Supplementary File of

"GridShift: A Faster Mode-seeking Algorithm for Image Segmentation and Object Tracking"

Abhishek Kumar[1], Oladayo S. Ajani[1], Swagatam Das[2], and Rammohan Mallipeddi[1]
[1]Department of Artificial Intelligence, Kyungpook National University, Daegu 37224, South Korea
[2]Electronics and Communication Sciences Unit, Indian Statistical Institute Kolkata 700108, India
abhishek.kumar.eee13@itbhu.ac.in, oladayosolomon@gmail.com,
swagatamdas19@yahoo.co.in, mallipeddi.ram@gmail.com

1. Primary differences between MS++ and GS

The most recent attempt to speedup MS is MS++ proposed by Jang and Jiang [5]. The improvement process in MS++ involves 1) partitioning the input domain into a grid, 2) assigning each data point to its associated grid, and 3) searching for the estimated mode for each point from its grid as well as grids within its neighborhood. Although MS++ is more than 1000x faster than MS, Park [6] claimed that without parallel computing, MS++ is not sufficiently fast yet. Therefore, [6] proposed a variant of MS++ known as α-MS++, which combines the use of an auxiliary hash table, a speedup factor (α) to reduce the number of iterations required until convergence as well as seeking more accurate modes using more numbers of neighboring grid cells while reducing the size of grid cells to minimize the computational redundancy of MS++. Although the experimental results in MS++ and α-MS++ are considerably faster than MS and MS++, respectively, we will show that GS is still orders of magnitude faster than both MS++ and α-MS++. Although both GS and MS++ use grid-based neighborhood search, the basic framework of GS is different from MS++ in the following aspects:

i) MS++ updates the location of each data point by using the weighted mean of the data points of 1-neighboring grid cells. On the other hand, GS updates the centroid of each active data cell by using the weighted mean of the centroid of the 1-neighboring active grid cells.

ii) In MS++, the shifted location of all data points associated with the same grid cell has the same value at any particular iteration. In GS, the centroid can also be treated as the location of all data points associated with the same grid cells. However, these centroids' locations may differ from the shifted location calculated in MS++ as GS updates centroids in a sequential manner, i.e.,

$$\mathcal{S}^{(t)}(j) \leftarrow \frac{\sum_{(v\in\{-1,0,1\}^d)} w_v \mathcal{S}^{(t)}(j+v)}{\sum_{(v\in\{-1,0,1\}^d)} w_v}, \qquad (1)$$
$$\mathcal{S}^{(t+1)}(j) = \mathcal{S}^{(t)}(j),\ \forall j \in \{1,2,\ldots,k^{(t)}\}$$

However, MS++ updates in a parallel fashion which is equivalent to the following equation (in terms of Eqn. (1).

$$\mathcal{S}^{(t+1)}(j) \leftarrow \frac{\sum_{(v\in\{-1,0,1\}^d)} w_v \mathcal{S}^{(t)}(j+v)}{\sum_{(v\in\{-1,0,1\}^d)} w_v}. \qquad (2)$$

For better understanding, in Fig. 1, we show the shift of centroids for both cases on a sample of nine centroids. As shown in this figure, Eqn. 2 does not take the neighbor's updated location into account in the shifting of centroids. Alternatively, in Eqn. 1, the location of neighbors that have already been updated in the shifting sequence is used instead of their older location. In comparison with Eqn. 2, Eqn 1 provides better convergence of data points towards modes. We will add clearer explanations on this in the final version.

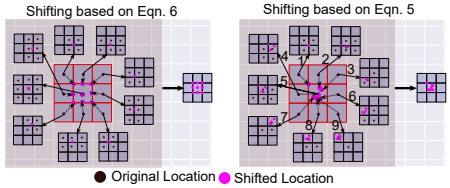

Figure 1. Shifting of centroids using Eqns. 5 and 6.

iii) The time complexity of MS++ is $\mathcal{O}(n3^d)$ per iteration, where n is the number of data points. On the other hand, the time complexity of the GS is $\mathcal{O}(m3^d)$ per iteration, where $m << n$ with the increase of iterations. Therefore, GS is faster than MS++ by n/m_{avg}

times theoretically, where m_{avg} is the average value of m throughout the iterations.

2. Theoretical Analysis of GridShift

In essence, Mean Shift is a clustering algorithm based on the following intuition. For dataset $X = \{x_i\}_{i=1}^n \subseteq \mathbb{R}^n$, let's assume the probability density function (PDF) is $p(z)$. In this dataset, we can expect k clusters if PDF $p(z)$ has k modes. Additionally, suppose an optimization algorithm, such as gradient descent, is run with a starting point x_m and converges to the j-th mode. In that case, it can be considered that the data point x_m is part of the cluster that belongs to j-th mode [4].

The PDF of datasets is not available in practice, and only the data points are accessible. To implement the intuition mentioned above, one must first estimate the PDF, a process known as density estimation [7]. Kernel density estimators (KDEs) are the most popular method for estimating density. Let $K(z)$ be the Kernal function satisfying

$$K(z) \geq 0, \text{ and } \int K(z)dz = 1, \tag{3}$$

its KDE is

$$\hat{p}(z) = \frac{1}{n}\sum_{i=1}^{n} K(z - x_i). \tag{4}$$

In mean shift (MS) algorithms, there are two main kernels that are employed: the Gaussian kernel

$$K_G(z; h) = c. \exp\left(-\frac{||z||^2}{2h^2}\right),$$

and the Epanechnikov kernel

$$K_E(z; h) = c. \max\left\{0, 1 - \frac{||z||^2}{h^2}\right\},$$

where c is the constant to ensure that the kernel will integrate to 1 [1–3].

By iterating through the following equation, initialized at each x_i, the MS algorithm attempts to estimate modes of $\hat{p}(z)$ based on the its KDE [2]:

$$z \leftarrow \frac{1}{\sum_{i=1}^{n} g\left(||z - x_i||^2\right)} \sum_{i=1}^{n} g\left(||z - x_i||^2\right) x_i \tag{5}$$

where $g(||z||^2) \propto -K'(||z||^2)$, i.e. kernel $K(||z||^2)$ is the shadow of the kernel $g(||z||^2)$.

Even though MS exists, GridShift (GS) uses grid-based KDE. The definition of grid-based KDE is as follows.

$$\hat{p}(z) = \frac{1}{n}\sum_{i=1}^{n} K_{GS}(z - x_i), \tag{6}$$

where

$$K_{GS}(z - x_i; h) = \begin{cases} c.\left(a - ||z - x_i||^2\right), & \text{if } \mathcal{M}(z, x_i; h) \leq 1 \\ c.a, & \text{otherwise,} \end{cases} \tag{7}$$

$$\mathcal{M}(z, x_i; h) = \max\left\{\left|\left\lfloor \frac{x_{i,1}}{h} \right\rfloor - \left\lfloor \frac{z_1}{h} \right\rfloor\right|, \ldots, \left|\left\lfloor \frac{x_{i,d}}{h} \right\rfloor - \left\lfloor \frac{z_d}{h} \right\rfloor\right|\right\}. \tag{8}$$

Here a and c are positive constants to satisfy kernel function conditions defined in Eqn 3. Therefore, $g(.)$ can be defined as follows:

$$g(||z - x_i||^2; h) = \begin{cases} 1, & \text{if } \mathcal{M}(z, x_i; h) \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

2.1. Function Analysis

GS attempts to find local maxima (modes) of the KDE $\hat{p} = \sum K_{GS}(z - x_i; h)$. We omit constants and scaling to define functions ϕ and f instead of K_{GS} and $\hat{p}$ since they do not affect optimization:

$$\begin{aligned} \phi(z - x_i) &= \begin{cases} ||z - x_i||^2, & \text{if } \mathcal{M}(z, x_i; h) \leq 1 \\ a, & \text{otherwise} \end{cases} \\ f(z) &= \sum_{i=1}^{n} \phi(z - x_i). \end{aligned} \tag{10}$$

A mode of $\hat{p}$ corresponds to the local minima of $f(z)$. Let's examine the properties of the loss function $f(z)$.

Lemma 1. *Let us define $\mathcal{P}(z) = \{i : \mathcal{M}(z, x_i; h) < 1\}$, then we get*

$$\begin{aligned} \nabla f(z) &= \sum_{i \in \mathcal{P}(z)} 2(z - x_i) \\ \nabla^2 f(z) &= 2|\mathcal{P}(z)|I. \end{aligned} \tag{11}$$

If $\mathcal{P}(z)$ is not an empty set, then $f(z)$ is strongly convex; otherwise, it is locally convex.

Proof. Here, the function $f(z)$ is local convex because of $\nabla^2 f(z) \geq 0$. Further, if $|\mathcal{P}(z)| \neq \emptyset$, then $\nabla^2 f(z) \geq I$, which means that $f(z)$ is strongly convex locally. □

Lemma 2. *If a point z^* is a local minimum for $f(z)$, then $\mathcal{P}(z^*) \neq \emptyset$ and $z^* = \frac{1}{|\mathcal{P}(z^*)|}\sum_{i \in \mathcal{P}(z^*)} x_i$.*

Proof. In order to be stationary, a point z^* must meet the following criteria:

$$\begin{aligned} &\nabla f(z^*) = 0 \\ \Rightarrow &z^* = \frac{1}{|\mathcal{P}(z^*)|} \sum_{i \in \mathcal{P}(z^*)} x_i \end{aligned} \tag{12}$$

If $\mathcal{P}(z^*) \neq \emptyset$, then $\nabla^2 f(z^*) > 0$; therefore, z^* is a local minimum. In the case of $\mathcal{P}(z^*) \neq \emptyset$, $f(z^*) = n.a$ (a global maximum). □

Definition 1. *If two points y and z lie in the same grid cell, i.e. $\lfloor \frac{y}{h} \rfloor = \lfloor \frac{z}{h} \rfloor$, then*

$$y^* = z^* = \frac{1}{|\mathcal{P}(z^*)|} \sum_{i \in \mathcal{P}(z^*)} x_i. \tag{13}$$

Therefore, all the points within a grid cell have the same local minima.

Due to the same $\mathcal{P}(.)$ value for all grid cell points, these points have the same local minima. This property of KDE K_{GS} motivates us to develop a new framework, GridShift (GS), faster than the original MS. In GS, we called the local minima of a grid cell the centroid. Within each iteration, centroids (local minima) are updated. Utilizing the centroids of the previous iteration, we update these centroids.

2.2. Convergence Guarantee

Let us define a mapping $g^{(t)} : X \leftarrow \mathcal{C}^{(t)}$ for any dataset $X(= \{x_1, x_2, \ldots, x_n\}) \in \mathbb{R}^d$, such that each data point $x_i \in X$ is assigned to one of the $k^{(t)}$ active grid cells (clusters) $c_i^{(t)} \in \mathcal{C}^{(t)}$. Therefore,

$$\begin{aligned} &\mathcal{C}^{(t)} = \{c_1^{(t)}, c_2^{(t)}, \ldots, c_{k^{(t)}}^{(t)}\}, \text{ and} \\ &c_i^{(t)} \cap c_j^{(t)} = \phi, \forall i, j \in \{1, 2, \ldots, k^{(t)}\},\ i \neq j. \end{aligned} \tag{14}$$

Here, each active cell $c_i^{(t)}$ has a set of 1-neighboring active grid cells, $\mathcal{P}_{c_i}^{(t)} \subseteq \mathcal{C}^{(t)}$.

Corollary 1. *The value of $f(r_i)$ obtained by GS is strictly decreasing unless $r_i^{(t)} = r_i^{(t-1)}$.*

Proof. In GS, we update

$$r_j^{(t)} = \frac{\sum_{i \in \mathcal{P}(r_j^{(t-1)})} m_i^{(t-1)} r_i^{(t-1)}}{\sum_{i \in \mathcal{P}(r_j^{(t-1)})} m_i^{(t-1)}}, \tag{15}$$

where m_i represents number of data points resident in ith grid cell. At a particular point $\tilde{z}$, define $\overline{f}(z|\tilde{z})$ using following equation.

$$\overline{f}(z|\tilde{z}) = \sum_{i \in \mathcal{P}(\tilde{z})} m_i \|z - r_i\|^2 + \left(n - \sum_{i \in \mathcal{P}(\tilde{z})} m_i\right) a. \tag{16}$$

Then,

$$\begin{aligned} &f(r_j^{(t-1)}) - f(r_j^{(t)}) \\ &\geq \overline{f}(r_j^{(t-1)}|r_j^{(t-1)}) - \overline{f}(r_j^{(t)}|r_j^{(t-1)}) \\ &= \sum_{i \in \mathcal{P}(r_j^{(t-1)})} m_i \|r_j^{(t-1)} - r_i^{(t-1)}\|^2 - \sum_{i \in \mathcal{P}(r_j^{(t-1)})} m_i \|r_j^{(t)} - r_i^{(t-1)}\|^2 \\ &= \sum_{i \in \mathcal{P}(r_j^{(t-1)})} m_i \left(\|r_j^{(t-1)} - r_i^{(t-1)}\|^2 - \|r_j^{(t)} - r_i^{(t-1)}\|^2 \right) \\ &= \left(\sum_{i \in \mathcal{P}\left(r_j^{(t-1)}\right)} m_i \right) \|r_j^{(t-1)} - r_j^{(t)}\|^2 > 0 \end{aligned} \tag{17}$$

Therefore, $f(r_j^{(t-1)}) > f(r_j^{(t)})$, unless $f(r_j^{(t-1)}) = f(r_j^{(t)})$ for $r_j^{(t)} = r_j^{(t-1)}$. □

Theorem 1. *For any given dataset $X \in \mathbb{R}^d$, the $\{\mathcal{C}^{(t)}\}_{t=1,2,\ldots}$ estimated by successive proposed grid cells shifts attains convergence, i.e. $\mathcal{C}^{(i)} == \mathcal{C}^{(i++)}$, where i is a finite number.*

Proof. From corollary 1, since the value of $f(r)$ is monotonically non-increasing, GS attains convergence to the local minima of function defined in Eqn. (10). As we know that Set $\mathcal{C}$ contains the centroid of active grid cells (local minima). Therefore, sequence $\{\mathcal{C}^{(t)}\}_{t=1,2,\ldots}$ attains convergence.

From Eqn. (17), we can have

$$f(r_j^{(t-1)}) - f(r_j^{(t)}) \geq \|r_j^{(t-1)} - r_j^{(t)}\|^2, \tag{18}$$

After summing both sides for $t = 1, \ldots, i$, we get

$$f(r_j^{(0)}) - f(r_j^{(i)}) \geq \sum_{t=1}^{i} \|r_j^{(t-1)} - r_j^{(t)}\|^2. \tag{19}$$

As we know, the right-hand side of the above equation is positive unless convergence is attained. if we calculate the maximum value of $\lambda > 0$ such that

$$\|r_j^{(t-1)} - r_j^{(t)}\| \geq \lambda > 0,\ \forall t = 1, \ldots, i, \tag{20}$$

then

$$i \leq \frac{f(r_j^{(0)}) - f(r_j^{(i)})}{\lambda} \tag{21}$$

which is a finite number. □

Theorem 2. *For any X, there exists $T \in \mathbb{N}$ such that $\mathcal{Q}_{c_i}^{(t)} = c_i^{(t)}$, $\forall i \in \{1, 2, \ldots, k^{(t)}\}$ for all $t \geq T$.*

Proof. As we know, at convergence, we have

$$r_j^{(t+1)} = \frac{\sum_{i \in \mathcal{P}(r_j^{(t)})} m_i^{(t)} r_i^{(t)}}{\sum_{i \in \mathcal{P}(r_j^{(t)})} m_i^{(t)}} = r_j^{(t)}, \tag{22}$$

that implies $\mathcal{P}(r_j^{(t)}) = j$, i.e. $\mathcal{Q}_{c_i}^{(t)} = c_i^{(t)}$. □

The above two theorems confirm that GS attains convergence after a finite number of iterations when the active grid cells do not have any other active members in their 1-neighborhood to update their attributes further.

2.3. Convergence Rate

In this subsection, we analyze the behavior of GS on mode seeking of a dataset sampled from a Gaussian distribution. We will prove that the number of active grid cells will form a non-increasing sequence, and centroids of these active grid cells will shrink towards the mean of the distribution with at least a cubic convergence rate.

Let $\phi(x; \mu, \Sigma)$ denotes a Gaussian probability density function, where μ and Σ are the mean and dispersion matrix of the density function, respectively. To remove the dependency on the random process, we consider infinite samples generated from density $q(x) = \phi(x; 0, diag(s_1^2, s_2^2, \ldots, s_d^2))$.

Theorem 3. *For dataset* $X = \{x_1, x_2, \ldots, x_n\}$ *where* $x_i \sim \mathcal{N}(0, diag(s_1^2, \ldots, s_d^2))$, *let centroids* $\{c_i^{(t+1)}\}_{i=1}^{k^{(t+1)}} \sim \int y p^{(t)}(y|c)dy$, *where* $p^{(t)}()$ *represents the distribution of* $\{c_i^{(t+1)}\}_{i=1}^{k^{(t+1)}}$ *and* $p^{(t)}(y|z) = k(z - y)q^{(t)}(y)/p^{(t)}(z)$. *Then (i)* $\{c_i^{(t+1)}\}_{i=1}^{k^{(t+1)}} \sim \mathcal{N}\left(0, diag\left(\left(s_1^{(t+1)}\right)^2, \ldots, \left(s_d^{(t+1)}\right)^2\right)\right)$, *with* $s_j^{(t+1)} = \left(1 + 2.25\frac{h^2}{s_j^2}\right)^{-1} s_j^{(t)}$ *and (ii)* $k^{(t+1)} = \prod_{j=1}^{d}\left(\left\lfloor \frac{6s_j^{(t+1)}}{h} \right\rfloor + 1\right)$, *where* $\{k^{(t)}\}_{t=1}^{\infty}$ *is non-decreasing sequence that converges to* 1.

Proof. To estimate the distribution of $\{c_i^{(t+1)}\}_{i=1}^{k^{(t+1)}}$, we have

$$p^{(1)}(y|c^{(0)}) \propto \exp\left\{-\frac{1}{2}\frac{\left\|y - \frac{c^{(0)}/(2.25h^2)}{/(2.25h^2)+1/(s^{(0)})^2}\right\|^2}{\frac{1}{1/(2.25h^2)+1/(s^{(0)})^2}}\right\}. \tag{23}$$

As we know,

$$c^{(1)} = E(y|x^{(0)}) = \left(\frac{c_1^{(0)}(s_1^{(0)})^2}{(s_1^{(0)})^2 + 2.25h^2}, \ldots, \frac{c_d^{(0)}(s_d^{(0)})^2}{(s_d^{(0)})^2 + 2.25h^2}\right). \tag{24}$$

Therefore, $c^{(1)}$ is also a Gaussian distribution with mean zero and standard deviation $s^{(1)} = \frac{(s^{(0)})^3}{(s^{(0)})^2+2.25h^2}$, which implies

$$s_j^{(t+1)} = \frac{(s_j^{(t)})^3}{(s_j^{(t)})^2 + 2.25h^2} = \left(1 + 2.25\frac{h^2}{(s_j^{(t)})^2}\right)^{-1} s_j^{(t)} \tag{25}$$

Thus, standard deviation is decreasing with increase of iteration and become zero at convergence. We can estimate the number of active grid cells according to standard deviation of this distribution as follows.

$$k^{(t+1)} = \prod_{j=1}^{d}\left(\left\lfloor \frac{6s_j^{(t+1)}}{h} \right\rfloor + 1\right). \tag{26}$$

We see that $k^{(t)}$ is a non-decreasing sequence and converges to 1 when $s_j^{(t+1)}$ becomes 0. □

3. Dataset

S.N	Dataset	n	d	k
1.	Phone Gyroscope	13932632	3	7
2.	Phone Accelerometer	13062475	3	7
3.	Watch Accelerometer	3540962	3	7
4.	Watch Gyroscope	3205431	3	7
5.	Still	949983	3	6
6.	Skin	245057	3	2
7.	Wall Robot	5456	4	4
8.	Sleep Data	1024	2	2
9.	Balance Scale	625	4	3
10.	User Knoweldge	403	5	5
11.	Vinnie	380	2	2
12.	PRNN	250	2	2
13.	Iris	150	4	3
14.	Transplant	131	3	2

Table 1. Brief summary of datasets used in experiment. n: number of data points, d: number of features, and k: number of clusters.

References

[1] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995. 2

[2] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 2

[3] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975. 2

[4] Kejun Huang, Xiao Fu, and Nicholas D Sidiropoulos. On convergence of epanechnikov mean shift. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[5] Jennifer Jang and Heinrich Jiang. Meanshift++: Extremely fast mode-seeking with applications to segmentation and object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4102–4113, 2021. 1

[6] Hanhoon Park. α-meanshift++: Improving meanshift++ for image segmentation. *IEEE Access*, 9:131430–131439, 2021. 1

[7] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015. 2