2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

On computing the average orientation of vectors and lines

Edwin Olson
University of Michigan
ebolson@umich.edu
http://april.eecs.umich.edu

Abstract— **Computing the "average" orientation of lines and rotations is non-trivial due to the need to account for the periodicity of rotation. For example, the average of +10 and +350 degrees is 0 (or 360) degrees, not 180. This problem arises in many disciplines, particularly those in which empirically collected data is processed or filtered.**

In this paper, we review a common but sub-optimal method for computing average orientation and provide a new geometric interpretation. We also propose a new method which provides additional insights into the geometry of the problem. Our new method also produces significantly more accurate results in the regime of operation usually encountered in robotics applications. We characterize this regime and provide guidance regarding which method to use, and when.

I. Introduction

Given a set of orientations (perhaps the direction of a noisy compass over time), it is often desirable to compute the average orientation. This is not a trivial operation due to the periodicity of rotations. For example, when averaging rotations of +10 degrees and +350 degrees, we want to obtain an answer of 0 (or 360) degrees, rather than 180 degrees. As the number of inputs increases, it becomes more difficult to account for the 2π periodicity of rotations.

Computing average orientations arises in many disciplines. In the specific case of the authors, it arose in the case of a difficult Simultaneous Localization and Mapping (SLAM) problem for which the initial solution was very noisy, but for which the orientation of the robot was known accurately for a handful of poses (due to GPS). Without a reasonable initialization (generally considered to mean heading errors less than $\pi/4$), many SLAM algorithms are prone to divergence due to the fact that the Jacobians do not necessarily point in the direction of the global minimum [1]. By interpolating the orientation of the known points along the robot trajectory, the stability of the method is improved. This operation requires computing weighted averages of the GPS headings.

Closely related to the idea of an "average orientation" is that of a "canonical orientation". In feature descriptor algorithms like SIFT [2] and SURF [3], a histogram of directions is used to find a canonical orientation to describe a pixel patch. The same idea has been applied to LIDAR data [4]. An average orientation could serve a similar role and would have the potential advantage of avoiding quantization noise resulting from constructing the histogram.

Average orientation also arises in evaluation metrics; in judging the performance of a mapping or localization system, it can be useful to characterize the average heading error of the system [5]. The average orientation error also arises in the context of evaluating the dead-reckoning performance of both robotic systems [6] and humans [7]. Human perception of orientation also makes use of directional statistics [8].

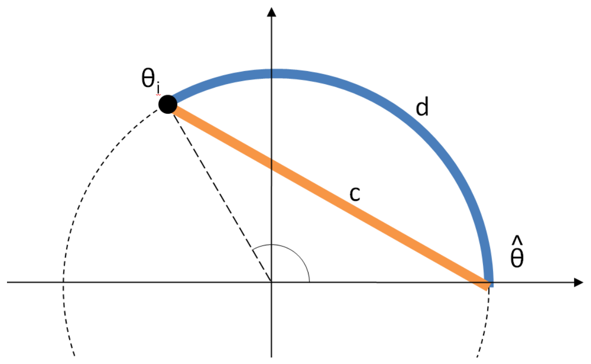

Fig. 1. Arc versus chord distance. Given an estimate of average orientation $\hat{\theta}$ and an observation θ_i, we can consider two distance measures: the arc length (d) and the chord length (c).

More broadly, computing average orientation arises in animal behavior (such as the behavior of homing pigeons), geology (the directions of particles in sediments), meteorology (the direction of the wind), etc. These and other examples are discussed in [9], [10].

Computing averages of orientations falls into the subject of directional statistics, which despite having several well-written texts [9], [10], is largely treated in an ad-hoc fashion by most application-oriented papers.

This paper begins with a review and analysis of the widely used "vector sum" algorithm. We provide a new geometric interpretation of the algorithm that makes it easier to understand the consequences of its underlying approximations.

The central contribution of this paper is a new method for computing average orientation, which we describe in both probabilistic and geometric terms. Our evaluation demonstrates that, for many applications, our approach computes better estimates than the vector sum algorithm.

Finally, we discuss an important variant of the average orientation problem that arises when working with undirected lines. These problems have a periodicity of π, rather than 2π. We show how both methods can be adapted to this domain, and compare their performance.

II. The vector sum method

The vector sum method is the typical approach used to compute average orientation. We will first describe the method, review its properties, then give a new geometric interpretation.

A. Method

Suppose we are given a set of direction measurements θ_i for $i \in [1, N]$. We first compute the unit vector for each measurement, then sum those vectors:

$$x = \sum_i \cos(\theta_i) \quad (1)$$
$$y = \sum_i \sin(\theta_i)$$

Now, we compute the angle of the vector sum using the four-quadrant arc tangent; this will be our estimate of the mean orientation:

$$\bar{\theta} = \arctan(y, x) \quad (2)$$

This method produces "reasonable" results, however the specific properties of this method are not obvious. Many application-oriented papers employ this approach without any additional consideration. One might ask: Is the method optimal? What objective function does the method optimize?

In particular, many applications assume that angular measurements are contaminated with additive Gaussian noise, i.e., that there is some true orientation $\hat{\theta}$ and that the observations are of the form $\theta_i = \hat{\theta} + w_i$, where $w_i \sim N(0, \sigma^2)$. As we will show in the following sections, the vector sum method is not always an appropriate method under these assumptions. But first, we'll need to review some of the essential ideas in directional statistics.

B. The wrapped normal distribution

Let us consider more carefully the properties that a probability distribution *should* have in the case of circular data. Specifically, because an orientation of θ is indistinguishable from a rotation of $\theta + 2\pi$, the probability density f must satisfy:

$$f(\theta) = f(\theta + 2\pi) \quad (3)$$

Further, suppose we observe θ_i. This observation is indistinguishable from $\theta_i + 2\pi$, and $\theta_i + 4\pi$, etc. Our distribution should account for this ambiguity.

For example, suppose that we are observing orientations with a mean value of $\mu = 45$ degrees, corrupted by Gaussian noise w_i with variance σ^2. If we observe a value of 50 degrees, it is possible that we have observed a value in which $w_i = 5$, but it is also possible that $w_i = 365$, or more generally, $w_i = 5 + 2\pi k$ for any integer k. All of these events (for different values of k) map to the same observed value, thus the density associated with an observation is the sum of the densities for every value of k:

$$f(\theta) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} e^{-(\theta + 2\pi k - \mu)^2/2\sigma^2} \quad (4)$$

Intuitively, we can imagine the usual Gaussian distribution "wrapped" around the unit circle. Since the Gaussian distribution has infinite support, it wraps around an infinite number of times, adding to the density at every orientation as it goes around. Since the density of a Gaussian distribution drops off rapidly with distance from the mean, the density is generally dominated by the first few wraps (and often just the first wrap).

As in the non-circular case, we require that a density integrate to 1; however, we only consider an interval of 2π degrees; any contiguous interval of 2π degrees (by Eqn. 3) will do:

$$\int_{-\pi}^{\pi} f(x)dx = 1 \quad (5)$$

The wrapped normal distribution represents the most rigorous way of representing circular data contaminated by Gaussian noise, however it is analytically difficult to work with. This is because the density includes a sum; when computing the maximum likelihood value of μ given a set of observations, we typically take the logarithm in order to simplify the expression and solve for μ. However, the summation prevents us from doing this.

C. The von Mises distribution

There exists a probability distribution such that, if w_i is distributed according to it, the vector sum method is optimal. This distribution is the von Mises distribution [9]. This distribution is also sometimes called the Circular Normal distribution, but we will not use that name here as it implies a closer relationship to the Normal (Gaussian) distribution than actually exists.

The von Mises distribution is:

$$f(\theta, \mu) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa cos(\theta - \mu)} \quad (6)$$

The κ parameter determines the spread of the distribution, and plays the same role as $1/\sigma^2$ in a Gaussian distribution. The leading term is simply for normalization; $I_o(\kappa)$ represents the modified Bessel function of the first kind and zero order. For our purposes here, only the exponential term is interesting, since it determines the shape of the distribution. It is obvious that it is *not* of the same form as a Gaussian distribution, which would have a quadratic loss of the form $e^{(\theta-\mu)^2/\sigma^2}$.

The properties of the von Mises distribution are not obvious upon inspection. In contrast, the maximum likelihood of the Gaussian distribution's mean value μ is the one that minimizes the squared distance between the mean and each point. From a geometric point of view, what property does the maximum likelihood solution of the von Mises distribution have?

D. Geometric Interpretation

As we saw above, the vector sum algorithm computes the optimal estimate of θ assuming that the noise is governed by the von Mises distribution. It is not clear, however, what this means in terms of the geometry of the points around the unit circle. In this section, we derive a useful geometric interpretation.

Let us again consider observations θ_i for $i \in [1, N]$. The unit vectors corresponding to each observation lie on the unit circle. Now, let us find a $\hat{\theta}$ that minimizes the squared distance between $\hat{\theta}$ and each of the θ_is.

The challenge is to define distance in a way that is computationally convenient. The length of the arc between $\hat{\theta}$ and θ_i would be exactly right if we assume our noise is normally distributed and that there was no "wrapping" (i.e., that the noise was bounded by π degrees); see Fig. 1. However, the arc distance is difficult to incorporate into a closed-form solution.

Alternatively, we can use the chord distance (again, see Fig. 1). The chord distance and arc distance are approximately the same when $\hat{\theta}$ and θ_i are nearby. Thus, we can approximate the maximum likelihood μ (assuming Gaussian noise) by minimizing the squared chord distance.

Using basic trigonometry, the vector r_i between the unit vectors corresponding to $\hat{\theta}$ and θ_i is:

$$r_i = \begin{bmatrix} \cos(\hat{\theta}) \\ \sin(\hat{\theta}) \end{bmatrix} - \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \tag{7}$$

The squared distance between those points is then just $r_i^T r_i$. We sum this over all observation points, expanding the product:

$$\begin{aligned} \chi^2 = \quad & \textstyle\sum_i \cos^2(\hat{\theta}) - 2\cos(\hat{\theta})\cos(\theta_i) + \cos^2(\theta_i) + \\ & \sin^2(\hat{\theta}) - 2\sin(\hat{\theta})\sin(\theta_i) + \sin^2(\hat{\theta}) \end{aligned} \tag{8}$$

Noting that $\cos^2(x) + \sin^2(x) = 1$, we can simplify to:

$$\chi^2 = \quad \textstyle\sum_i 2 - 2\cos(\hat{\theta})\cos(\theta_i) - 2\sin(\hat{\theta})\sin(\theta_i) = 0 \tag{9}$$

We wish to minimize the sum of squared distances, so we differentiate with respect to $\hat{\theta}$ and set to zero:

$$\frac{\partial \chi^2}{\partial \hat{\theta}} = \sum_i \sin(\hat{\theta})\cos(\theta_i) - \cos(\hat{\theta})\sin(\theta_i) = 0 \tag{10}$$

Collecting terms and pulling our unknown ($\hat{\theta}$) to the left side, we obtain:

$$\frac{\sin(\hat{\theta})}{\cos(\hat{\theta})} = \sum_i \frac{\sin(\theta_i)}{\cos(\theta_i)} \tag{11}$$

$$\hat{\theta} = \arctan\left(\sum_i \cos(\theta_i), \sum_i \sin(\theta_i)\right) \tag{12}$$

In other words, the estimate of $\hat{\theta}$ that minimizes the squared chord distance is the same as the vector sum algorithm.

We can relate this directly to the form of the von Mises distribution as well. First, we note that the squared chord length is $2 - \cos(\theta_i - \hat{\theta})$. (This is easily derived from Fig. 1 using the law of cosines.) In minimizing the sum of these costs, the 2 is inconsequential: we can drop the term without changing the answer. Now, we are left with precisely the exponentiated expression in the von Mises distribution.

III. Proposed Method

We now propose a new method for computing the average orientation. In many cases, this method better approximates the wrapped normal distribution than the von Mises distribution, while still being computationally inexpensive to compute.

A. Interpretation

Our method can be described in both geometric and probabilistic terms. In the previous section, we showed that the von Mises distribution corresponds to minimizing the squared *chord* length associated with each observation. The *arc* length would be a more accurate method if we assume Gaussian noise since the length of the arc is proportional to the difference in orientation. (As a reminder, the maximum likelihood estimator of μ minimizes the sum of the *squares* of differences in orientation; this is the reason we use the *squared* arc length.)

In fact, using the squared arc length is optimal if the noise is bounded by π. When this does not hold, using the arc length is an approximation: for an observation with noise greater than π, we will pick the shorter direction around the circle, rather than take the "long way" around the unit circle. This type of error occurs only with extremely low probability, or with Gaussians with very large standard deviations.

Probabilistically, minimizing the squared arc length corresponds to wrapping a normal distribution around the unit circle without overlap, i.e., cutting off the tails of the normal distribution that would otherwise continue to wrap around the unit circle. Because the probability mass in the tails was removed, the total probability mass will be less than 1.0. We can correct for this by rescaling the distribution:

$$f(\theta) = \frac{1}{\mathrm{erf}(\pi/(\sigma\sqrt{2}))} \frac{1}{\sigma\sqrt{2\pi}} e^{-(\theta-\mu)^2/2\sigma^2} \tag{13}$$

From an analytic point-of-view, this density is fairly awkward: it cannot be easily evaluated due to the presence of the error function. However, our method (while producing maximum likelihood estimates for the distribution) does not require us to ever evaluate the densities, so this is not a practical problem.

We've now described the geometric (minimize the squared arc length) and probabilistic (truncate the Gaussian) interpretation of our method. We now show that the maximum likelihood solution can be quickly and exactly computed.

B. Implementation

The trick is to "unwrap" the observations back onto a line, and to use conventional (non-circular) estimators to compute the mean. The challenge is that the position of each observation on the line is not uniquely determined, since it can be freely translated by any multiple of 2π. Of course, no matter what the mean is, no point will be farther than π degrees away from it. Otherwise, a lower error solution could be obtained by moving that point by a multiple of 2π degrees so that it is closer to the mean.

Consequently, there exists an arrangement of observations that span no more than 2π degrees, and for which all points are within π of the mean (which is still unknown at this point). Supposing that our points were in this arrangement, the simple arithmetic mean would yield the optimal mean as measured by the fact that the mean squared arc length is minimized.

The basic idea of our method is to construct the possible arrangements of the observations, compute their mean and error, and output the best. At first, it may seem that there are too many possible arrangements of observations: each observation can appear at an infinite number of positions (all spaced 2π apart).

By the argument above, however, the only arrangements of consequence are those that span less than 2π degrees. Each of those arrangements has a "left-most" observation—an observation whose orientation is numerically smaller than all the other observations. Once a left-most point is identified, the positions of all the other points is fully determined due to the constraint that the arrangement span less than 2π degrees.

If we have N observations, then there are only N candidate arrangements. (Each observation takes turns being the left-most; once selected, the position of all the other observations is uniquely determined.)

Suppose that our observations are mapped to some arbitrary interval of width 2π (i.e., by moving each point to be $[0, 2\pi]$). We can construct the N arrangements by repeatedly taking the left most point and moving it to the right by adding 2π to it. Now, some other point is the left most point. After N iterations, we will have constructed all N arrangements of the observations.

At each iteration, we compute the mean and squared error. These statistics can be updated incrementally after each iteration by appropriately updating the first and second moments of the observations. For clarity, the first moment M_1 and second moment M_2 are simply:

$$M_1 = \sum_i \theta_i \tag{14}$$

$$M_2 = \sum_i \theta_i^2 \tag{15}$$

We can compute the mean and squared error as:

$$\hat{\theta}_i = M_1/N \tag{16}$$

$$\chi_i^2 = M_2 - M_1^2/N \tag{17}$$

Now, suppose that we take the left most point (call it θ_i) and move it to the far right by adding 2π. We can incrementally update the moments as:

$$M_1' = M_1 + 2\pi \tag{18}$$

$$\begin{aligned} M_2' &= M_2 - \theta_i^2 + (\theta_i + 2\pi)^2 \qquad (19) \\ &= M_2 + 4\pi\theta_i + (2\pi)^2 \end{aligned}$$

We simply repeat the process of moving the left-most point to the right N times, and record the mean that minimizes the squared answer. The complete algorithm is given in Alg. 1.

Algorithm 1 Compute mean theta using squared arc-length

```
1: $M_1 = 0$
2: $M_2 = 0$
3: {Map all points to $[0, 2\pi]$ and initialize moments}
4: for $i = 1$ to $N$ do
5:     $\theta_i = mod2pi(\theta_i)$
6:     $M_1 = M_1 + \theta_i$
7:     $M_2 = M_2 + \theta_i^2$
8: end for
9: Sort $\theta$s into increasing order
10: $\sigma^2 = \infty$
11: for $i = 0$ to $N$ do
12:     if $i >= 1$ then
13:         {Move the $i$th observation to the right by $2\pi$}
14:         $M_1 = M_1 + 2\pi$
15:         $M_2 = M_2 + 4\pi\theta_i + (2\pi)^2$
16:     end if
17:     $\hat{\theta}_i = M_1/N$
18:     $\sigma_i^2 = M_2 - 2 * M_1 * \hat{\theta}_i + N * \hat{\theta}_i^2$
19:     if $\sigma_i^2 < \sigma^2$ then
20:         $\hat{\theta} = \hat{\theta}_i$
21:         $\sigma^2 = \sigma_i^2$
22:     end if
23: end for
24: return $\hat{\theta}$
```

Our proposed method exactly computes the maximum likelihood estimate of μ assuming that the noise is distributed according to the truncated Gaussian distribution. As we've also shown, this is equivalent to minimizing the squared arc length.

IV. Evaluation

In this section, we evaluate the performance of the vector sum algorithm and the proposed method.

A. Approximation accuracy

Both algorithms attempt to approximate the wrapped normal distribution. The distributions are plotted for $\sigma = 0.5, 1.0, 2.0$ in Fig. 2. For small σ, both algorithms closely approximate the wrapped normal distribution. As the observation noise increases, however, there are two distinct regimes:

- At moderate noise levels ($\sigma < 1.6$), the proposed method out-performs the vector sum method. Intuitively, this is because the "chord" distance is taking a short cut around the circle, resulting in too much probability mass away from the mean.
- At higher noise levels ($\sigma > 1.6$), the vector sum method out-performs the proposed method. At these high noise levels, it becomes more likely that observations will have more than π noise. In other words, the "wrapping" of the wrapped normal distribution starts to flatten out the distribution. The flatter behavior of the von Mises distribution (due to taking "short cuts" through the center of the unit circle) is a better fit.

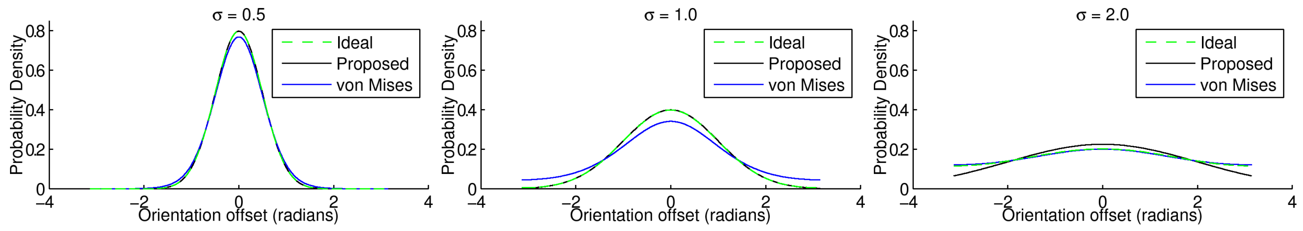

Fig. 2. Probability density functions for $\sigma = .5$, 1.0, and 2.0. For small sigmas, the performance of the proposed distribution (formed by truncating the Normal distribution) is significantly closer to the wrapped Gaussian than the von Mises distribution (see left two figures). The relative improvement of the proposed method increases with σ, until significant probability mass begins to wrap around (right). Also see Fig. 3.

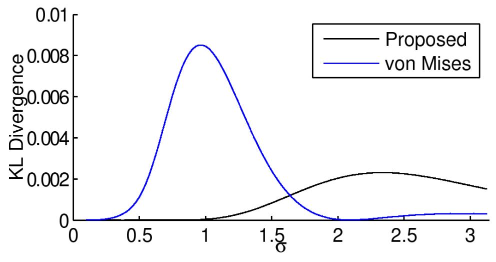

Fig. 3. KL divergence as a function of σ. The KL divergence from the wrapped normal (ideal) distribution to the proposed and von Mises distribution shows the relative accuracy of the two approximations. The proposed distribution not only has significantly better worst-case performance, but also has significantly less error in low-variance regime ($\sigma < 1.639$) in which most applications operate.

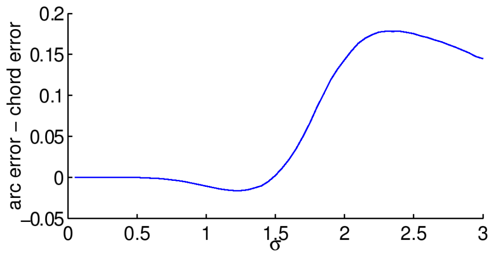

Fig. 4. Chord versus arc error as a function of σ. For each value of σ, a Monte Carlo simulation was run in which 50 directional observations were generated. Both the chord and arc (proposed) method were used to estimate the original mean. The sample standard deviation measures the performance of the estimator; the difference between the two are shown above. Values below zero indicate that the proposed method works better for that σ; values above zero indicate that the chord approximation is better. As predicted by the KL divergence results, the cross-over point occurs at around $\sigma = 1.5$, or about 85 degrees.

Fig. 2 shows the probability densities at three discrete points. Alternatively, we can compute the KL divergence [11] between the wrapped normal distribution and the von Mises or truncated Gaussian. As shown in Fig. 3, the KL divergence between the wrapped normal distribution and the von Mises distribution peaks at around $\sigma = 1$, then becomes smaller again. The truncated Gaussian distribution has significantly lower divergence in the low-to-moderate noise regime, but does worse in the high-noise regime.

B. Empirical performance

Our KL divergence data suggests that the truncated Gaussian is a better approximation of the wrapped normal distribution for $\sigma \in [0, 1.6\pi]$. To verify this, we conducted

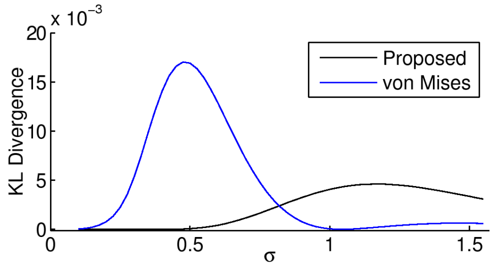

Fig. 6. KL divergence as a function of σ (line case).

a Monte Carlo simulation in which we generated noisy observations and estimated the mean using both methods. The error for both methods was compared and plotted in Fig. 4; negative values indicate that the vector sum algorithm had a larger error than the proposed method. Positive values indicate the opposite. The curve is very smooth due to a very large number of trials.

This data reveals a surprising result: while the performance of the proposed algorithm is indeed better for low-to-moderate σ, the vector sum method does much better in the high noise regime. In this case, the KL divergence analysis was approximately correct in the cross-over point (where vector sum would surpass the proposed method), but the magnitude of the KL divergence results were not especially predictive of the Monte Carlo results.

Still, the performance of the proposed method is demonstrably better than the vector sum method over the most useful range of σs; it is not until $\sigma = 1.6$ (i.e., a standard deviation of *90 degrees*) that the vector sum method surpasses it.

For maximum accuracy over the broadest possible range of σ, one could use both methods, switching between the two based on the estimate of σ.

C. Line variant

We have so far considered the case of observations with a periodicity of 2π, such as compass readings. In some applications, the data may have a periodicity of π, such as in estimating the average orientation of the walls in a room from LIDAR data.

Both methods can be easily adapted to this case. In the case of our proposed method, the changes are trivial: the scale factor is adjusted so that the density integrates to 1 over the smaller interval $[-\pi/2, \pi/2]$.

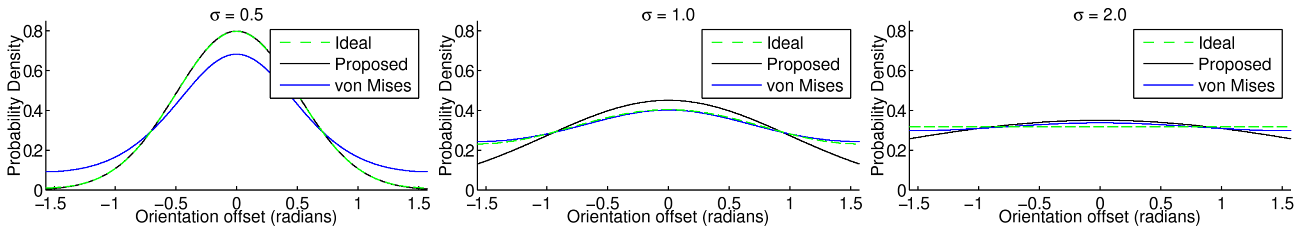

Fig. 5. Probability density functions for $\sigma = .5$, 1.0, and 2.0 (line case). See also Fig. 6.

In the case of the vector sum algorithm, the orientations of the observations can be multiplied by two, thus rescaling the problem so that it has a periodicity of 2π. This has an interesting consequence, as the probability density function becomes:

$$f(\theta, \mu) = \frac{1}{\pi I_0(\kappa)} e^{\kappa cos(2(\theta-\mu))} \tag{20}$$

Note that the factor of two is taken *inside* the cosine expression; the resulting density is not a simple rescaling of the von Mises distribution.

To examine the effect of these modifications, we again plotted the distributions for different values of σ (see Fig. 5) and the KL divergence (see Fig. 6). Note that, in computing κ for the von Mises distribution, we used $1/(2\sigma)^2$ instead of the original $1/\sigma^2$, in agreement with the variance of a random variable multiplied by two. Interestingly, the results from this case mirror almost exactly those of the 2π case.

D. Complexity analysis

From a complexity stand-point, the proposed method has additional costs versus the simpler vector sum algorithm. Both algorithms conceptually handle the observations one at a time, but the proposed method needs them to be in sorted (left-to-right) order, which incurs an $O(N \log N)$ cost. In addition, this requires the observations to be stored in memory; the vector sum algorithm can be applied to datasets of virtually unbounded size since the data does not need to be stored.

Algorithm	Computation	Memory
Vector sum	$O(N)$	$O(1)$
Proposed method	$O(N \log N)$	$O(N)$

E. Runtime performance

While the asymptotic computational complexity of our method is slower than the vector sum algorithm, the real-world difference is quite small. For example, our Java implementation of the vector sum method takes 0.30 ms for 1280 points; our method takes 0.35 ms for the same data. Additional runtime data is shown in Fig. 7.

We attribute the similarity in performance (despite the difference in asymptotic costs) to two factors: the vector sum operation uses relatively expensive trigonometric operations, whereas our proposed method uses only simple arithmetic operations. The sorting method used is the standard Arrays.sort method, which also relies only on simple comparisons, and is well optimized.

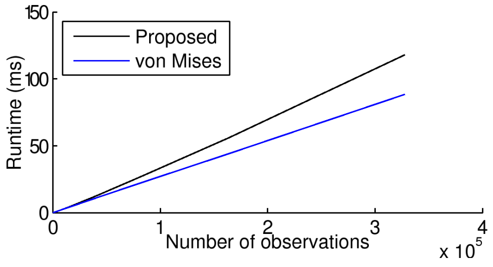

Fig. 7. Computational complexity. Despite having different asymptotic complexities, the runtimes of the two methods are similar.

V. Conclusion

This paper considers the problem of computing average orientation, a problem made difficult by the periodicity of the data. In addition to providing an introduction to the commonly-used vector sum product, we provide a new geometric interpretation for that method. We also propose and evaluate a new method that better approximates the ideal wrapped normal distribution.

Implementations of our algorithm are available from our web site http://april.eecs.umich.edu/. This research was supported by U.S. DoD grant W56HZV-04-2-0001.

References

[1] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2262–2269.
[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.
[3] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.
[4] R. Zlot and M. Bosse, "Place recognition using keypoint similarities," in *in 2D lidar maps, in International Symposium on Experimental Robotics*, 2008.
[5] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, Aug. 2006.
[6] J. Borenstein and L. Feng, "Gyrodometry: A new method for combining data from gyros and odom etry in mobile robots," in *In Proceedings of the 1996 IEEE International Conference onRobotics and Automation*, 1996, pp. 423–428.
[7] E. Hunt and D. Waller, "Orientation and wayfinding: A review," Tech. Rep., 1999.
[8] H. Choo and S. Franconeri, "Perception of average orientation," *Journal of Vision*, vol. 09, no. 8, 2009.
[9] S. R. Jammalamadaka and A. SenGupta, *Topics in Circular Statistics (Multivariate Analysis Vol. 5)*. World Scientific, 2001.
[10] K. V. Mardia and P. E. Jupp, *Directional Statistics*. Wiley, 2000.
[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.