative effects of the different median filters on a synthetic image with impulsive noise. There, we see that the separable median filter can lead to distortions

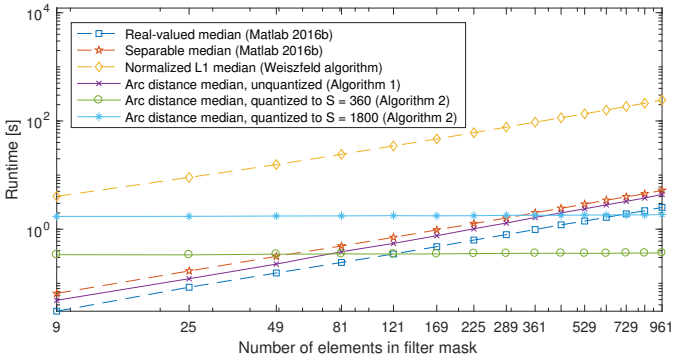


Fig. 5: Runtime of our methods for a randomly generated image of size 500×500 and filter masks of size $R \times R$ with $R = 3, 5, \ldots, 31$. Our method for non-quantized data (Algorithm 1) is slightly faster than the separable median filter, which mainly consists of applying a real-valued median filter twice. Furthermore it is around 60 times faster than the normalized L_1 median filter, which is computed using the Weiszfeld algorithm. The runtime of our algorithm for quantized data (Algorithm 2) is constant with respect to the filter size.

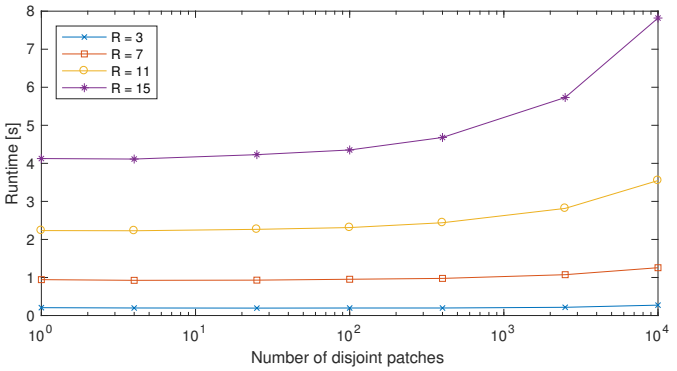


Fig. 6: Total runtime of the arc distance median filter (Algorithm 1) for patchwise processing of a 1000×1000 image. If the filter masks are small relative to the patch size, there is almost no loss of efficiency.

at the sharp edges. The normalized L_1 median filter and the arc distance median filter give comparably good results in terms of reconstruction quality.

5.2 Runtimes and potential for parallelization

We first compare the runtimes of the proposed arc distance median filter with those of the separable median filter and the normalized L_1 median filter for unquantized data. Figure 5 shows the CPU times for the image size 500×500 with different filter sizes. The reported runtimes are averages over ten images which were generated randomly using the uniform distribution on the unit circle. (See also Figures 4, 7, 8, and Tables 2, 3 for runtimes for other types of images.) We observe that Algorithm 1 needs less than twice the time of the real-valued median filter implemented in Matlab. Hence, Algorithm 1 is slightly faster than the separable median filtering, which essentially consists of executing the real-valued median filter twice. On the other hand, Algorithm 1 is around 60 times faster than the normalized L_1 median filter. The limit for real-time processing, i.e., processing 25 frames per second, is currently around 450×450 pixel for a 3×3 filter mask for the non-quantized algorithm. We note that the computational efforts of the proposed algorithms only depend on the filter size and the image size, but not on the image content.

Figure 5 also shows that the runtime of Algorithm 2 is constant with respect to the filter size. For data quantized to $S = 360$ values, Algorithm 2 is faster for filter masks larger than 7×7, whereas Algorithm 1 is faster for smaller filters. At quantization level $S = 1800$, the critical size is 19×19.

A simple yet effective way to parallelize the proposed algorithms is to process the filtering on disjoint image patches. This only requires adjustment of the boundary conditions according to the continuation at the patch boundaries. To explore the potential of this patchwise filtering, we partitioned an image of size 1000^2 into quadratic patches of side lengths between 1000 and 10 pixels, so that we obtain from one up to 10^4 patches. In Figure 6 we report the runtimes of the patchwise processing. We observe that there is almost no loss in efficiency if the filter mask is small in relation to the patch size. If the filter size is close to the patch size the total runtime gets around twice as high. Thus, the workload can be distributed efficiently to a large number of processors so that parallelization on the GPU seems feasible.

5.3 Experimental results on real data

5.3.1 Smoothing of phase data – InSAR images

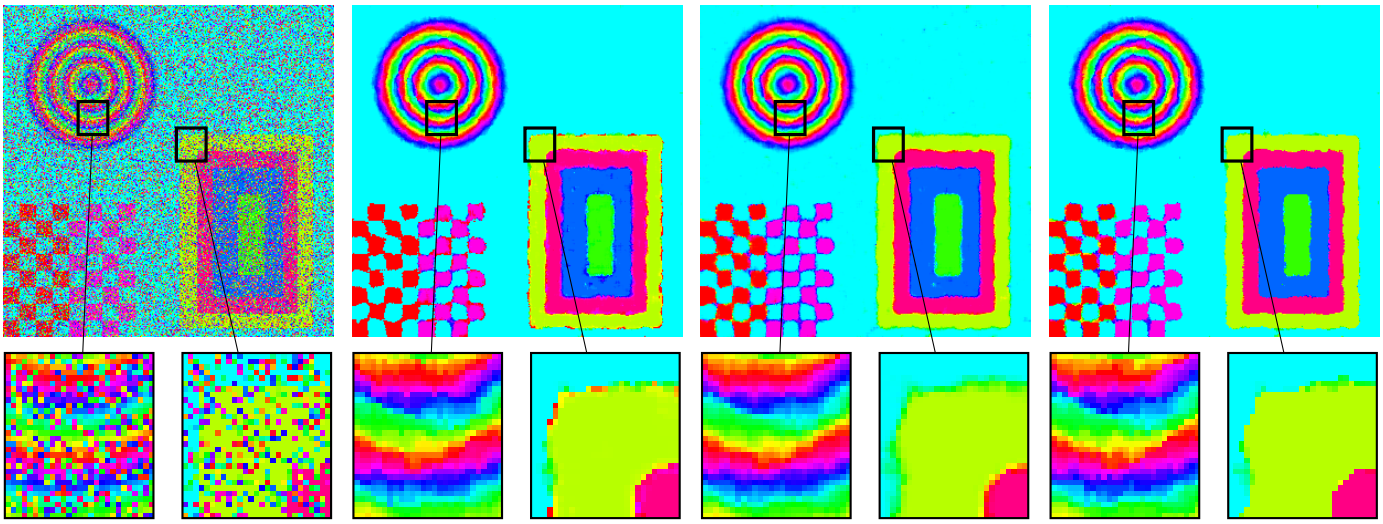
Synthetic aperture radar (SAR) is a radar technique for sensing the earth's surface from measurements taken by aircrafts or satellites. Interferometric synthetic aperture radar (InSAR) images consist of the phase difference between two SAR images, recording a region of interest either from two different angles of view or at two different points in times. Important applications of InSAR are the creation of accurate digital elevation models and the detection of terrain changes; cf. [5, 44]. As InSAR data consists of phase values that are defined modulo 2π, their natural data space is the unit circle. In Figure 7, we illustrate circle-median filtering to an InSAR image taken from [45]. We observe that the results of the separable median is less smooth than those of the normalized L_1 median and the arc distance median. The results of the normalized L_1 median and the arc distance median are comparable with respect to quality, but the latter requires significantly less computation time.

5.3.2 Smoothing the angular component in polar coordinates – Optical flow images

In planar flow fields each pixel contains a two-dimensional displacement vector, i.e., $y_{ij} = (z_1, z_2) \in \mathbb{R}^2$. Such images appear for example in optical flow [6] or wind fields [7]. One way of smoothing flow fields consists of applying a classical real-valued median filter on both components separately [46]. A drawback of that approach is that it is not invariant with respect to rotations. Rotation-invariant filtering can be achieved by filtering based on the vector median or by median filtering in polar coordinates. The latter means that we write the displacement vector in polar coordinates $(z_1, z_2) = (r \cos\omega, r \sin\omega)$, and apply the arc distance median filter to the angular component ω and the classical real-valued median filter to the radial component r.

Filter size		Separable median filter 3^2	5^2	7^2	9^2	11^2	Normalized L_1 median filter 3^2	5^2	7^2	9^2	11^2	Arc distance median filter 3^2	5^2	7^2	9^2	11^2
$\mathrm{NRI_c}$ ∅	0.1	9.6	7.3	5.9	4.9	4.3	**9.7**	**7.3**	*5.9*	*4.9*	*4.3*	*9.7*	*7.3*	**5.9**	**4.9**	**4.3**
(different	0.2	10.6	9.9	8.7	7.8	7.2	**11.2**	**10.1**	**8.8**	**7.9**	**7.2**	*11.0*	*10.0*	*8.8*	*7.9*	*7.2*
noise levels)	0.3	9.3	11.2	10.2	9.4	8.8	**10.3**	**11.5**	**10.4**	**9.5**	*8.9*	*10.1*	*11.4*	*10.3*	*9.5*	**8.9**
	0.4	7.3	11.2	11.1	10.4	9.9	**8.3**	*12.0*	**11.4**	*10.6*	*10.1*	*8.2*	**12.0**	*11.4*	**10.6**	**10.1**
	0.5	5.3	9.7	10.9	10.8	10.5	**6.2**	*11.1*	*11.8*	*11.3*	*10.8*	*6.1*	**11.5**	**12.0**	**11.4**	**10.9**
	0.6	3.7	7.3	9.2	9.8	10.0	*4.3*	*8.8*	*10.8*	*11.2*	*11.0*	**4.3**	**9.5**	**11.9**	**11.9**	**11.5**
	0.7	2.3	4.9	6.7	7.6	8.0	*2.8*	*6.0*	*8.3*	*9.5*	*10.0*	**2.8**	**6.6**	**9.9**	**11.5**	**11.7**
	0.8	1.3	2.8	4.2	5.2	5.9	*1.5*	*3.4*	*5.2*	*6.6*	*7.6*	**1.6**	**3.8**	**6.1**	**8.3**	**9.9**
	0.9	0.6	1.2	1.9	2.5	3.2	*0.6*	*1.4*	*2.2*	*3.0*	*3.8*	**0.7**	**1.5**	**2.4**	**3.4**	**4.4**
Runtime ∅ [s]		*0.04*	*0.09*	*0.17*	*0.27*	*0.39*	2.40	6.39	12.35	20.30	30.28	**0.03**	**0.07**	**0.14**	**0.22**	**0.33**

TABLE 2: Quantitative comparison of circle-median filters on a synthetic data set. The numbers represent the noise reduction index $\mathrm{NRI_c}$. The normalized L_1 median filter yields the best results for lower noise levels combined with smaller filter sizes whereas the arc distance median filter provides the best results at higher noise levels as well as for larger filter sizes. The proposed algorithm is much faster than the normalized L_1 median filter.



(a) Noisy data (Size 300×300, 50% impulsive noise)

(b) Separable median filter
CPU time: 0.097 s, $\mathrm{NRI_c}$: 8.6

(c) Normalized L_1 median filter
CPU time: 8.097 s, $\mathrm{NRI_c}$: 9.4

(d) Arc distance median filter
CPU time: 0.090 s, $\mathrm{NRI_c}$: 9.7

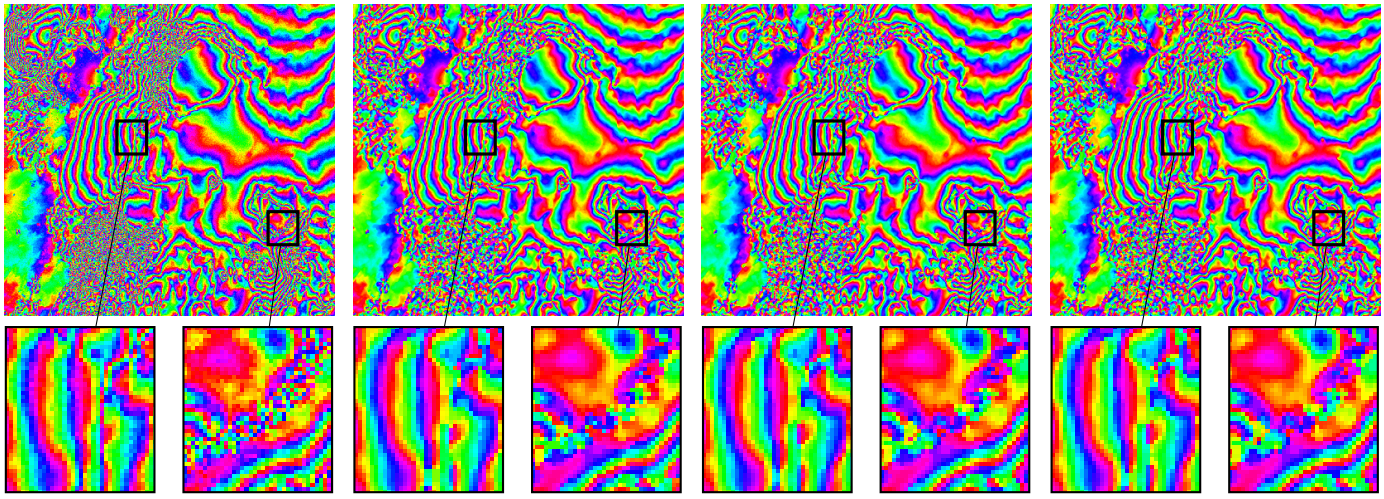
Fig. 4: Qualitative comparison of circle-median filters for a synthetic image using a filter of size 7×7. We observe that the separable median filter creates erroneous values at some edges. The normalized L_1 median filter and the arc distance median filter produce comparable results with respect to smoothness; the arc distance median filter has slightly sharper edge localization. (The circle-valued data are visualized as hue component in the HSV color space.)

This is particularly interesting when the focus is on smoothing the angular component. Regularization of optical flow images in polar representations has been used for example in [47]. To compare these approaches for smoothing flow fields, we have computed optical flow images using seven different methods discussed and implemented in [46]. In Table 3, we report the average angular error on the Middlebury training dataset [48] for these methods and the three mentioned filtering approaches. We observe that the polar filtering (using the arc distance median) gives consistently better results than the separable median at only moderately higher runtime. The vector median yields the best results in average. Filtering in polar coordinates almost attains the quality of the vector median, and it is more than 60 times faster. Considering the ratio of the average improvement and the average runtime, the filtering in polar coordinates turns out to be the most efficient of the three discussed approaches. We remark that, when considering the endpoint error as a measure of quality, filtering in polar coordinates only yields a slight improvement and, in that case, the vector median clearly gives the best results. We further point out that passing to weighted and bilateral filters typically improves upon the results of the present unweighted filters [16, 17, 46]. We briefly discuss the challenges of weighted median filters for circular data at the end of this paper. To summarize, median filtering in polar coordinates can be an efficient, rotation-invariant alternative to the classical median filtering approaches.

5.3.3 *Smoothing of orientation data – Time series of wind directions*

Eventually, we illustrate the utility of the proposed method for smoothing time series of orientations. The present data set consists of wind directions at station SAUF1 (St. Augustine, Florida) recorded every 10 minutes in the year 2014.[2] The data is given quantized to integer angles in degrees, thus $K = 360$. We smooth the data over around one day,

2. Wind direction data available at http://www.ndbc.noaa.gov/historical_data.shtml.



(a) Data (376×400)

(b) Separable median filtering
CPU time: 0.040 s

(c) Normalized L_1 median filtering
CPU time: 2.576 s

(d) Arc distance median filtering
CPU time: 0.032 s (Algorithm 1)

Fig. 7: Median filtering of an InSAR image from [45] using a 3×3 filter mask. The results of the separable median is less smooth than the others which can be observed in particular in the right zoomed image. The results of the normalized L_1 median and the arc distance median are comparable, but the latter requires significantly less computation time.

	Original (from [46])	Compo-nentwise	Vector median	Polar coord.
HS	4.688	4.620	**4.605**	*4.611*
BA	3.847	3.828	**3.805**	*3.806*
Classic-C	3.501	3.501	**3.481**	*3.484*
Classic++	3.301	3.299	**3.275**	*3.279*
Classic+NL-Full	2.660	2.657	**2.644**	*2.649*
Classic+NL	2.655	2.655	**2.637**	*2.641*
Classic+NL-Fastp	2.645	2.649	**2.634**	*2.639*
Improvement ∅		0.002	**0.026**	*0.022*
Runtime ∅ [s]		**0.106**	11.391	*0.183*
Improv./Runtime ∅		*0.020*	0.002	**0.121**

TABLE 3: *Top:* Average angular error for various methods of optical flow estimation on the Middlebury training set, and the results of subsequent median-type smoothing using a 5×5 filter mask. *Bottom:* Average improvement in quality, average runtime, and ratio between improvement and runtime.

that is, we use the filter size $R = 24 \cdot 6 + 1 = 145$ and $T = 1$. The smoothed signal facilitates to identify the trends of the wind directions; see Figure 8.

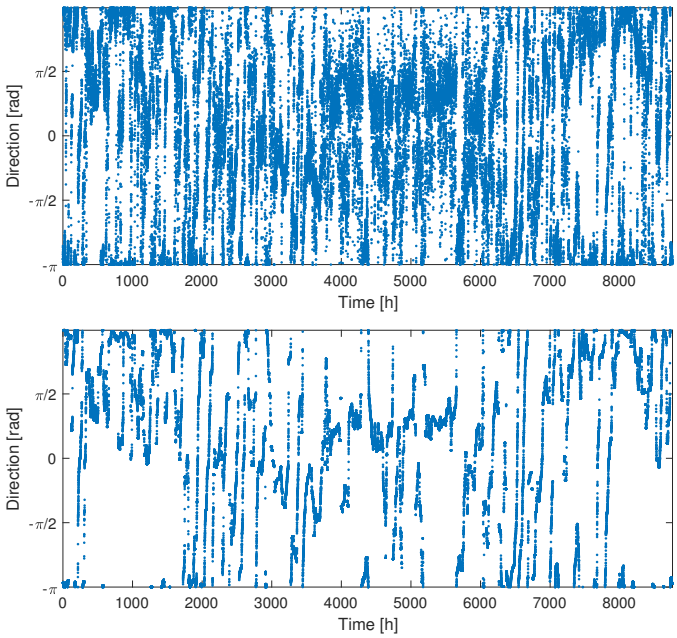


Fig. 8: *Top:* Wind directions at station SAUF1 (St. Augustine, Florida) recorded every 10 minutes in the year 2014. *Bottom:* Arc distance median filter with filter size 145. The data is given quantized to $K = 360$ angles. Therefore, we use Algorithm 2. The computation time amounts to only 0.09 seconds for the signal of length $N = 52549$.

6 Discussion and Conclusion

We start this section by discussing a conjecture on the complexity of arc distance median filtering for non-quantized circle-valued data. Then we conclude with summarizing our contributions and discussing future research.

6.1 Conjecture on the complexity of arc distance median filtering for non-quantized circle-valued data

As mentioned in the introduction, the best known algorithm for median filtering for (non-quantized) real-valued data, the Gil-Werman algorithm [1], has squared logarithmic complexity in the size of the filter mask, and it almost reaches the theoretical lower bound for median filtering for real-valued data. For circle-valued data, approaches similar to the Gil-Werman algorithm seem unfeasible, since that algorithm takes advantage of the fact that a real-valued median is characterized by the bisecting property on the real line. We have seen that the analogous bisecting property is only necessary but not sufficient for the arc distance median; see Example 2. We conjecture that the linear complexity of the first proposed algorithm (Algorithm 1) cannot be improved further which is motivated as follows. The defining functional in (1) is non-convex, and it can have as many (isolated) local minima as data items in the mask.

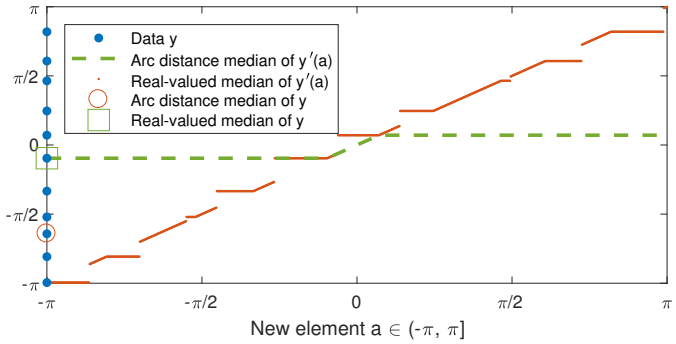


Fig. 9: Medians of the data $y'(a) = (y_2, \dots, y_{11}, a)$ in dependance of the element a using data y as in Figure 2, right. The real-valued median, i.e., when considering the angles of y' as real-valued data, only has three possible candidates. In contrast, the arc distance median has as many candidates as data points (corresponding to the steps and a itself as the diagonal).

This happens, for example, when the data in the mask are approximately equally distributed on the unit circle. To determine the global solution, one has to keep track of the functional values of all those local minima. In particular, if the number of local minima is in the order of the filter size, the number of required updates per processed element is proportional to the size of the filter mask. In contrast, the defining functional of the real-valued median has either one data point or an interval connecting exactly two data items as local minimizers which are automatically global minimizers by convexity. This can be exploited, e.g., when inserting a new data item and updating the median by descending from the old median to the nearby local minimizer. For circle data, the situation is different; the global minimizer might change drastically. Let us, for example, consider data y as in Figure 2, right. Assume that we have computed the arc distance median for y and that we want to compute the median of data $y'(a) = (y_2, \dots, y_{11}, a)$. Figure 9 shows the arc distance median of y' in dependance of the valued $a \in \mathbb{T}$. We observe that the introduction of only one new element a leads to 11 different possibilities for the arc distance medians – as many possibilities as data points (in contrast to the three possibilities for real-valued data).

6.2 Conclusion and future research

In this work, we have first discussed different means of defining a median filter on the unit circle. We have identified the arc distance median filter as a robust edge preserving filter which preserves the values of the data and gives an almost everywhere unique filter output for odd filter size. As such, it shares the desired properties of the real valued median filter. Then, we have derived efficient algorithms for arc distance median filtering for quantized as well as for non-quantized data. For fast computations, both algorithms employ an efficient scheme based on suitable recurrence relations. The proposed algorithm for quantized data scales constantly with the size of the filter mask which is optimal. For non-quantized data, we have developed a linearly scaling algorithm. We have observed that, for small filter masks, the runtime of the proposed algorithm for non-quantized data applied to quantized data is even lower than the runtime of the proposed algorithm for quantized data. Further we have seen that the proposed methods provide higher quality than separable median filtering in even slightly less time. On the other hand, they achieve comparable quality to the results of normalized L_1 median filtering in a significantly shorter time. Due to their practical performance and their efficiency, we believe that the proposed algorithms have the potential to become a standard tool for processing circle-valued data.

The extension of the algorithms to 3D images and other shapes of the filter masks seems straightforward. A particularly interesting direction of future research is a weighted variant of the arc distance median filter, because appropriate weighting can improve the result even further; see [16, 17]. The weighted arc distance median is given by multiplying each summand in (1) by a non-negative weight w_{ij}. The challenge is that the elements change their weights when the filter mask is shifted. Here histogram based schemes similar to those in [16, 17] seem to be appropriate to efficiently compute the required functional values. Another direction is the investigation of iterated circle-median filters as it has been done in [49] for the real-valued case.

Acknowledgement

This work was supported by the German Research Foundation (DFG STO1126/2-1, WE5886/4-1, and WE5886/3-1).

References

[1] J. Gil and M. Werman, "Computing 2-dimensional min, median and max filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 5, 1993.

[2] S. Perreault and P. Hébert, "Median filtering in constant time," *IEEE T Image Process*, vol. 16, no. 9, pp. 2389–2394, 2007.

[3] Y. Sowa, A. Rowe, M. Leake, T. Yakushi, M. Homma, A. Ishijima, and R. Berry, "Direct observation of steps in rotation of the bacterial flagellar motor," *Nature*, vol. 437, no. 7060, pp. 916–919, 2005.

[4] J. Davis and R. Sampson, *Statistics and Data Analysis in Geology*. New York: Wiley, 2002.

[5] D. Massonnet and K. Feigl, "Radar interferometry and its application to changes in the earth's surface," *Reviews of Geophysics*, vol. 36, pp. 441–500, 1998.

[6] B. K. Horn and B. Schunck, "Determining optical flow," in *Technical Symposium East*. International Society for Optics and Photonics, 1981, pp. 319–331.

[7] H. Schultz, "A circular median filter approach for resolving directional ambiguities in wind fields retrieved from spaceborne scatterometer data," *Journal of Geophysical Research: Oceans*, vol. 95, no. C4, pp. 5291–5303, 1990.

[8] K. Mardia, *Statistics of directional data*. Academic Press, 1972.

[9] G. Ducharme and P. Milasevic, "Spatial median and directional data," *Biometrika*, vol. 74, no. 1, pp. 212–215, 1987.

[10] N. Fisher, "Spherical medians," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 342–348, 1985.

[11] N. Nikolaidis and I. Pitas, "Nonlinear processing and analysis of angular signals," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3181–3194, 1998.

[12] P. Fletcher, S. Venkatasubramanian, and S. Joshi, "The geometric median on Riemannian manifolds with application to robust atlas estimation," *NeuroImage*, vol. 45, no. 1, pp. S143–S152, 2009.

[13] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979.

[14] B. Weiss, "Fast median and bilateral filtering," in *ACM Transactions on Graphics*, vol. 25, no. 3, 2006, pp. 519–526.

[15] Q. Yang, N. Ahuja, and K.-H. Tan, "Constant time median and bilateral filtering," *Int J Comput Vision*, vol. 112, no. 3, pp. 307–318, 2014.

[16] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *Proc IEEE ICCV*, 2013, pp. 49–56.

[17] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (WMF)," in *Proc IEEE CVPR*, 2014, pp. 2830–2837.

[18] N. Fisher, *Statistical Analysis of Circular Data*. Cambridge University Press, 1995.

[19] K. Mardia and P. Jupp, *Directional statistics*. John Wiley & Sons, 2009, vol. 494.

[20] R. Liu and K. Singh, "Ordering directional data: concepts of data depth on circles and spheres," *The Annals of Statistics*, pp. 1468–1484, 1992.

[21] X. He and D. Simpson, "Robust direction estimation," *The Annals of Statistics*, pp. 351–369, 1992.

[22] S. Purkayastha, "An almost sure representation of sample circular median," *Journal of Statistical Planning and Inference*, vol. 46, no. 1, pp. 77–91, 1995.

[23] ——, "On the asymptotic efficiency of the sample circular median," *Statistics and Decisions*, vol. 13, no. 3, pp. 243–252, 1995.

[24] T. Wehrly and E. Shine, "Influence curves of estimators for directional data," *Biometrika*, vol. 68, no. 1, pp. 334–335, 1981.

[25] B. Otieno and C. Anderson-Cook, "Measures of preferred direction for environmental and ecological circular data," *Environmental and Ecological Statistics*, vol. 13, no. 3, pp. 311–324, 2006.

[26] N. Fisher and C. Powell, "Statistical analysis of two-dimensional palaeocurrent data: methods and examples," *Journal of the Geological Society of Australia*, vol. 36, no. 1, pp. 91–107, 1989.

[27] P. Berens, "CircStat: A MATLAB toolbox for circular statistics," *J Stat Softw*, vol. 31, no. 10, pp. 1–21, 2009.

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

[29] M. Giaquinta, G. Modica, and J. Souček, "Variational problems for maps of bounded variation with values in S^1," *Calc. Var.*, vol. 1, no. 1, pp. 87–121, 1993.

[30] T. Chan, S. Kang, and J. Shen, "Total variation denoising and enhancement of color images based on the CB and HSV color models," *Journal of Visual Communication and Image Representation*, vol. 12, pp. 422–435, 2001.

[31] D. Cremers and E. Strekalovskiy, "Total cyclic variation and generalizations," *J Math Imaging Vis*, vol. 47, no. 3, pp. 258–277, 2013.

[32] J. Lellmann, E. Strekalovskiy, S. Koetter, and D. Cremers, "Total variation regularization for functions with values in a manifold," in *Proc IEEE ICCV*, 2013, pp. 2944–2951.

[33] A. Weinmann, L. Demaret, and M. Storath, "Total variation regularization for manifold-valued data," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2226–2257, 2014.

[34] R. Bergmann, F. Laus, G. Steidl, and A. Weinmann, "Second order differences of cyclic data and applications in variational denoising," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 2916–2953, 2014.

[35] M. Storath, A. Weinmann, and M. Unser, "Exact algorithms for L^1-TV regularization of real-valued or circle-valued signals," *SIAM Journal on Scientific Computing*, vol. 38, no. 1, pp. A614–A630, 2016.

[36] A. Weinmann, L. Demaret, and M. Storath, "Mumford-Shah and Potts regularization for manifold-valued data," *J Math Imaging Vis*, vol. 55, no. 3, pp. 428–445, 2016.

[37] P. Grohs and M. Sprecher, "Total variation regularization on Riemannian manifolds by iteratively reweighted minimization," *Information and Inference*, 2016, to appear.

[38] E. Strekalovskiy and D. Cremers, "Total variation for cyclic structures: Convex relaxation and efficient minimization," in *IEEE Proc CVPR*, 2011, pp. 1905–1911.

[39] C. Small, "A survey of multidimensional medians," *International Statistical Review*, pp. 263–277, 1990.

[40] E. Weiszfeld, "Sur le point pour lequel la somme des distances de n points donnés est minimum," *Tohoku Mathematical Journal*, vol. 43, no. 355-386, p. 2, 1937.

[41] H. Fritz, P. Filzmoser, and C. Croux, "A comparison of algorithms for the multivariate L_1-median," *Computational Statistics*, vol. 27, no. 3, pp. 393–410, 2012.

[42] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th International Conference on Computer Vision*, 2001, pp. 416–423, http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[43] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[44] F. Rocca, C. Prati, and A. Ferretti, "An overview of SAR interferometry," in *Proceedings of the 3rd ERS Symposium on Space at the Service of our Environment, Florence*, 1997.

[45] K.-H. Thiel, X. Wu, and P. Hartl, "ERS-tandem-interferometric observation of volcanic activities in Iceland," *ESA SP*, pp. 475–480, 1997. [Online]. Available: https://earth.esa.int/workshops/ers97/papers/thiel/index-2.html

[46] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Proc IEEE CVPR*, 2010, pp. 2432–2439. [Online]. Available: http://cs.brown.edu/~dqsun/research/software.html

[47] Y. Adato, T. Zickler, and O. Ben-Shahar, "A polar representation of motion and implications for optical flow," in *Proc IEEE CVPR*, 2011, pp. 1145–1152.

[48] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int J Comput Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[49] E. Arias-Castro and D. Donoho, "Does median filtering truly preserve edges better than linear filtering?" *The Annals of Statistics*, pp. 1172–1206, 2009.

Martin Storath received his Diploma degree in Mathematics in 2008, his Honours degree in Technology Management in 2009, and his doctoral degree in Mathematics in 2013, all from Technische Universität München. He worked as researcher at the Helmholtz Zentrum München and at the Biomedical Imaging Group, EPFL. Currently, he is a postdoc at the Image Analysis and Learning Group, Universität Heidelberg. His research interests include image processing, variational methods, and inverse problems.

Andreas Weinmann received his Diploma degree in mathematics and computer science from Technische Universität München in 2006 and his Ph.D. degree from Technische Universität Graz in 2010 (both with highest distinction). He worked as a researcher at Technische Universität München. Currently, he is affiliated with the Helmholtz Center Munich and the Hochschule Darmstadt. His research interests are applied analysis as well as signal and image processing.