

เอกสารอ้างอิง

1. Fukunaga K., Hostetler LD. The estimation of the gradient of a density function, with applications in pattern recognition. **IEEE Transactions on Information Theory** 1975; 21: 32 – 40.
2. Cheng Y. Mean shift, mode seeking, and clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence** 1995 August; 17: 790–799.
3. Comaniciu D, Meer P. Mean shift analysis and applications. **Proceedings of Int'l Conf. Computer Vision**; 1999. p. 1197–1203.
4. Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. **IEEE Transactions on Pattern Analysis and Machine Intelligence** 2002 May; 24: 603 – 619.
5. DeMenthon D. Spatio-temporal segmentation of video by hierarchical mean shift analysis. **Statistical Methods in Video Processing Workshop**; June 1–2; Copenhagen, Denmark; 2002. p. 1–6.
6. Carreira-Perpiñán MÁ. Fast nonparametric clustering with Gaussian blurring meanshift. **Proceedings of the 23rd Int. Conf. on Machine learning**; Pennsylvania; 2006. p. 153–160.
7. Lerner E. **Edge Detection using Mean Shift Smoothing** [Online] 2005 [cited 2009 November 09]. Available from <http://www.cs.bgu.ac.il/~ben-shahar/Teaching/ComputationalVision/-StudentProjects.php>
8. Fashing M, Tomasi C. Mean shift is a bound optimization. **IEEE Transactions on Pattern Analysis and Machine Intelligence** 2005 March; 27(3): 471–474.
9. Padungweing P, Chiewchanwattana S, Sunat K. Resilient Mean Shift Algorithm. **13th Int. Conf. on Systems, Signals and Image Processing**; Sep 21–23; Hungary; 2006. p. 181–184.
10. Zhang K, Kwok JT, Tang M. Accelerated Convergence Using Dynamic Mean Shift. **In the 9th European Conference on Computer Vision (ECCV 2006)**; May; Graz, Austria; 2006. p. 257–268.
11. Carreira-Perpiñán MÁ. Acceleration Strategies for Gaussian Mean-Shift Image Segmentation. **Proc. of the 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition**; 2006. p. 1160 – 1167.
12. Liying Z, Jingtao Z, Qianyu W. Mean-shift-based color segmentation of images containing green vegetation. **Computers and Electronics in Agriculture** 2009; 65: 93–98.
13. สิทธิโชค ยอดระยับ. การเขียนโปรแกรม Digital Image Processing ด้วย Visual Basic. กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น); 2550.
14. ปุณณรัตน์ พิชญ์ไพบุลย์. Computer Graphic:สำหรับการออกแบบ. กรุงเทพฯ: ด่านสุทธาการพิมพ์; 2542.

15. อีรยุทธ สว่างศรี. การแยกส่วนภาพวัตถุวิดีโอที่เลือกบริเวณที่สนใจได้บนพื้นฐานองค์ประกอบสีและเทคนิคเชิงสัญญาณ [วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า]. กรุงเทพฯ: คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย; 2548.
16. มนตรี กาญจนเดชะ. Image processing [ไม่ได้ตีพิมพ์]. เอกสารประกอบการสอน ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์: มหาวิทยาลัยสงขลานครินทร์; 2548.
17. นวภัค เอื้ออนันต์. Image segmentation [ไม่ได้ตีพิมพ์]. เอกสารประกอบการสอน ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์: มหาวิทยาลัยขอนแก่น; 2545.
18. นภัสมล โมงเย็น. การตรวจจับความเร็วรถยนต์แบบทันกาลในเวลากลางคืนโดยใช้การประมวลผลภาพวีดิทัศน์ [วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์]. กรุงเทพฯ: คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย; 2549.
19. การแบ่งกลุ่มข้อมูล [ออนไลน์] 2550 [อ้างเมื่อ 1 กรกฎาคม 2552]. จาก <http://th.wikipedia.org/wiki/การแบ่งกลุ่มข้อมูล>
20. โอภาส เอี่ยมสิริวงศ์. โครงสร้างข้อมูล(Data structures)เพื่อการออกแบบโปรแกรมคอมพิวเตอร์. กรุงเทพฯ: ซีเอ็ดดูเคชั่น; 2549.
21. Tarjan RE. Depth first search and linear graph algorithms. **SIAM Journal on Computing** 1972; 1(2): 146-160.
22. การค้นหาแบบลึกก่อน [ออนไลน์] 2551 [อ้างเมื่อ 1 กรกฎาคม 2552]. จาก <http://202.28-.94.55/web/320417/2548/work1/g25/technoreport1.htm>
23. Carreira-Perpiñán MÁ. Mode finding for mixtures of Gaussian distributions. **IEEE Transactions on Pattern Analysis and Machine Intelligence** 2000 Nov; 22: 1318-1323.
24. การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธี [ออนไลน์] 2547 [อ้างเมื่อ 12 กันยายน 2552]. จาก http://www.geocities.com/kua_sa/Chapte2.ppt
25. Bunting P, Lucas R. The delineation of tree crowns in Australian mixed species forests using hyperspectral compact airborne spectrographic imager (CASI) data. **Remote Sensing of Environment** 2006; 101(2): 230-248.
26. Gee Ch, Bossu J, Jones G, Truchetet F. Crop/weed discrimination in perspective agronomic images. **Computers and Electronics in Agriculture** 2008; 60(1): 49-59.

ภาคผนวก

ภาคผนวก ก

บทความที่ผ่านการนำเสนอแบบปากเปล่า และตีพิมพ์ในวารสาร

รายงานการประชุมวิชาการ

ในงาน KST 2009 จัดขึ้นที่ มหาวิทยาลัยบูรพา จังหวัดชลบุรี

Transport Mean Shift Algorithm ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย

พนิดา ผดุงเวียง¹, สิริภัทร เชี่ยวชาญวัฒน์² และ คำรณ สุนัตติ³

^{1,2} ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น

³ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร

¹panida_k@windowslive.com, ²sunkra@kku.ac.th and ³khamron_sunat@yahoo.com

บทคัดย่อ

การแยกส่วนประกอบของภาพและการจัดกลุ่มข้อมูล เป็นกระบวนการที่สำคัญในวิเคราะห์และทำความเข้าใจข้อมูล ขั้นตอนวิธีการย้ายค่าเฉลี่ยเป็นขั้นตอนหนึ่งที่น่าสนใจนำมาใช้ในการแก้ปัญหาที่ อย่างไรก็ตามเนื่องจากการทำงานที่ซ้ำซ้อน จึงทำให้ต้องใช้เวลาในการดำเนินการตามขั้นตอนวิธีสูง บทความนี้ได้นำเสนอการลดจำนวนข้อมูลในแต่ละรอบโดยให้ข้อมูลที่มีจุดหมายเดียวกันถูกลำเลียงไปโดยใช้ตัวแทน ซึ่งทำให้การคำนวณลดลงอย่างมาก โดยตัวแทนนี้ยังคงดำเนินการตามขั้นตอนวิธีการย้ายค่าเฉลี่ย ทำให้ผลลัพธ์ยังคงเหมือนกับการใช้ขั้นตอนวิธีการย้ายค่าเฉลี่ยเดิม การหาส่วนของกราฟที่เชื่อมถึงกันได้ถูกนำมาใช้ในการจัดกลุ่มข้อมูล และแยกส่วนประกอบของภาพ ขั้นตอนวิธีที่นำเสนอขึ้นนี้ เรียกว่าขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย ผลจากทดลองจัดกลุ่มข้อมูลและการแยกส่วนประกอบของภาพพบว่า สามารถเพิ่มความเร็วได้ถึง 7 – 70 เท่าเมื่อเทียบกับขั้นตอนการย้ายค่าเฉลี่ยเดิม

คำสำคัญ: ขั้นตอนวิธีการย้ายค่าเฉลี่ย, ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย, การจัดกลุ่มข้อมูล, การแยกส่วนประกอบของภาพ, ส่วนของกราฟที่เชื่อมถึงกัน

Abstract

Image segmentation and clustering are important process in data analysis and the nature discovery of the given data. To acquire the solution, the Mean shift algorithm is the one among the first-thought techniques. However, it's a lot of repeating action that causes a very high computational time. In this paper proposes the algorithm called Transport Mean Shift algorithm (TMS). The idea is the data having the same mode will be reduced and transported by the same leader. The leaders are further computed using the standard mean shift algorithm. Thus, in the next iteration of performing, only a few leaders will be evaluated by the algorithm. This can significantly improve the computational time of the algorithm. Graph connected component is the tool for clustering and segmenting the final data. The TMS was tested on the clustering and image segmentation problems. The experimental results on six data set showed that our algorithm achieves 7 – 70 times faster than the standard mean shift algorithm.

Key Words: mean shift algorithm, Transport Mean Shift algorithm, clustering, image segmentation, Graph connected component

1. บทนำ

ขั้นตอนวิธีการย้ายตามค่าเฉลี่ย (Mean-Shift algorithm) เดิมได้ถูกนำเสนอในชื่อว่า "valley -seeking procedure"[1] ต่อมาได้มีการศึกษาอัลกอริทึมเพื่อนำมาใช้วิเคราะห์ภาพ [2] สามารถช่วยในการแยกองค์ประกอบภาพ (image segmentation) [3,4] รวมทั้งสามารถใช้ใน

การติดตามภาพเคลื่อนไหว(tracking)[5] ปัจจุบันมีการนำขั้นตอนวิธีการย้ายตามค่าเฉลี่ยมาประยุกต์ใช้ในหลายด้าน ไม่ว่าจะเป็นการแยกองค์ประกอบของภาพ(image segmentation) การค้นหาข้อมูลในภาพ(tracking) การวิเคราะห์และจัดกลุ่มข้อมูล (Analysis and Clustering)

ขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเป็นวิธีหาจุดที่มีความหนาแน่นของข้อมูลที่มีมากที่สุด โดยเริ่มจากการคำนวณค่าความหนาแน่นของทุกจุดเทียบกับจุดพิจารณา แล้วคำนวณค่าเฉลี่ยเพื่อหาตำแหน่งการย้ายใหม่ การเปลี่ยนแปลงนี้จะเป็นลักษณะการเดินทางตามเส้นทางไปยังตำแหน่งที่มีความหนาแน่นของข้อมูลมากขึ้นจนกระทั่งตำแหน่งการย้ายที่เกิดจากการคำนวณ ไม่มีการเปลี่ยนตำแหน่งหรือเปลี่ยนแปลงน้อยกว่าค่าที่ยอมรับได้ ก็จะถือว่าตำแหน่งที่ทำการย้ายมานั้นเป็นจุดที่มีความหนาแน่นของข้อมูลสูงสุด และจุดที่ลู่เข้าไปที่จุดเดียวกันจะถือว่าอยู่ในกลุ่มเดียวกัน อย่างไรก็ตามขั้นตอนวิธีการย้ายตามค่าเฉลี่ยนี้จะทำงานช้า โดยในแต่ละรอบจะมีการทำงานเป็น $O(kn^2m)$ [6] ซึ่งยังไม่เหมาะสมที่จะนำมาใช้กับการใช้งานจริง โดยเฉพาะกับการแยกองค์ประกอบของภาพที่ปัจจุบันมีจำนวนจุดของข้อมูลสูงมาก การเพิ่มความเร็วของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยจึงมีความสำคัญ มีการนำเสนอการเพิ่มความเร็วขั้นตอนวิธีการย้ายตามค่าเฉลี่ยด้วยเทคนิคของเครือข่ายประสาทเทียม [7] ทำให้ขั้นตอนนี้ใช้จำนวนรอบน้อยลง และยังคงมีความถูกต้องเมื่อเทียบกับวิธีการเดิม หลังจากนั้น มีการนำเสนอการเพิ่มความเร็วของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบพิจารณาตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้า[6] โดยมีการจัดกลุ่มของข้อมูลในแต่ละรอบและตัดข้อมูลกลุ่มที่มีสมาชิกบางตัวย้ายไปตำแหน่งสูงสุดแล้ว ทำให้ขั้นตอนต่อไปมีจำนวนข้อมูลน้อยลง การคำนวณจึงเร็วขึ้นตามไปด้วย และมีการนำเสนอการลดจำนวนข้อมูลด้วยการจัดกลุ่มของข้อมูลเพื่อหาตัวแทนก่อนทำขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบพิจารณาตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้า [8] วิธีนี้ให้ผลเร็วแต่มีความคลาดเคลื่อนสูง อย่างไรก็ตามขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบพิจารณา

ตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้าจะให้ผลลัพธ์คลาดเคลื่อนจากขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบไม่พิจารณาตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้า ซึ่งบทความฉบับนี้จะเรียกว่าขั้นตอนวิธีการย้ายตามค่าเฉลี่ยต่อมา มีการนำเสนอการเพิ่มความเร็วขั้นตอนวิธีการย้ายตามค่าเฉลี่ยสำหรับการแยกส่วนประกอบของภาพ [9] วิธีคือ หนึ่งรวมตำแหน่งของภาพที่อยู่ใกล้กันเป็นเซลล์เดียวกัน จากนั้นใช้วิธีการย้ายตามค่าเฉลี่ย และเมื่อเซลล์ย้ายไปยังเซลล์ที่มีการย้ายตำแหน่งไปก่อนหน้าแล้ว ก็จะหยุดคำนวณ และถือว่าเป็นกลุ่มเดียวกัน วิธีที่สองหาตัวแทนของจุดที่อยู่ใกล้กันในโดเมนของภาพกำหนดโดยระยะห่างคงที่ค่าหนึ่ง แล้วดำเนินการคล้ายกับวิธีแรก ทั้งสองวิธีได้ผลเร็วมาก แต่ความคลาดเคลื่อนจะเกิดขึ้นตั้งแต่ตอนแรกที่มีการรวมกลุ่มของจุดบนภาพ อีกสองวิธีเป็นการลดการคำนวณของฟังก์ชันหาความหนาแน่นและใช้วิธีลู่เข้าแบบกำลังสี่ทำให้จำนวนรอบในการย้ายน้อยกว่า แต่ใช้การคำนวณที่สูงมากจึงทำให้ช่วยเพิ่มความเร็วไม่มากเท่าที่ควร ดังนั้นในบทความ นี้มีจุดประสงค์ที่จะนำเสนอขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยมาใช้ในการจัดกลุ่มข้อมูลและแบ่งส่วนประกอบของภาพ

บทความนี้มีแนวความคิดมาจากการลดการย้ายตำแหน่งที่ซ้ำซ้อนของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิม ซึ่งมีข้อมูลจำนวนมากที่ต้องย้ายตำแหน่งไปในเส้นทางเดียวกันหรือใกล้เคียง สุดท้ายตำแหน่งเหล่านี้ก็จะย้ายไปยังตำแหน่งที่มีความหนาแน่นสูงสุดตำแหน่งเดียวกัน ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยที่นำเสนอ ได้ทำการลดจุดเหล่านี้ โดยถือว่าเป็นจุดที่ต้องถูกลำเลียง โดยอาศัยตัวแทนตัวเดียวกันจะช่วยให้การคำนวณในรอบต่อไปมีจำนวนตำแหน่งที่ต้องคำนวณน้อยลง

ส่วนที่ 2 จะกล่าวถึงทฤษฎีโดยสรุปของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยจะถูกนำเสนอในส่วนที่ 3 จากนั้นวิธีการทดลองและผลการทดลองจะกล่าวถึงในส่วนที่ 4 ส่วนที่ 5 เป็นสรุปผลการทดลอง

2. ทฤษฎีที่เกี่ยวข้อง

2.1 ขั้นตอนวิธีการย้ายตามค่าเฉลี่ย(Mean-Shift algorithm)

กำหนดให้ข้อมูลนำเข้าเป็น $\mathbf{X} \in R^m$ โดยที่ $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ และ $x_i = [x_{1,i}, x_{2,i}, \dots, x_{m,i}]^T$ การประมาณความหนาแน่นของข้อมูล ณ ตำแหน่ง \mathbf{x} ใด ๆ สามารถคำนวณได้ดังสมการ

$$p(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}\right) \quad (1)$$

โดยที่ σ เป็นค่าคงที่และ $K(t)$ เป็นฟังก์ชันเคอร์เนล ตำแหน่งที่มีความหนาแน่นสูงสุดของบริเวณที่พิจารณา คือตำแหน่งที่ $\nabla p(\mathbf{x}) = 0$ ซึ่งสามารถหาได้โดยการคำนวณเพื่อย้ายตำแหน่งหลาย ๆ รอบโดยการย้ายในแต่ละรอบคำนวณได้ดังสมการ

$$\mathbf{x}^{(r+1)} = f(\mathbf{x}^{(r)}) \quad (2)$$

โดยที่

$$f(\mathbf{x}) = \frac{\sum_i^n K'\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma}\right) \mathbf{x}_i}{\sum_{j=1}^n K'\left(\frac{\|\mathbf{x} - \mathbf{x}_j\|^2}{\sigma}\right)} \quad (3)$$

เมื่อ $K'(t) = \frac{\partial K(t)}{\partial t}$ ขั้นตอนวิธีนี้เรียกว่า ขั้นตอนวิธีการย้ายตามค่าเฉลี่ย จาก [10] ถ้าฟังก์ชันเคอร์เนลเป็นฟังก์ชันเกาส์เซียน ($K(t) = e^{-t^2}$) สมการที่ (2), (3) จะเป็นดังสมการ

$$\mathbf{x}^{(r+1)} = \sum_{i=1}^n p(i | \mathbf{x}^{(r)}) \mathbf{x}_i \quad (4)$$

$$p(i | \mathbf{x}^{(r)}) = \frac{\exp\left(-\frac{1}{2} \frac{\|\mathbf{x}^{(r)} - \mathbf{x}_i\|^2}{\sigma}\right)}{\sum_{j=1}^n \exp\left(-\frac{1}{2} \frac{\|\mathbf{x}^{(r)} - \mathbf{x}_j\|^2}{\sigma}\right)} \quad (5)$$

เมื่อสิ้นสุดขั้นตอนวิธีการย้ายตามค่าเฉลี่ย ทุกตำแหน่งจะถูกย้ายไปที่ตำแหน่งที่มีความหนาแน่นสูงสุดในบริเวณของตำแหน่งนั้น หลังจากนั้นก็จะสามารถดำเนินการจัดกลุ่มข้อมูลโดยใช้ขั้นตอนวิธีในการจัดกลุ่มต่อไป

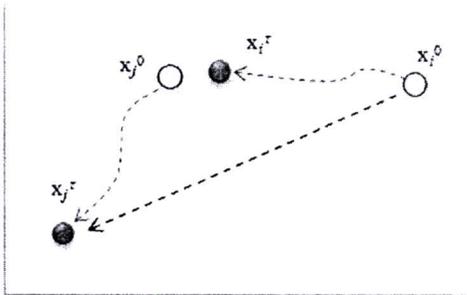
2.2 Graph Connect Component

คุณสมบัติการเชื่อมต่อกันของกราฟสามารถนำมาใช้ในการจัดกลุ่มข้อมูลได้ สำหรับกราฟแบบไม่มีทิศทางกราฟา ส่วนของกราฟที่ทุก ๆ โหนดมีเส้นทาง (Path) เชื่อมต่อกัน (Graph Connect Component) แต่ละโหนดของส่วนของกราฟนี้สามารถแทนด้วยตำแหน่งของจุด ดังนั้นส่วนของกราฟที่เชื่อมต่อกัน จึงถือได้ว่าเป็นกลุ่มหนึ่งกลุ่มให้ n เป็นจำนวนโหนด และ E เป็นจำนวนเส้นเชื่อม การหาส่วนของกราฟที่เชื่อมต่อกัน สามารถหาได้โดยใช้ขั้นตอนวิธีแบบค้นหาเชิงลึก ซึ่งวิธีนี้จะใช้เวลาเป็น $O(n+E)$ [11]

การหาส่วนของกราฟที่เชื่อมต่อกันนี้ได้ถูกนำมาใช้ในการจัดกลุ่มข้อมูล หลังจากดำเนินการตามขั้นตอนการย้ายค่าเฉลี่ยแล้ว ซึ่งตำแหน่งสุดท้ายของข้อมูลที่อยู่กลุ่มเดียวกันจะอยู่ในบริเวณใกล้เคียงกัน ดังนั้น เมื่อให้ข้อมูลเหล่านี้เป็นโหนด ข้อมูลที่อยู่ใกล้เคียงกันในระยะที่ยอมรับได้ จะถือว่ามีเส้นเชื่อมต่อกัน จากนั้นก็สามารถใช้ขั้นตอนวิธีการหาส่วนของกราฟที่เชื่อมต่อกันจัดกลุ่มข้อมูลเหล่านี้ได้

3. ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย (Transport Mean Shift Algorithm)

ในแต่ละรอบของขั้นตอนวิธีการย้ายตามค่าเฉลี่ย จะมีจุดที่จะย้ายไปในเส้นทางเดียวกับจุดอื่น ดังแสดงตัวอย่างในรูปที่ 1 ซึ่งในรอบการย้ายในรอบที่ τ นี้ ตำแหน่ง i ย้ายตำแหน่งไปยังจุดใหม่ ซึ่งย้ายมาใกล้ตำแหน่ง j จะเห็นว่าการคำนวณรอบต่อไปตำแหน่ง i จะย้ายไปตามลักษณะเดียวกับตำแหน่ง j ดังนั้นตำแหน่ง i ควรพิจารณาตามการย้ายของตำแหน่ง j เสมือนกับว่าให้ตำแหน่ง j เป็นตัวแทน ดังนั้นตำแหน่ง i ก็ไม่จำเป็นต้องคำนวณในรอบต่อไป



รูปที่ 1 แสดงการย้ายตำแหน่งของจุดสองจุดในรอบที่ r

ในบทความนี้ได้มีการนำเสนอขั้นตอนวิธีในการหาค่าตำแหน่ง i เพื่อลดการคำนวณในรอบถัดไป และตำแหน่ง j เพื่อใช้เป็นตัวแทนในการลำเลียงสมาชิก ซึ่งตำแหน่ง j ที่ใช้เป็นตัวแทนนี้จะดำเนินการตามขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม ดังนั้นตำแหน่งสุดท้ายของ j จะเป็นตำแหน่งเดียวกันกับขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม

กำหนดให้ G เป็นกราฟไม่มีทิศทาง ที่มีจำนวนโหนดเท่ากับ n แต่ละโหนดแทนตำแหน่งของข้อมูลซึ่งมีทั้งหมด n ตำแหน่ง กราฟ G จะมีเส้นเชื่อม (i, j) ก็ต่อเมื่อในรอบที่ r ตำแหน่ง x_i^r อยู่ใกล้กับ x_j^0 มากที่สุด จะเรียก j เป็นตำแหน่งตัวแทนของ i หรือ x_j^r อยู่ใกล้กับ x_i^0 มากที่สุดจะเรียก i เป็นตำแหน่งตัวแทนของ j ดังสมการ

$$\begin{aligned} i &= \arg \min_j (\|x_i^r - x_j^0\|^2), \\ j &= \arg \min_i (\|x_j^r - x_i^0\|^2) \end{aligned} \quad (6)$$

กำหนด $h \in N^1$ โดยที่ h_i เป็นไปตามกฎดังต่อไปนี้

1. $h_i = \arg \min_j (\|x_i^r - x_j^0\|^2)$, กฎเบื้องต้น
2. $1 \leq h_i \leq n$
3. $h_i = h_{h_i}$, กฎแห่งความต่อเนื่อง

จะเห็นว่า h_i คือตำแหน่งตัวแทนของตำแหน่ง i ที่ระบุโดยสมการที่ (6) เป็นไปตามกฎเบื้องต้น ซึ่งระบุค่า h_i ในตอนแรกของแต่ละรอบเท่านั้น ตำแหน่งใดที่มีการเชื่อมโยงถึงกันก็จะใช้ตัวแทนเดียวกัน เป็นไปตามกฎข้อที่ 3 คือ กฎแห่งความต่อเนื่อง โดยเมื่อพิจารณากฎข้อนี้แล้ว h_i ไม่จำเป็นต้องเป็นไปตามกฎเบื้องต้น โดยที่กราฟ G จะต้องมีเส้นเชื่อมทุกๆ (i, h_i) ในแต่ละรอบจะพิจารณาค่าตำแหน่งที่

$\nabla p(x) = 0$ โดยใช้เงื่อนไขการหยุดเช่นเดียวกับ ขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิมตามเงื่อนไข

$$\|x^r - x^{r-1}\|^2 \leq \text{threshold} \quad (7)$$

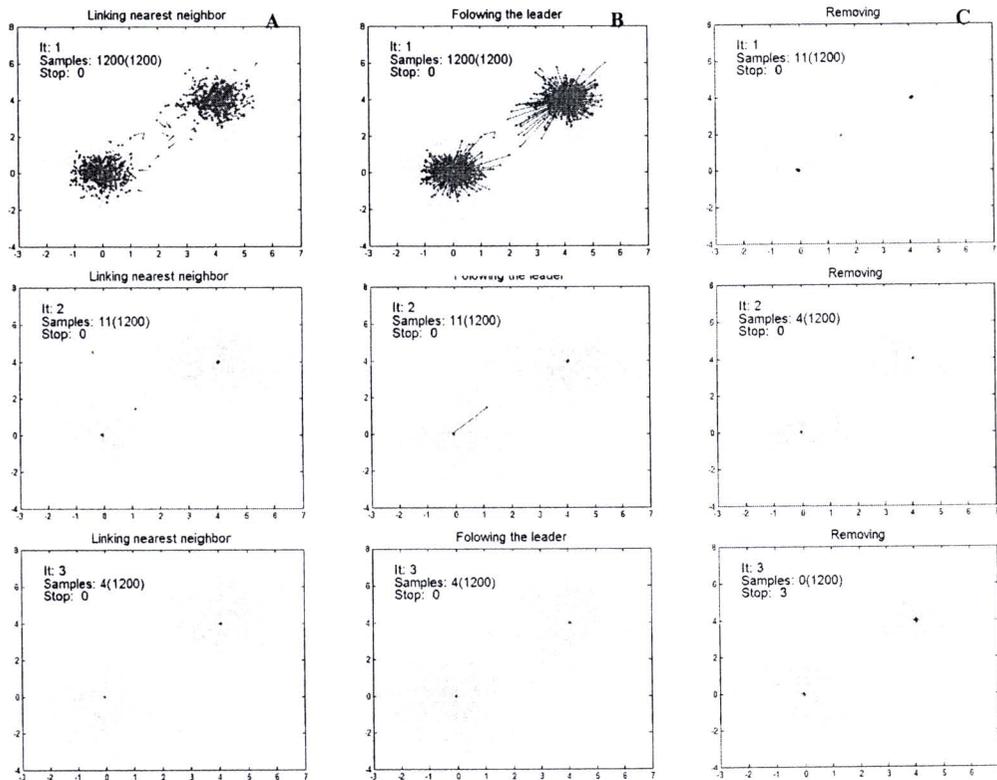
ในแต่ละรอบ ตำแหน่งที่หยุดแล้วจะไม่ถูกนำมาคำนวณหรือพิจารณาอีกในรอบต่อไป ให้ $X \in R^m$ แทนตำแหน่งที่หยุดแล้ว และ s_i คือ ตำแหน่งสุดท้ายของ x_i ซึ่งเป็นตำแหน่งที่ $\nabla p(x_i) = 0$ จะเห็นได้ว่า S คือตำแหน่งหยุดของตัวแทนทุกตัวนั่นเอง ทุกจุดที่มี s_i เป็นตัวแทนก็จะถือว่าถูกลำเลียงมาหยุดที่ตำแหน่ง s_i นี้ด้วย ดังนั้นในการจัดกลุ่ม S จะถูกนำไปพิจารณาว่าตัวแทนที่หยุดแต่ละตัวนั้นเข้าสู่ไปสู่ความหนาแน่นสูงสุดเดียวกันหรือไม่อีกด้วย

ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย

1. ย้ายตำแหน่งตามขั้นตอนวิธีการย้ายค่าเฉลี่ยแบบเดิม
2. หาค่าตำแหน่งที่ใกล้ที่สุดระหว่างตำแหน่งที่ย้ายมาที่ตำแหน่งเริ่มต้นตามกฎเบื้องต้น
3. หามatricซ์ตำแหน่งของตัวแทน (h) ซึ่งเป็นไปตามกฎแห่งความต่อเนื่อง
4. ในจำนวนของตัวแทนนี้ หาตัวแทนที่หยุดการคำนวณตามเงื่อนไขในสมการที่ (7)
5. คงข้อมูลที่เป็นตัวแทนซึ่งยังไม่หยุดการคำนวณ เพื่อใช้ในการคำนวณในรอบต่อไป
6. ทำซ้ำขั้นตอน 1-5 จนกระทั่งไม่เหลือข้อมูลในการคำนวณ

4. วิธีการทดลองและผลการทดลอง

การทดลองนี้ทำขึ้นเพื่อเปรียบเทียบการทำงานระหว่างขั้นตอนวิธีการย้ายค่าเฉลี่ยแบบเดิม กับขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยที่ได้นำเสนอขึ้นใหม่ โดยทั่วไปแล้วผลการทำงานของขั้นตอนวิธีการย้ายค่าเฉลี่ยแบบเดิม จะขึ้นอยู่กับ ค่า σ และการกำหนดค่า threshold บทความนี้จึงได้ทำการทดลองเปรียบเทียบการเปลี่ยน σ บางค่าเพื่อให้เห็นความแตกต่าง โดยแต่ละการทดลองใช้ตัวแปรเดียวกัน ทดลองบนคอมพิวเตอร์และสิ่งแวดล้อมเดียวกัน



รูปที่ 2 แสดงผลการจัดกลุ่มบางขั้นตอนของข้อมูลทดสอบโดยใช้ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย

4.1 การจัดกลุ่มข้อมูล

เพื่อให้เห็นการทำงานของขั้นตอนวิธีได้อย่างชัดเจน ผู้วิจัยได้ทำการสุ่มข้อมูล 2 มิติ จำนวน 1200 ตำแหน่ง แบ่งเป็น 2 กลุ่ม เป็นแบบการกระจายตัวปกติแล้ว ดำเนินการตาม ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยที่ได้ นำเสนอ ไปแล้วข้างต้น ได้ผลดังรูปที่ 2

จากรูปที่ 2 แต่ละแถวจะแสดงภาพผลลัพธ์ในแต่ละรอบของการคำนวณซึ่งมีทั้งหมด 3 รอบ แต่ละคอลัมน์ แสดงผลลัพธ์บางช่วงของขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยดังนี้ คอลัมน์ทางซ้ายมือเป็นภาพของจุดที่ย้ายแล้ว กับจุดที่อยู่ใกล้ที่สุดตามขั้นตอนที่ 1 และ 2 ของขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแสดงโดยเส้นเชื่อม คอลัมน์กลาง แสดงเส้นเชื่อมระหว่างจุดที่ย้ายแล้วกับตัวแทน ตามขั้นตอนที่ 3 คอลัมน์ทางขวามือ แสดงจุดที่ยังเหลืออยู่ หลังจากขั้นตอนที่ 5 ของขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย

ซึ่งจำนวนจุดในรอบที่ 1, 2 และ 3 จะเหลืออยู่ 11, 4 และ 0 ตามลำดับจะเห็นว่าขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย สามารถลดจำนวนจุดที่ใช้คำนวณลงได้เป็นจำนวนมาก

เพื่อเปรียบเทียบความเร็วในการจัดกลุ่มข้อมูลกับ ขั้นตอนวิธีการย้ายค่าเฉลี่ย จึงได้ทำการทดลองจัดกลุ่ม ข้อมูล 2 ชุดคือข้อมูลที่จำลองขึ้นในขั้นตอนแรกเป็นข้อมูล 2 มิติ จำนวน 1200 ตำแหน่งแบ่งเป็น 2 กลุ่มแบบการกระจายตัวปกติ และข้อมูลจริงที่นิยมใช้ คือ ข้อมูลของดอก iris ขนาด 4 มิติจำนวน 150 ข้อมูล ได้ผลดังตารางที่ 1

โดยในการทดลองเปรียบเทียบความเร็วนี้ เวลาที่นำมา เปรียบเทียบของขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย ได้รวม ขั้นตอนวิธีการที่นำเสนอตั้งแต่ข้อ 1 ถึง 6 เข้าไปทั้งหมด เพื่อเปรียบเทียบกับขั้นตอนวิธีการย้ายค่าเฉลี่ยแบบเดิม

จะเห็นได้ว่า จากผลการทดลองที่ทำการทดลองจำนวน 10 การทดลองแล้วทำการหาค่าเฉลี่ย ในตารางที่ 1 จำนวน

รอบที่ใช้ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยได้ลดลงอย่างมาก เมื่อเปรียบเทียบเวลาจะเห็นได้ว่าจากการทดลองนี้สามารถเพิ่มความเร็วได้ถึง 7 – 40 เท่า เมื่อเทียบกับขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยแบบเดิม ทั้งนี้ขึ้นอยู่กับกำหนดค่า σ และจำนวนข้อมูล หากข้อมูลมีจำนวนมากจะเพิ่มความเร็วได้หลายเท่ากว่าเนื่องจากขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยแบบเดิมในแต่ละรอบจำนวนข้อมูลจะเท่าเดิม ในขณะที่วิธีที่นำเสนอจะใช้น้ำหนักตัวแทนเท่าที่จำเป็นเท่านั้นซึ่งน้อยมากดังแสดงในคอลัมน์ข้อมูลที่เหลือแต่ละรอบของตารางที่ 1 โดยตัวแทนเหล่านี้จะหยุด ณ ตำแหน่งเดียวกับขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยแบบเดิม

ตารางที่ 1 เปรียบเทียบความเร็วในการจัดกลุ่มข้อมูลของขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยกับ ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยเดิม

ข้อมูล	σ	จำนวนรอบ		ข้อมูลเหลือแค่รอบ	เวลาเฉลี่ยใน 10 รอบ(วินาที)		
		MS	TMS		MS	TMS	Speed up
1	0.5	16	4	23,7,4,0	8.5382	0.7203	11.85
	0.1	85	12	350,94,39,22,16,7,3,3,3,1,1,0	40.2036	0.9878	40.70
2	0.5	16	4	5,2,2,0	0.1100	0.0149	7.38
	0.2	34	7	19,5,5,4,3,1,0	0.2229	0.0184	12.11

4.2 การแยกส่วนประกอบของภาพ

ผู้ทดลองได้ทำการทดลองแยกส่วนประกอบของภาพโดยใช้ภาพที่นิยมใช้ในการเปรียบเทียบจำนวน 4 ภาพ เพื่อเปรียบเทียบจำนวนรอบ เวลา และแสดงจำนวนข้อมูลที่เหลืออยู่ในแต่ละรอบของขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยนี้ รวมทั้งวัดจำนวนเท่าของการเพิ่มความเร็วรายละเอียดและคุณสมบัติของภาพแสดงในตารางที่ 2 และผลการทดลองที่ทำการทดลองจำนวน 10 การทดลองแล้วทำการหาค่าเฉลี่ย แสดงในตารางที่ 3

จากผลการทดลองที่แสดงในตารางที่ 3 จะเห็นว่าวิธีที่นำเสนอใหม่นี้สามารถเพิ่มความเร็วในการแยกส่วนประกอบภาพได้สูงถึง 13 – 70 เท่า และจำนวนข้อมูล

ที่เหลือในแต่ละรอบ ก็ลดลงอย่างมาก เช่นในภาพ Smallbowelloop ซึ่งในภาพนี้การคำนวณในรอบที่ 2 จะเหลือข้อมูลเพียง 24 pixel จากทั้งหมด 3250 pixel ตัวอย่างผลการแยกส่วนประกอบของภาพทั้ง 4 ภาพโดยใช้ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยแสดงดังรูปที่ 2

ตารางที่ 2 แสดงการใช้ σ จำนวนรอบ เวลา เปรียบเทียบระหว่าง ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยกับ ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยเดิม

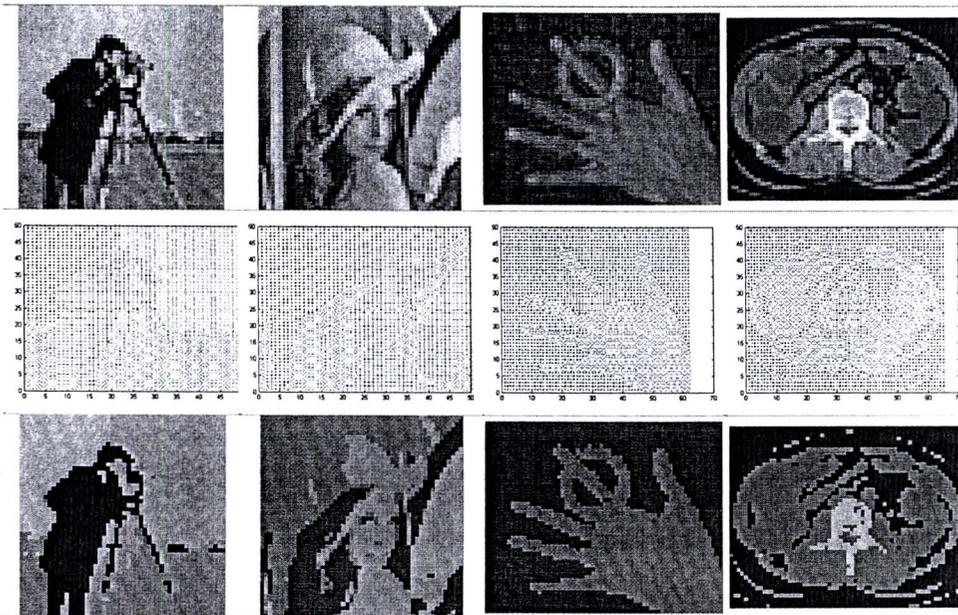
ข้อมูล	คุณสมบัติ	ข้อมูลที่น่าสนใจ
1 Cameraman	ระดับสีเทา50x50	(x,y,I)
2.Lena	ภาพสี 50x50	(x,y,R,G,B)
3. Hand	ภาพสี 62x50	(x,y,R,G,B)
4. Smallbowelloop	ระดับสีเทา65x50	(I)

ตารางที่ 3 เปรียบเทียบความเร็วในการจัดกลุ่มข้อมูลเมื่อเทียบกับ ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ย

ข้อมูล	σ	จำนวนรอบ		ข้อมูลเหลือแต่ละรอบ	เวลาเฉลี่ยใน 10 รอบ(วินาที)		
		MS	TMS		MS	TMS	Speed up
1	400	120	10	71, 23, 10, 2, 1, 1, 1, 1, 1, 0	266.85	3.80	70.22
	200	51	7	135, 36, 15, 9, 5, 2, 0	115.83	3.85	30.09
2	500	66	6	211, 43, 18, 7, 3, 0	198.13	5.37	36.90
	200	81	9	487, 156, 60, 23, 10, 5, 2, 1, 0	241.06	5.86	41.14
3	500	21	8	137, 28, 13, 4, 2, 1, 1, 0	106.69	8.20	13.01
	200	78	10	514, 202, 94, 39, 21, 12, 4, 2, 1, 0	368.72	9.24	39.90
4	100	73	3	62, 3, 0	191.96	6.24	30.76
	90	83	3	24, 3, 0	201.03	6.24	32.22

5. สรุปผลการทดลอง

ขั้นตอนวิธีการไล่เรียงค่าเฉลี่ยได้ถูกนำมาพิจารณาใหม่เพื่อช่วยลดความซ้ำซ้อนในการคำนวณโดยการหาตัวแทนของตำแหน่งที่ถูเข้าไปยังจุดที่มีความหนาแน่นสูงสุดเดียวกัน โดยใช้แนวความคิดของจุดใกล้ที่สุด และความสัมพันธ์ของการเชื่อมถึงกัน ได้ระหว่างแต่ละจุด เพื่อลดจุดเหล่านั้น



รูปที่ 2 แสดงผลการแยกส่วนประกอบของภาพ แถวแรกเป็นภาพต้นฉบับ แถวกลางแสดงพิกเซลที่อยู่ในกลุ่มเดียวกัน แถวล่างแสดงผลการแยกส่วนประกอบของภาพในตารางที่ 2

แล้วใช้เฉพาะตัวแทนในการลำเลียงเพื่อคำนวณในรอบต่อไป โดยที่ตัวแทนยังคงใช้ขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิม ส่งผลให้ตำแหน่งตัวแทนในแต่ละรอบ เป็นตำแหน่งเดียวกันกับตำแหน่งของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิมนั่นเอง ตามขั้นตอนวิธีการลำเลียงย้ายตามค่าเฉลี่ยที่นำเสนอขึ้น จะสามารถลดจำนวนข้อมูลที่เหลือในแต่ละรอบลงอย่างชัดเจน โดยสามารถเพิ่มความเร็วได้ถึง 7 – 40 เท่าในการจัดกลุ่มข้อมูลและเพิ่มความเร็วการแยกส่วนประกอบภาพได้สูงถึง 13 – 70 เท่า เมื่อเทียบกับขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิม

6. อ้างอิง

- [1] K. Fukunaga, L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp. 32 – 40, 1975.
- [2] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, August 1995.
- [3] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in Proc. Int'l Conf. Computer Vision, 1999, pp. 1197–1203.
- [4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603 – 619, May 2002.
- [5] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," in Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, 2002.
- [6] M. Á. Carreira-Perpiñán, "Fast nonparametric clustering with Gaussian blurring mean-shift," Proc. of the 23rd Int. Conf. on Machine learning table of contents, Pennsylvania, 2006.
- [7] P. Padungweing, S. Chiewchanwattana, and K. Sunat, "Resilient Mean Shift Algorithm," 13th Int. Conf. on Systems, Signals and Image Processing, Sep 21-23, Hungary, 2006.
- [8] K. Zhang, James T. Kwok, and M. Tang. "Accelerated Convergence Using Dynamic Mean Shift." In the 9th European Conference on Computer Vision (ECCV 2006), Graz, Austria, May 2006.
- [9] M. Á. Carreira-Perpiñán, "Acceleration Strategies for Gaussian Mean-Shift Image Segmentation," Proc. of the 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2006.
- [10] M. Á. Carreira-Perpiñán, "Mode finding for mixtures of Gaussian distributions," *IEEE Trans. PAMI*, vol. 22, no. 11, pp. 1318-1323, Nov. 2000.
- [11] R.E. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146-160, 1972.

ภาคผนวก ข
บทความที่ผ่านการนำเสนอแบบปากเปล่า และตีพิมพ์ในวารสาร
รายงานการประชุมวิชาการ
ในงาน NCSEC 2009 จัดขึ้นที่ โรงแรมมณเฑียรริเวอร์ไซด์ไฮเทล
กรุงเทพมหานคร

การแยกส่วนประกอบภาพโดยใช้ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแบบครอบคลุม Image Segmentation via Generalized Transport Mean Shift Algorithm

พนิดา ผดุงเวียง¹, ลีรภัทร เขียวชาอุวัฒน์² และ คาร์ณ สุนติ³

^{1,2} ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น E-mail: ¹panida2m@gmail.com, ²sunkra@kku.ac.th and

³ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร E-mail: khamron_sunat@yahoo.com

บทคัดย่อ

การแยกส่วนประกอบของภาพเป็นกระบวนการที่สำคัญในการเรียนรู้ของเครื่อง การรู้จำและการดำเนินการกับภาพเคลื่อนไหว ขั้นตอนวิธีการย้ายค่าเฉลี่ยเป็นขั้นตอนหนึ่งที่นิยมนำมาใช้ในการแก้ปัญหานี้ โดยทุก ๆ จุดจะถูกย้ายไปยังตำแหน่งที่มีความหนาแน่นสูงสุดของตัวมัน ซึ่งมีหลายจุดที่ถูกย้ายไปยังเส้นทางเดียวกัน การลดจำนวนจุดเหล่านี้จะช่วยให้การคำนวณเร็วขึ้น บทความนี้ได้นำเสนอการลดจำนวนข้อมูลในแต่ละรอบโดยให้ข้อมูลที่มีจุดหมายเดียวกันถูกลำเลียงไปโดยใช้ตัวขนส่ง ซึ่งทำให้การคำนวณลดลงอย่างมาก โดยตัวขนส่งนี้ยังคงดำเนินการตามขั้นตอนวิธีการย้ายค่าเฉลี่ย ทำให้ผลลัพธ์ยังคงเหมือนกับการใช้ขั้นตอนวิธีการย้ายค่าเฉลี่ยเดิม ขั้นตอนวิธีที่นำเสนอขึ้นนี้ เป็นขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแบบครอบคลุมที่สามารถให้ผลเดียวกับขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย และขั้นตอนวิธีการย้ายค่าเฉลี่ยแบบเดิม ผลจากการทดลองการแยกส่วนประกอบของภาพพบว่า สามารถเพิ่มความเร็วได้ถึง 5-333 เท่าเมื่อเทียบกับขั้นตอนการย้ายค่าเฉลี่ยเดิมโดยมีความคลาดเคลื่อนน้อย

คำสำคัญ: ขั้นตอนวิธีการย้ายค่าเฉลี่ย, ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแบบครอบคลุม, การแยกส่วนประกอบของภาพ

Abstract

Image segmentation is an important process in machine learning, pattern recognition and video processing. Mean Shift algorithm is the one among the powerful techniques that can automatically find the mode of density then the segmentation easy be performed. Every position are shifted in to the position near their mode through their trajectory. However, there are a lot of positions which are shifted through the same trajectory and shifted into the same mode. Thus, reducing those positions can reduce the computational time. In this paper, the Generalized Transport Mean Shift algorithm (GTMS) is proposed to remedy the problem. The idea is the data points that tend to

be shifted through the same trajectory will be transported by the same transporter, and be excluded from the computation in the next iteration. The transporters are, then, computed by using the standard mean shift algorithm. The experimental result on image segmentation using well known images showed that, our algorithm achieves 5 – 333 times faster than the standard mean shift algorithm with low error.

Keywords: Mean Shift algorithm, Generalized Transport Mean Shift algorithm, image segmentation

1. บทนำ

ขั้นตอนวิธีการย้ายค่าเฉลี่ย (Mean-Shift algorithm) เป็นวิธีหาจุดที่มีความหนาแน่นของข้อมูลที่มากที่สุดโดยเริ่มจากการคำนวณค่าความหนาแน่นของทุกจุดเทียบกับจุดพิจารณา แล้วคำนวณค่าเฉลี่ยเพื่อหาตำแหน่งการย้ายใหม่ การเปลี่ยนแปลงนี้จะเป็นลักษณะการเดินทางตามเส้นทางไปยังตำแหน่งที่มีความหนาแน่นของข้อมูลมากขึ้นจนกระทั่งตำแหน่งการย้ายที่เกิดจากการคำนวณไม่มีการเปลี่ยนแปลงหรือเปลี่ยนแปลงน้อยกว่าค่าที่ยอมรับได้ก็จะหยุดคำนวณ และจุดที่เข้าสู่ที่จุดเดียวกันจะถือว่าอยู่ในกลุ่มเดียวกัน ขั้นตอนวิธีนี้เดิมได้ถูกนำเสนอในชื่อว่า "valley-seeking procedure" [1] ต่อมาได้มีการศึกษาขั้นตอนวิธีเพื่อนำมาใช้วิเคราะห์ภาพ [2] สามารถช่วยในการแยกองค์ประกอบของภาพ (image segmentation) [3, 4] รวมทั้งยังสามารถนำมาใช้ในการติดตามภาพเคลื่อนไหว (tracking) [5] ปัจจุบันมีการนำขั้นตอนวิธีการย้ายค่าเฉลี่ยมาประยุกต์ใช้ในหลายด้านไม่ว่าจะเป็นการแยกองค์ประกอบของภาพ (image segmentation) การค้นหาข้อมูลภาพเคลื่อนไหว (tracking) การวิเคราะห์ และจัดกลุ่มข้อมูล (Analysis and Clustering) มีการนำขั้นตอนวิธีนี้มาใช้ในการแยกส่วนประกอบของภาพอย่างกว้างขวางเนื่องจากข้อมูลภาพมีช่วงของข้อมูลที่จำกัดแน่นอน ทำให้การกำหนดความกว้าง (bandwidth) ของฟังก์ชันความหนาแน่นเป็นไปได้โดยง่าย อย่างไรก็ตามขั้นตอนนี้มีการทำงานเป็น $O(kn^2m)$ [6] ซึ่งยังไม่เหมาะสมที่

จะนำมาใช้กับการใช้งานจริง โดยเฉพาะกับการแยกองค์ประกอบของภาพที่ปัจจุบันมีจำนวนจุดของข้อมูลสูงมาก การเพิ่มความเร็วของขั้นตอนวิธีการย้ายตามค่าเฉลี่ยจึงมีความสำคัญ

มีการนำเสนอการเพิ่มความเร็วด้วยเทคนิคของเครือข่ายประสาทเทียม [7] ทำให้ขั้นตอนนี้ใช้จำนวนรอบน้อยลงและยังมีความถูกต้องเมื่อเทียบกับวิธีการเดิม จากนั้นมีการนำเสนอการเพิ่มความเร็วแบบพิจารณาตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้า [6] โดยมีการจัดกลุ่มของข้อมูลในแต่ละรอบและตัดข้อมูลกลุ่มที่มีสมาชิกบางตัวย้ายไปตำแหน่งสูงสุดแล้ว ทำให้ขั้นตอนต่อไปมีจำนวนข้อมูลน้อยลง การคำนวณจึงเร็วขึ้นตามไปด้วย และมีการนำเสนอการลดจำนวนข้อมูลด้วยการจัดกลุ่มของข้อมูลเพื่อหาตัวแทนก่อนทำขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบพิจารณาตามข้อมูลที่ย้ายไปแล้วในรอบก่อนหน้า [8] วิธีนี้ให้ผลเร็วแต่มีความคลาดเคลื่อนสูงทั้งจำนวนกลุ่มข้อมูลและสมาชิกในแต่ละกลุ่ม

ต่อมามีการนำเสนอการเพิ่มความเร็วสำหรับการแยกส่วนประกอบของภาพ [9] วิธีวิธีคือ หนึ่งรวมตำแหน่งของภาพที่อยู่ใกล้กันเป็นเซลล์เดียวกัน จากนั้นใช้วิธีการย้ายตามค่าเฉลี่ย และเมื่อเซลล์ย้ายไปยังเซลล์ที่มีการย้ายตำแหน่งไปก่อนหน้าแล้ว ก็จะหยุดคำนวณ และถือว่าเป็นกลุ่มเดียวกัน วิธีที่สองหาตัวแทนของจุดที่อยู่ใกล้กันในโดเมนของภาพ กำหนดโดยระยะห่างคงที่ค่าหนึ่ง แล้วดำเนินการคล้ายกับวิธีแรก ทั้งสองวิธีได้ผลเร็วมาก แต่ความคลาดเคลื่อนจะเกิดขึ้นตั้งแต่ตอนแรกที่มีการรวมกลุ่มของจุดบนภาพ อีกสองวิธีเป็นการลดการคำนวณของฟังก์ชันหาความหนาแน่นและใช้วิธีรู้เข้าแบบกำลังสี่ทำให้จำนวนรอบในการย้ายน้อยกว่า แต่ใช้การคำนวณที่สูงมากจึงทำให้ช่วยเพิ่มความเร็วไม่มากเท่าที่ควร หลังจากนั้นได้มีการนำเสนอการใช้ผู้ขนส่ง และผู้พ่วงเพื่อลดการย้ายตำแหน่งที่ซ้ำซ้อน [10] ซึ่งเป็นวิธีได้ผลเร็วแต่ยังมีความคลาดเคลื่อนอยู่บ้าง

ในบทความนี้มีจุดประสงค์ที่การนำเสนอขั้นตอนวิธีการไล่เลียงย้ายค่าเฉลี่ยแบบครอบคลุมมาใช้ในการแยกส่วนประกอบของภาพ โดยจะนำการพิจารณาของเวกเตอร์มาใช้กับส่วนของการเลือกผู้ขนส่ง และผู้พ่วงเพื่อลดความคลาดเคลื่อนของการย้ายไปยังผู้ขนส่งต่างกลุ่ม

2. ทฤษฎีที่เกี่ยวข้อง

2.1 ขั้นตอนวิธีการย้ายตามค่าเฉลี่ย (Mean-Shift algorithm)

กำหนดให้ข้อมูลนำเข้าเป็น $X \in R^n$ โดยที่ $X = (x_1, x_2, \dots, x_n)$ และ $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,m}]^T$ การประมาณความหนาแน่นของข้อมูล ณ ตำแหน่ง x ใด ๆ สามารถคำนวณได้ดังสมการ

$$p(x) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{\|x - x_i\|^2}{\sigma}\right) \tag{1}$$

โดยที่ σ เป็นค่าคงที่และ $K(t)$ เป็นฟังก์ชันเคอร์เนล ตำแหน่งที่มีความหนาแน่นสูงสุดของบริเวณที่พิจารณานี้คือตำแหน่งที่ $\nabla p(x) = 0$ ซึ่งสามารถหาได้โดยการคำนวณเพื่อย้ายตำแหน่งหลาย ๆ รอบโดยการย้ายในแต่ละรอบคำนวณได้ดังสมการ

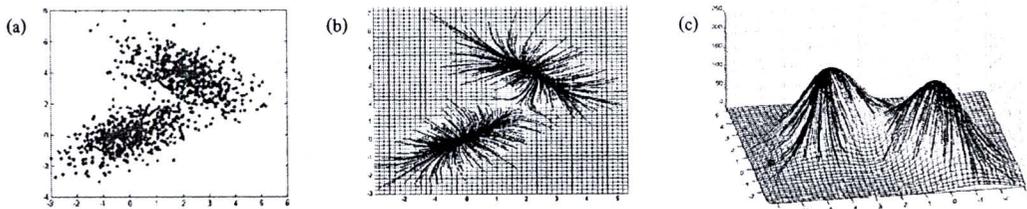
$$x^{(r+1)} = f(x^{(r)}) \tag{2}$$

โดยที่ $f(x) = \frac{\sum_{i=1}^n K'\left(\frac{\|x - x_i\|^2}{\sigma}\right) x_i}{\sum_{j=1}^n K'\left(\frac{\|x - x_j\|^2}{\sigma}\right)}$ \tag{3}

เมื่อ $K'(t) = \frac{\partial K(t)}{\partial t}$ ขั้นตอนวิธีนี้เรียกว่า ขั้นตอนวิธีการย้ายตามค่าเฉลี่ย จาก [11] ถ้าฟังก์ชันเคอร์เนลเป็นฟังก์ชันเกาส์เซียน ($K(t) = e^{-t^2}$) สมการที่ (2) และ (3) จะเป็นดังสมการ

$$x^{(r+1)} = \sum_{i=1}^n p(i | x^{(r)}) x_i \tag{4}$$

$$p(i | x^{(r)}) = \frac{\exp\left(-\frac{1}{2} \frac{\|x^{(r)} - x_i\|^2}{\sigma}\right)}{\sum_{j=1}^n \exp\left(-\frac{1}{2} \frac{\|x^{(r)} - x_j\|^2}{\sigma}\right)} \tag{5}$$



รูปที่ 2 (a) ข้อมูลทดลองในสองมิติ (b) เส้นทางการเคลื่อนที่ของข้อมูลไปยังจุดสูงสุดของความหนาแน่นในสองมิติ และ (c) ในสามมิติ

เมื่อสิ้นสุดขั้นตอนวิธีการย้ายตามค่าเฉลี่ย ทุกตำแหน่งจะถูกย้ายไปที่ตำแหน่งที่มีความหนาแน่นสูงสุดบริเวณของตำแหน่งนั้น หลังจากนั้นก็จะสามารถดำเนินการจัดกลุ่มข้อมูลโดยใช้ขั้นตอนวิธีในการจัดกลุ่มต่อได้โดยง่าย โดยในบทความนี้จัดกลุ่มโดยการวัดระยะทางระหว่างจุดเหล่านี้ หากมีค่าน้อยกว่า $G/2$ จะถือว่าอยู่ในกลุ่มเดียวกัน

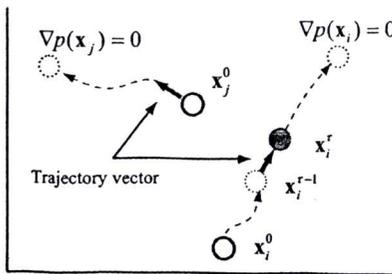
2.2 ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย (Transport Mean Shift Algorithm)

ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ย [10] เป็นขั้นตอนวิธีในการลดการคำนวณในรอบถัดไป โดยหาตำแหน่งที่เป็นผู้ขนส่ง ในการลำเลียงผู้ห้วง ซึ่งผู้ขนส่งนี้จะดำเนินการตามขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม ทำให้ตำแหน่งสุดท้ายเป็นตำแหน่งเดียวกันกับขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม ในแต่ละรอบจะมีการใช้กราฟเก็บเส้นเชื่อมของผู้ขนส่งและผู้ห้วง ในแต่ละรอบจะพิจารณาตำแหน่งที่ $\nabla p(x) = 0$ โดยใช้เงื่อนไขการหยุดเช่นเดียวกับ ขั้นตอนวิธีการย้ายตามค่าเฉลี่ยเดิมตามเงื่อนไข

$$\|x^f - x^{f-1}\|^2 \leq \text{threshold} \quad (6)$$

ในบทความนี้ใช้ $\text{threshold} = 10^{-7}$ สำหรับทุกขั้นตอนวิธีที่นำมาเปรียบเทียบกัน ในแต่ละรอบตำแหน่งที่หยุดแล้วจะไม่ถูกนำมาคำนวณหรือพิจารณาอีกในรอบต่อไป และเมื่อการทำงานเสร็จสิ้น จะใช้ส่วนที่เชื่อมกันของกราฟ (Graph connected component) [12] เพื่อกำหนดให้ตำแหน่งสุดท้ายของผู้ห้วงเป็นตำแหน่งเดียวกับผู้ขนส่ง

3. ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแบบครอบคลุม (Generalized Transport Mean Shift Algorithm)



รูปที่ 1 การย้ายตำแหน่งของ x_i ในรอบที่ T

ในขั้นตอนวิธีนี้ได้เพิ่มการพิจารณาทิศทางเพื่อเลือกตัวขนส่งหรือผู้ขนส่ง จากรูปที่ 1 จะเห็นว่า การย้ายของ ตำแหน่ง i ในรอบที่ T นี้ จะย้าย

มาใกล้ตำแหน่ง j ซึ่งหากเราให้ j เป็นผู้ขนส่งสุดท้ายแล้ว i จะถูกระบุให้มีตำแหน่งเดียวกับ j ซึ่งไม่ใช่จุดยอดที่ตำแหน่งสุดท้ายของ i ควรจะอยู่ หากพิจารณาทิศทางของเส้นทางของตำแหน่งที่ i ในรอบปัจจุบันเทียบกับในรอบก่อนหน้า จะเห็นว่ามีความแตกต่างกับทิศทางของเส้นทางของจุด j ในตอนเริ่มต้น ดังนั้นก่อนที่จะระบุให้ j เป็นตัวแทนจึงควรมีการตรวจสอบทิศทางดังที่กล่าวนี้ ในการตรวจสอบที่นำเสนอในบทความนี้จะใช้ข้อมูลที่จะเป็นจะต้องคำนวณอยู่แล้วในขั้นตอนวิธีการย้ายค่าเฉลี่ยเดิม ทำให้ไม่เสียเวลาในการคำนวณใหม่ และในแต่ละรอบจะไม่ใช้กราฟในการเชื่อมแต่ละจุด เพื่อลดการใช้พื้นที่เก็บข้อมูลระหว่างการทำงาน

เพื่อแก้ปัญหาดังที่กล่าวมาแล้ว บทความนี้ได้นำเสนอ การใช้เมตริกซ์ 4 เมตริกซ์ โดยเมตริกซ์แรกเก็บผู้ขนส่งของผู้ขนส่งตัวที่ i เมตริกซ์ที่สองเก็บเวกเตอร์ทิศทางเดินทางของผู้ขนส่งตัวที่ i เมตริกซ์ที่ 3 เก็บข้อมูลที่แสดงถึงการตัดสินใจแบบตรรกะของผู้ขนส่งตัวที่ i ที่ระบุว่าข้อมูลเข้าสู่ความหนาแน่นสูงสุดแล้วหรือไม่ เมตริกซ์สุดท้ายเก็บข้อมูลที่แสดงถึงการตัดสินใจแบบตรรกะของผู้ขนส่งตัวที่ i ที่ระบุว่า ในรอบต่อไปจะต้องคำนวณ i หรือไม่ รายละเอียดของแต่ละเมตริกซ์เป็นดังนี้

ให้ $T \in I^m$ คือเมตริกซ์ผู้ขนส่ง โดยที่ $t_i \in T$ หมายถึง ผู้ขนส่งของผู้ขนส่งตัวที่ i ในรอบปัจจุบัน

$$t_i = \arg \min_j (\|x_i^f - x_j^f\|^2) \quad (7)$$

ให้ $V \in R^{m \times m}$ เป็นเมตริกซ์ระบุทิศทางของเมตริกซ์ผู้ขนส่งหาจาก

$$v_i = \frac{x_i^f - x_i^0}{\|x_i^f - x_i^0\|} \quad (8)$$

ซึ่งเป็นเวกเตอร์หนึ่งหน่วยที่บอกทิศทางของผู้ขนส่งตัวที่ i และสองเมตริกซ์สุดท้ายคือ $C_{in}, c_i \in C$ และ $A_{in}, a_i \in A$ ระบุไว้ดังนี้

$$c_i = \begin{cases} 1 & \text{if the } i^* \text{ data convert} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$a_i = \begin{cases} 1 & \text{if the } i^* \text{ data should be present} \\ & \text{to the next iteration} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

ขั้นตอนวิธีการลำเลียงย้ายค่าเฉลี่ยแบบครอบคลุม (GTMS) แสดงได้ดัง Algorithm 1 ในรูปแรกนั้นเป็นขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม ยกเว้นการคำนวณเวกเตอร์ทิศทางในขั้นตอนที่ 2 ขณะที่ขั้นที่ 6, 9 และ 10 เป็นขั้นตอนที่ต้องดำเนินการอยู่แล้วในขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม อย่างไรก็ตามถ้าผู้ขนส่งถูกพบในขั้นที่ 8 ก็จะไม่ต้องคำนวณใน

ขั้นตอนที่ 9 ซึ่งเป็นการคำนวณฟังก์ชันยกกำลังของทุก ๆ ระยะห่างในขั้นตอนที่ 6 ตามสมการ (5) หากใช้ฟังก์ชันเคอร์เนลเป็นฟังก์ชันเกาส์เซียน และไม่จำเป็นต้องคำนวณ x , ในรอบที่ τ ซึ่งเป็นการหาค่าเฉลี่ยโดยกำหนดน้ำหนักของทุก ๆ จุดใน X ตามสมการ (4) ดังนั้นจะเห็นว่าถ้ามีการพบผู้ขนส่งจำนวนมากก็จะเหลือจุดน้อยสำหรับการคำนวณระยะห่างต่อในขั้นตอนที่ 6 ของรอบต่อไป รวมทั้งการลดการคำนวณในขั้นตอนที่ 9 ดังที่กล่าวมาแล้ว ดังนั้น GTMS ควรจะเรียกว่าขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิมเป็นอย่างมากซึ่งจะแสดงในผลการทดลองการแยกส่วนประกอบของภาพ

ตัวแปร α ทำให้ขั้นตอนวิธีมีความยืดหยุ่นมากขึ้น สามารถปรับเพื่อให้ทำงานเหมือนกับ MS รวมทั้ง TMS ได้ ซึ่งง่ายต่อการเลือกค่าที่เหมาะสมกับงานซอฟต์แวร์ที่สร้างขึ้นมาเพื่อใช้กับงานเฉพาะด้าน α จะถูกใช้ในการตัดสินใจเลือกผู้ขนส่งซึ่งมีทิศทางขนานกับทิศทางของการย้ายของข้อมูลที่พิจารณาในรอบปัจจุบัน ถ้า $\alpha = 0$ หมายความว่า ผู้ขนส่งจะถูกกำหนดให้กับผู้พ่วง ก็ต่อเมื่อมีทิศทางการย้ายขนานกัน ถ้า $\alpha < 0$ หมายความว่า GTMS จะให้ผลเหมือนกับ MS ทุกประการ เพราะว่าจะไม่มีการระบุผู้ขนส่งให้ผู้พ่วง เงื่อนไขในการหยุดมีเพียงขั้นตอนตาม MS ในขั้นตอนที่ 10 และ GTMS จะให้ผลเดียวกับ TMS ได้ โดยการกำหนดค่า $\alpha > 1$ ในกรณีนี้ตัวที่อยู่ใกล้ข้อมูลที่ย้ายมาในรอบปัจจุบันจะถูกระบุเป็นผู้ขนส่งเสมอ และข้อมูลที่เป็นผู้พ่วงนั้นก็เลยไม่ถูกคำนวณในรอบต่อไป ดังนั้น การคำนวณจะเรียกว่า MS โดยผลลัพธ์มีความคลาดเคลื่อนเล็กน้อยดังที่กล่าวมาแล้วข้างต้น

Algorithm1 (GTMS)

```

Initialization: C = false, A = true
for each  $x_i \in X$  do
1. Run one iteration of MS from  $x_i$  to construct  $x_i^*$ 
2. Calculate the shift distance and the trajectory vector
    $z = x_i^* - x_i$ 
    $s = \|z\|^{0.5}$ 
    $v_i = z/s$  //  $v_i$  is a unit trajectory vector
3. Considering the convergence
   if  $s \leq \text{threshold}$  then
      $c_i = \text{true}$ ,  $a_i = \text{false}$ 
   end if
4. Set  $i^{\text{th}}$  to be a transporter itself,  $t_i = i$ 
end for
5.  $\tau = 1$ ,  $U = V$ 
while these are  $a_i = \text{true}$  do
6. Calculate the distance ( $d_k$ ) from  $x_i^*$  to  $x_k \in X$ ,  $1 \leq k \leq n$ 
7. Find the  $j^{\text{th}}$  data that nearest to the  $i^{\text{th}}$  data
    $j = \text{argmin}_k d_k$ 
8. Investigate the transporter
   if  $i \neq j$  &  $t_i \neq i$  &  $\frac{1}{2}(1 - v_i^T \cdot u_j + |v_i^T \cdot u_j|) < \alpha$  then
      $t_i = j$ ,  $a_i = \text{false}$  // assign transporter
     and inactive the trailer
   else
     Compute  $x_i^{\tau+1}$ 
10. Follow step 2 and 3
   end if
    $\tau = \tau + 1$ 
end while

```

อย่างไรก็ตามถ้าเราพิจารณาธรรมชาติของ ขั้นตอนวิธีการย้ายตามค่าเฉลี่ยแบบเดิม α นี้ควรมีค่าเพิ่มขึ้นเมื่อจำนวนรอบเพิ่มขึ้น เพราะว่าเมื่อจำนวนรอบเพิ่มขึ้น ข้อมูลจะถูกย้ายไปยังตำแหน่งที่ใกล้กับจุดยอดซึ่งจะอยู่ไกลกว่าข้อมูลอื่นที่ไม่ได้อยู่ภายใต้จุดยอดเดียวกันดังแสดงในรูปที่ 2 ในรอบที่เพิ่มขึ้นเหล่านี้ผู้ขนส่งจะมีโอกาสสูงที่จะอยู่ภายใต้จุดยอดเดียวกันกับข้อมูลที่เหลือ ดังนั้นควรที่จะผ่อนคลายเงื่อนไขของการขนานกัน ในบทความนี้ได้เสนอวิธีการผ่อนคลายเงื่อนไขของการขนานกันของคุณสมบัติ โดยกำหนดให้ α_0 เป็นค่าความขนานเริ่มต้น ในรอบแรก α_p เป็นค่าความขนานเมื่อขั้นตอนวิธีมาถึงขั้นตอนที่ τ_p ค่า α สามารถกำหนดได้ดังนี้

$$\alpha = \min(\alpha_p, \tau(\alpha_0 + \alpha_p/\tau_p)) \quad (11)$$

4. วิธีการทดลองและผลการทดลอง

การทดลองนี้ทำขึ้นเพื่อเปรียบเทียบการทำงานระหว่าง MS กับ TMS และ GTMS โดยทั่วไปแล้วผลการทำงานของ MS จะขึ้นอยู่กับค่า σ และการกำหนดค่า threshold บทความนี้จึงได้ทำการทดลองเปรียบเทียบการเปลี่ยน σ บางค่าเพื่อให้เห็นความแตกต่าง โดยแต่ละการทดลองใช้ตัวแปรเดียวกัน ทดลองบนคอมพิวเตอร์และสิ่งแวดล้อมเดียวกัน

4.1 การแยกส่วนประกอบของภาพ

ตารางที่ 1 คุณสมบัติของภาพที่ใช้ในการทดลอง

ข้อมูลที	1 Hand	2. Hawk	3. Houses	4. Lena	5. Vegetable
คุณสมบัติ	ที 75x50	ที 75x50	ที 75x50	ที 50x50	ที 196x173

ผู้ทดลองได้ทำการทดลองแยกส่วนประกอบของภาพโดยใช้ภาพที่นิยมใช้ในการเปรียบเทียบผลของการแยกส่วนประกอบของภาพคือ Hand และ Lena รวมทั้งภาพ Hawk และ Houses จาก Berkeley Segmentation Dataset and Benchmark และภาพ Vegetable จาก [13] ในการเปรียบเทียบผลการทดลองจำนวน 5 ภาพเนื่องจากขนาดของภาพไม่เท่ากัน บทความนี้ได้ปรับข้อมูลโดยในภาพที่ 1-4 ให้สองมิติแรกซึ่งเป็นตำแหน่งบนภาพในแนวนอนและแนวตั้งให้มีค่าอยู่ในช่วง [1,100] เหมือนกันทุกภาพ ข้อมูลของภาพที่นำไปมาคือ (x, y, R, G, B) ส่วนภาพที่ 5 ใช้ข้อมูลตามบทความ [13] ซึ่งประกอบด้วย (x, y, G-R, G-B, H, S, I) โดย x และ y มีค่าอยู่ในช่วง [1,20] รายละเอียดและคุณสมบัติของภาพแสดงในตารางที่ 1 เพื่อใช้เปรียบเทียบเวลา ค่าความคลาดเคลื่อน และวัดจำนวนเท่าของการเพิ่มความเร็ว แสดงผลการทดลองในตารางที่ 2 รวมทั้งแสดงจำนวนข้อมูลที่เหลืออยู่ใน 5 รอบแรก ในตารางที่ 3

จากผลการทดลองที่แสดงในตารางที่ 2 TMS หมายถึง ขั้นตอนวิธีการลำดับเชิงซ้ายค่าเฉลี่ย GTMS1 หมายถึง ขั้นตอนวิธีการลำดับเชิงซ้ายค่าเฉลี่ยแบบครอบคลุม ที่ให้ผลเดียวกับ TMS ที่กำหนด $\alpha > 1$ และ GTMS2 หมายถึง ขั้นตอนวิธีการลำดับเชิงซ้ายค่าเฉลี่ยแบบครอบคลุมที่กำหนดค่า $\alpha_0=0$, $\alpha_p=0.5$ และ $\tau_p=10$ และ α ในแต่ละรอบคำนวณได้จากสมการ (11) ค่าความถูกต้องวัดได้โดยนับจำนวนข้อมูลที่จัดกลุ่มถูกต้องแต่ละขั้นตอนวิธีเมื่อเทียบกับการจัดกลุ่มของ MS หากด้วยจำนวนข้อมูลทั้งหมด ซึ่ง MS1 และ MS2 ได้ผลลัพธ์เดียวกัน ส่วนจำนวนเท่าของการเพิ่มความเร็วยังเป็นการวัดจำนวนเท่าของเวลาเมื่อเทียบกับ MS เป็นอัตราส่วนระหว่างเวลาของ MS กับแต่ละขั้นตอนวิธี ในภาพที่ 5 มีจำนวนจุดของภาพสูงถึง 33,908 ทำให้ TMS ไม่สามารถประมวลผลได้เนื่องจากต้องใช้หน่วยความจำมาก ขณะที่ MS ใช้เวลาในการประมวลผลสูงมากผู้วิจัยจึงไม่ทำการทดลอง MS กับภาพนี้

จะเห็นว่าวิธีที่นำเสนอใหม่ GTMS1 นี้สามารถเพิ่มความเร็วในการแยกส่วนประกอบภาพได้สูงถึง 130 – 333 เท่า และ สูงถึง 19 – 53 เมื่อเทียบกับ ขั้นตอนวิธีการซ้ายค่าเฉลี่ยแบบไม่ตัดตัวที่หุคออก (MS1) และแบบตัดตัวที่หุคออก (MS2) ตามลำดับ เมื่อไม่พิจารณาทิศทางของผู้ขนส่งและผู้พ่วง โดยมีความคลาดเคลื่อนเล็กน้อย ซึ่งหากต้องการให้ความคลาดเคลื่อนเมื่อเทียบกับขั้นตอนวิธีการซ้ายค่าเฉลี่ยลดลงก็สามารถเพิ่มการพิจารณาทิศทางของผู้ขนส่งและผู้พ่วงเข้าไปด้วย ดังในตารางของ GTMS2 จะสามารถลดความคลาดเคลื่อนอยู่ในช่วง 0.01 ถึง 4.4 เปอร์เซ็นต์ ซึ่งลดลงจาก TMS และ GTMS1 ที่มีความคลาดเคลื่อนอยู่ในช่วง 0.03 ถึง 5.88

ทั้งนี้หากต้องการให้มีความคลาดเคลื่อนน้อยลงก็สามารถลด α_p หรือเพิ่ม τ_p ได้ แต่จะทำให้การเพิ่มความเร็วน้อยลงตามไปด้วยเพราะ

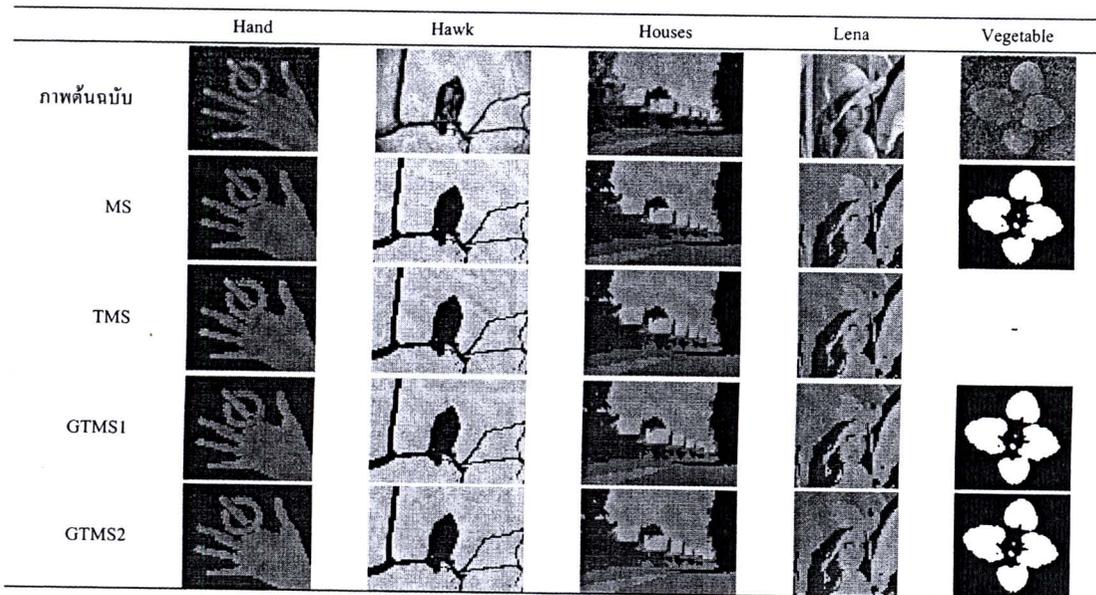
GTMS2 จะมีการคำนวณใกล้เคียงกับ MS เดิมมากขึ้น ทั้งนี้เนื่องจากจะมีจำนวนข้อมูลที่ต้องดำเนินการในแต่ละรอบมากขึ้น ดังจะเห็นในตารางที่ 3 จำนวนข้อมูลที่เหลือในแต่ละรอบของ GTMS1 ลดลงอย่างมาก เช่นในภาพ Vegetable เมื่อใช้ $\sigma = 100$ ในการคำนวณในรอบที่ 2 GTMS1 จะเหลือข้อมูลเพียง 15 ข้อมูล จากทั้งหมด 33,908 ข้อมูล ในขณะที่ GMS2 ข้อมูลเหลือ 23,797 ข้อมูล ซึ่งตัวอย่างผลการแยกส่วนประกอบของภาพทั้ง 5 ภาพแสดงดังรูปที่ 3

5. สรุปผลการทดลอง

บทความนี้ได้นำเสนอขั้นตอนวิธีการลำดับเชิงซ้ายค่าเฉลี่ยแบบครอบคลุมเพื่อช่วยในการเพิ่มความเร็วยังขั้นตอนวิธีการซ้ายค่าเฉลี่ยเดิม และลดข้อบกพร่องของขั้นตอนวิธีการลำดับเชิงซ้ายค่าเฉลี่ย โดยการลดการใช้กราฟในการแทนจุดข้อมูลในแต่ละรอบ และเพิ่มการพิจารณาทิศทางของผู้ขนส่งและผู้พ่วง โดยค่าที่ใช้ตัดสินใจความขนานกันของทิศทางทั้งสองข้อมูลมีความยืดหยุ่นในการกำหนดค่า โดยสามารถมีค่าเปลี่ยนแปลงตามความเหมาะสมของธรรมชาติของขั้นตอนวิธีการซ้ายค่าเฉลี่ยเมื่อจำนวนรอบมากขึ้น แล้วใช้เฉพาะผู้ขนส่งในการลำดับเชิงเพื่อคำนวณในรอบต่อไป โดยยังคงใช้ขั้นตอนวิธีการซ้ายตามค่าเฉลี่ยเดิมส่งผลให้ตำแหน่งตัวแทนในแต่ละรอบ เป็นตำแหน่งเดียวกันกับตำแหน่งของขั้นตอนวิธีการซ้ายตามค่าเฉลี่ยเดิมนั่นเอง ขั้นตอนวิธีที่นำเสนอนี้สามารถให้ผลลัพธ์เดียวกับขั้นตอนวิธีการซ้ายตามค่าเฉลี่ยหรือขั้นตอนวิธีการลำดับเชิงซ้ายตามค่าเฉลี่ย โดยมีความเร็วกว่า ขึ้นอยู่กับการกำหนดค่า α ที่ใช้ในการตัดสินใจเลือกผู้ขนส่ง จากผลการทดลองสามารถเพิ่มความเร็วยังการแยกส่วนประกอบภาพได้สูงถึง 3 – 333 เท่าเมื่อเทียบกับขั้นตอนวิธีการซ้ายค่าเฉลี่ยแบบเดิม ซึ่งขึ้นอยู่กับ σ , α และจำนวนข้อมูล

ตารางที่ 2 เปรียบเทียบความเร็วในการจัดแยกส่วนประกอบภาพของแต่ละขั้นตอนวิธี เมื่อเทียบกับขั้นตอนวิธีการซ้ายค่าเฉลี่ย

ข้อมูล	σ	เฉลี่ยใน 10 รอบ(วินาที)															
		MS		MS2		TMS				GTMS1				GTMS2			
		เวลา	เวลา	เวลา	Accuracy (%)	Speedup(เท่า) เมื่อเทียบกับ		เวลา	Accuracy (%)	Speedup(เท่า) เมื่อเทียบกับ		เวลา	Accuracy (%)	Speedup(เท่า) เมื่อเทียบกับ			
						MS	MS2			MS	MS2			MS	MS2		
1	500	747.73	73.19	9.92	98.35	75.35	7.38	2.98	98.35	250.53	24.52	12.92	98.74	57.84	5.66		
	200	643.89	64.35	10.17	97.13	63.27	6.32	3.15	97.13	204.13	20.40	9.20	98.77	69.98	6.99		
2	500	1442.80	134.22	14.35	99.44	100.52	9.35	4.32	99.44	333.81	31.05	12.05	99.52	119.73	11.14		
	200	1477.17	252.16	14.70	99.20	100.48	17.15	4.72	99.20	312.62	53.37	10.62	99.57	139.00	23.73		
3	500	500.00	75.61	12.36	97.97	40.43	6.11	3.82	97.97	130.66	19.76	13.20	98.64	37.85	5.72		
	200	1215.16	154.81	13.19	99.52	92.09	11.73	4.15	99.52	292.38	37.25	12.11	99.87	100.29	12.78		
4	500	304.60	49.20	5.57	96.04	54.62	8.82	1.84	96.04	165.51	26.74	6.30	96.36	48.33	7.81		
	200	577.83	57.29	6.72	94.12	85.92	8.52	2.42	94.12	238.41	23.64	6.97	95.60	82.85	8.22		
5	100	-	5327.93	-	-	-	-	604.82	99.97	-	8.81	1822.47	99.99	-	2.92		
	50	-	8499.35	-	-	-	-	606.04	99.96	-	14.02	1525.82	99.96	-	5.57		



รูปที่ 3 ผลของการแยกส่วนประกอบของภาพ

ตารางที่ 3 จำนวนข้อมูลที่เหลืออยู่ใน 5 รอบแรกของแต่ละขั้นตอนวิธี

ข้อมูล	σ	MS2	TMS	GMS1	GMS2
1	500	3100,3100,3100,3100,3100	3100,82,19,10,7	3100,82,19,10,7	3100,1822,1352,1224,1094
	200	3100,3100,3100,3100,3100,	3100,231,44,20,14	3100,231,44,20,14	3100,1964,1039,759,631
2	500	3750,3750,3750,3750,3750	3750,157,29,16,9	3750,157,29,16,9	3750,1432,991,829,749
	200	3750,3750,3750,3750,3750	3750,505,176,88,56	3750,505,176,88,56	3750,2168,1384,1087,839
3	500	3750,3750,3750,3750,3750	3750,104,20,13,10	3750,104,20,13,10	3750,2163,1254,1066,955
	200	3750,3750,3750,3750,3750	3750,361,97,46,27	3750,361,97,46,27	3750,2638,1655,1228,966
4	500	2500,2500,2500,2500,2500	2500,211,42,17,6	2500,211,42,17,6	2500,1556,985,720,593
	200	2500,2500,2500,2500,2500	2500,487,157,74,52	2500,487,157,74,52	2500,1812,1290,1016,827
5	100	33908,33908,33908,33908,33879	-	33908,15,7,3,3	33908,23797,20194,19456,19352
	50	33908,33908,33908,33908,33908	-	33908,46,14,10,7	33908,19443,12861,11407,10906

6. อ้างอิง

- [1] K. Fukunaga, L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp. 32 – 40, 1975.
- [2] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, August 1995.
- [3] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in Proc. Int'l Conf. Computer Vision, 1999, pp. 1197–1203.
- [4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603 – 619, May 2002.
- [5] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," in Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, 2002.
- [6] M. Á. Carreira-Perpiñán, "Fast nonparametric clustering with Gaussian blurring mean-shift," Proc. of the 23rd Int. Conf. on Machine learning table of contents, Pennsylvania, 2006.
- [7] P. Padungweing, S. Chiewchanwattana, and K. Sunat, "Resilient Mean Shift Algorithm," 13th Int. Conf. on Systems, Signals and Image Processing, Sep 21-23, Hungary, 2006.
- [8] K. Zhang, James T. Kwok, and M. Tang, "Accelerated Convergence Using Dynamic Mean Shift." In the 9th European Conference on Computer Vision (ECCV 2006), Graz, Austria, May 2006.
- [9] M. Á. Carreira-Perpiñán, "Acceleration Strategies for Gaussian Mean-Shift Image Segmentation," Proc. of the 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2006.
- [10] Pinda Padungweing, S. Chiewchanwattana, and K. Sunat, "Transport Mean Shift Algorithm," Proc. Of the Conf. Knowledge and Smart Technologies, July 24-25, Thailand, 2009.
- [11] M. Á. C. Perpiñán, "Mode finding for mixtures of Gaussian distributions," *IEEE Trans. PAMI*, vol. 22, Nov. 2000.
- [12] R.E. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146-160, 1972.
- [13] Z. Liying, Z. Jingtao, W. Qianyu, "Mean-shift-based color segmentation of images containing green vegetation," *Computers and Electronics in Agriculture*, vol. 65, pp. 93–98, 2009.

ภาคผนวก ค
บทความที่ผ่านการนำเสนอแบบปากเปล่า และตีพิมพ์ในวารสาร
รายงานการประชุมวิชาการ
ในงาน Future tech 2010 จัดขึ้นที่ Lotte Hotel เมืองบูซาน ประเทศเกาหลี



GTMS: A Simultaneous Mode seeking and Clustering

Panida Padungweang
Department of Computer Science
Khon Kaen University
Khon Kaen, Thailand
panida2m@gmail.com

Sirapat Chiewchanwattana
Department of Computer Science
Khon Kaen University
Khon Kaen, Thailand
sunkra@kku.ac.th

Khamron Sunat
Department of Computer Engineering
Mahanakorn University
Bangkok, Thailand
khamron_sunat@yahoo.com

Abstract—A mode seeking algorithm not only can automatically find mode of density of a given data but also can be used for data clustering. However, finding the mode of all data points produce redundant computations. In this paper, a simultaneous mode seeking and clustering which is called Generalized Transport Mean Shift (GTMS) algorithm was proposed. An idea of transportation was used for remedying the problem. For each iteration, the "transporter-trailer" characteristic was assigned to data points. The data points that tend to be shifted through the same trajectory, by considering shift direction, will be transported by the same transporter. Then they will be excluded from the computation in the next iteration. The transporters were, then, computed for finding a mode of density. The proposed algorithm was evaluated on clustering and image segmentation problems. The experimental results show that GTMS algorithm outperforms existing algorithms in both accuracy and time-consuming. It reduces redundancy computation by excluding data points more than 90% of image segmentation data after the second iteration only.

Keywords—Mean Shift algorithm; Generalized Transport Mean Shift algorithm; image segmentation; clustering; mode seeking

I. INTRODUCTION

Image segmentation and clustering are important processes in machine learning and pattern recognition. It is used for analyzing and investigating the given data nature. To acquire the solution, automatical finding mode of density of the given data is one of the powerful techniques. Normally the algorithm is informed by data's density. The local maximum of the density surface are assumed to be the modes. All data points are computed for finding their mode. Then, the data points which have the same mode can be assigned as same as cluster. The algorithm is useful for clustering [1, 2] image segmentation [3, 4] and tracking [5]. However, finding the mode of all data points require a repeated process which is time consuming. Thus, algorithm and its variation are extensively proposed to overcome this difficulty.

Mean shift algorithm is one of the powerful techniques for seeking modes of the given data. Standard mean-shift algorithm is the iterative procedure which can automatically find the mode of density of a data point. It begins with computing the weight of all points by function of their distances with the considering point. Then, compute the weighted mean point to get the shift position. This changed iteratively through the density raises up until the shift position is not changed, or be changed less than the acceptable distance. It is assumed that this position is the mode of the considering point and all data

points which converts into the same mode will be also assumed as same as cluster. However, the process of standard mean shift algorithm is very slow through the time complexity as $O(kn^2m)$ [6]. It is not suitable to be used, especially in the image segmentation application in which has very high data point. Improving the speed of the mean shift algorithm; therefore, is very important.

The speed up techniques for mean shift algorithm were frequently proposed. The speed up technique of the neural network learning was applied to mean shift algorithm and proposed in [7] which makes the acceleration more than it used to be. Moreover, the accurate result is the same. After that, the speed up of burring mean shift algorithm has been proposed [6] by clustering the data in each iteration and removing the cluster which has some data converted to its mode. Then, the next iteration will have less data so the computing follows faster. The proposed algorithm which finds the data cluster representation before making burring mean shift algorithm was proposed [8]. However, the burring mean shift algorithm can give different result compared with the standard mean shift algorithm. Since, the density estimation is computed by using the current position in each iteration but the density estimation of the standard one is usually computed by using the initial position of the given data set. Next, the improved speed up in mean shift algorithm for image segmentation was proposed [9]. There were four methods. Firstly, the neighborhood pixels are grouped into the same cell, and then the mean shift algorithm was used. The cell which is shifted in to the shifted cell in the previous iteration will stop computing and will be assumed as the same cluster. Secondly, the neighborhood pixel in spatial domain which was indicated by the specific distance are grouped into the same cluster and continued as the first method. Both methods had excellent speed. However, the wrong clustering result can occur even in the first step clustered the group of points. The other two methods were to approximate the E (Expectation) and M (Maximization) steps in each iteration by using a subset of the data and using the quadratic convergence technique which helped the less iteration. However it requires high computation. So it less helped to speed up as it should be. In many recent, Agglomerative Mean-shift clustering via query set compression [10] was proposed. The covering hyper ellipsoids were used to cluster data iteratively, which leads to hierarchical clustering via the mean shift process. The covering hyper ellipsoids need to compute the invert of the covariant matrix which is an extra cost and biased by data dimension. This algorithm also demonstrated the

use benefit of nature hill climbing algorithm that many data points are shifted through the same direction.

In this paper, Generalized Transport Mean Shift algorithm (GTMS) is proposed. The basic idea of mean – shift algorithm which is the shift process is presented in this algorithm. However, instead of finding the mode of all points, GTMS require few points called “transporter” which are representing their “trailers”. Moreover, finding the transporters does not require an extra cost. Since GTMS use the distance values which must be computed in the shift process. The trailers are the data points which are shifted in to the same mode of a transporter. The relationship “transporter-trailer” is investigated by considering direction of transporter’s trajectory and trailer’s shift direction. Then the trailers are excluded for the next iteration only the transporters are computed. In addition, a transporter can be assigned as a trailer in a next iteration this not only reduced the transporter to be computed but also performs a simultaneous hierarchical clustering.

The rest of this paper is organized as follows. In section II, we briefly explain the nature of mean shift algorithm and the generalized transport mean shift algorithm will be mentioned and proposed in Section III. Then, the experiment results on real-world clustering and image segmentation problem will be oriented in Section IV. Finally, Section V is the conclusion and future work

II. STANDARD MEAN SHIFT ALGORITHM

Let $\mathbf{X} \in R^m$ be a data set in an m -dimensional Euclidean space. $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ and $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T$. A probability density estimation of a given data \mathbf{x}_i is defined by

$$p(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n K \left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{\sigma} \right) \quad (1)$$

Where $K(t)$ is a kernel function and σ is a constant bandwidth such that $\sigma > 0$. A mode of the density is a position \mathbf{x} which $\nabla p(\mathbf{x}) = 0$. Mean shift is an iterative procedure for seeking the mode of density estimation with repeated shifting the position \mathbf{x}_i towards high density as

$$\mathbf{x}^{(r+1)} = f(\mathbf{x}^{(r)}) \quad (2)$$

with

$$f(\mathbf{x}) = \frac{\sum_{i=1}^n K' \left(\frac{\|\mathbf{x} - \mathbf{x}_i\|}{\sigma} \right) \mathbf{x}_i}{\sum_{j=1}^n K' \left(\frac{\|\mathbf{x} - \mathbf{x}_j\|}{\sigma} \right)} \quad (3)$$

where $K'(t) = dK/dt$. Using Gaussian function $K(t) = e^{-t^2}$, (2) and (3) can be reduced [11] to

$$\mathbf{x}^{(r+1)} = \sum_{i=1}^n p(i | \mathbf{x}^{(r)}) \mathbf{x}_i \quad (4)$$

and

$$p(i | \mathbf{x}^{(r)}) = \frac{\exp\left(-\frac{1}{2} \left\| \frac{\mathbf{x}^{(r)} - \mathbf{x}_i}{\sigma} \right\|^2\right)}{\sum_{j=1}^n \exp\left(-\frac{1}{2} \left\| \frac{\mathbf{x}^{(r)} - \mathbf{x}_j}{\sigma} \right\|^2\right)} \quad (5)$$

The algorithm terminates when the shift distance equal to zero or less than a tolerant threshold as follows:

$$\left\| \mathbf{x}^{(r)} - \mathbf{x}^{(r-1)} \right\| \leq \text{threshold} \quad (6)$$

The clustering is performed by representing each mode of the kernel density estimate as the cluster, and the data points are converted to their mode. Fig. 1 shows that, data points are shifted rising up to their mode. The solid black lines represent the trajectory of each data points.

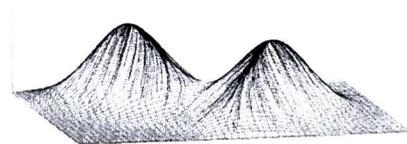


Figure 1. The trajectory of data point by applying mean shift algorithm to 2D dataset, the third axis denotes density of data

III. GENERALIZED TRANSPORT MEAN SHIFT ALGORITHM

In general, there are many positions for which are shifted through the same trajectory to place themselves at their mode as an example shown in the Fig. 1. Consider Fig. 2, the i^{th} data is shifted to the position which is closed to the original position of the k^{th} data at iteration r . Also, the direction of the shift vector of i^{th} data is paralleled to the trajectory vector of the k^{th} data. Therefore, the i^{th} data should be considered as the “trailer” of the k^{th} data which is assumed to be a “transporter” of the i^{th} data. Hence, the shifting of the i^{th} data need not be computed in the next iteration.

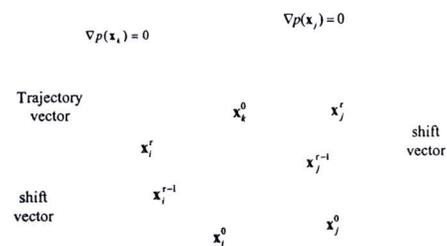


Figure 2. Trajectory vector of \mathbf{x}_k and shift direction of \mathbf{x}_i and \mathbf{x}_j at iteration r

Even though the j^{th} data at iteration r is also shifted to the position near the original position of the k^{th} data, its mode is different from the mode of the k^{th} data. One of the important ideas of this work is that, the most nearest point which is assigned as the transporter should have the same direction of trajectory vector as the direction of the shift vector of the trailer.

In order to acquire the solution, four matrixes are introduced. First matrix is a matrix of trajectory vector of all the data points. The second matrix stores indexes of the transporters. The last two matrixes are logical which indicate the convergence status and present status of the data points. The details of each matrix are as follows:

Let $\mathbf{U} \in R^{m \times n}$, $\mathbf{u}_i \in \mathbf{U}$ be a unit trajectory vector of the i^{th} data at the first iteration and can be computed as

$$\mathbf{u}_i = \frac{\mathbf{x}_i^1 - \mathbf{x}_i^0}{\|\mathbf{x}_i^1 - \mathbf{x}_i^0\|} \quad (7)$$

Let $\mathbf{T} \in \mathbf{1}^{1 \times n}$ be a transporter matrix, where $t_i \in \mathbf{T}$ denotes an index of a transporter of the i^{th} data, such that

$$t_i = \begin{cases} \arg \min_j (\|\mathbf{x}_i^r - \mathbf{x}_j^r\|^2) & \text{if } \delta_{ij}^r \leq \alpha \\ i & \text{otherwise} \end{cases} \quad (8)$$

where α is a constant threshold. δ_{ij}^r denotes the generalized angle between the trajectory vector (\mathbf{u}_i) and the shift vector (\mathbf{v}_j^r). $\delta \in [0, 1]$, is defined by

$$\delta_{ij}^r = \frac{1}{2} \left(1 - \frac{\mathbf{v}_j^r \cdot \mathbf{u}_i}{\|\mathbf{v}_j^r\| \cdot \|\mathbf{u}_i\|} \right) \quad (9)$$

Clearly, the vectors are parallel and has the same direction if $\delta = 0$. If $\delta = 1$ means the vectors are still parallel but has an opposite direction of (180 degree).

The last two matrix $\mathbf{C}_{1 \times n}$ be a matrix of the convergence status, $c_i \in \mathbf{C}$ and $\mathbf{A}_{1 \times n}$ be a matrix of the present status, $a_i \in \mathbf{A}$ are expressed as follows:

$$c_i = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ data converge} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$a_i = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ data should be present} \\ & \text{to the next iteration} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The Generalized Transport Mean Shift Algorithm (GTMS) is shown in **Algorithm 1**. The trajectory vector is computed in step 2 and is assigned in step 5. Step 6, 9 and 10 are normally performed by the mean shift algorithm. However if the transporter of the i^{th} data is found in step 8, step 9 need not be computed which is computing exponential of all distance values in step 6 using (5). Besides, the shift position of the i^{th} data is assigned to t_i . Therefore if there are many trailers found, there are few data that needs to be computed in the next iteration.

Algorithm1 (GTMS)

Initialization:
C is initialized to false.
A is initialized to true.
 Initialize the value of parameter α .
for each $\mathbf{x}_i \in \mathbf{X}$ **do**
 1. Compute \mathbf{x}_i^1
 2. Calculate the shift distance and the trajectory vector
 Set $\mathbf{z} = \mathbf{x}_i^1 - \mathbf{x}_i$
 Set $s = \|\mathbf{z}\|^2$
 Set $\mathbf{v}_i = \mathbf{z}/s^{0.5}$
 3. Considering the convergence
 if $s \leq \text{threshold}$ **then**
 $c_i = \text{true}$, $a_i = \text{false}$
 end if
 4. Set t_i to be a transporter itself, $t_i = i$
 end for
 5. Set $\tau = 1$, $\mathbf{U} = \mathbf{V}$ // \mathbf{u}_i is a unit trajectory vector
 while these are $a_i = \text{true}$ **do**
 6. Calculate the distance (d_k) from \mathbf{x}_i^r to $\mathbf{x}_k \in \mathbf{X}, 1 \leq k \leq n$
 7. Find the j^{th} data that nearest to the i^{th} data
 $j = \arg \min_k d_k$
 8. Investigate the transporter
 Compute $\delta = 1/2(1 - \mathbf{v}_j^r \cdot \mathbf{u}_i / \|\mathbf{v}_j^r\| \|\mathbf{u}_i\|)$
 if $i \neq j$ & $t_i \neq i$ & $\delta \leq \alpha$ **then**
 set $t_i = j$, $a_i = \text{false}$ // assign transporter
 and inactive trailer
 else
 9. Compute $\mathbf{x}_i^{\tau+1}$
 10. Follow step 2 and 3
 end if
 Increase the number of iteration by setting $\tau = \tau + 1$.
 endwhile

According to step 8, the transporter of the i^{th} is found and assigned to t_i . This means the i^{th} data should be the same mode as its transporter. Hierarchical clustering is simultaneously performed in this step using matrix **T**. At the convergence step the depth first search algorithm [12] is used for retrieving all trailers of the remaining transporters. At the convergence step, any cluster may have many transporters assigned in the same cluster. Thus, it would be easy to perform clustering for those transporters by using the distance threshold. The appropriate value of threshold is a proportion of the bandwidth of a density estimator. In this paper we use a half bandwidth to be the desired threshold.

Although there is an additional parameter (α) in our algorithm, this parameter makes the algorithm more flexible. It is used for choosing a suitable transporter-trailer relationship. Normally, α is in between [0, 1]. For $\alpha = 0$ the trailer is transported by the transporter only the shift vector of the trailer and the trajectory vector of the transporter are parallel. For $\alpha = 1$, the angle of those vectors is in range 0 -180 degrees is acceptable. Hence, the value of (9) is spanned in range [0,1]. However, the parameter α can be assigned out of this range. In

the case of $\alpha > 1$, the nearest position of shift position at the i^{th} data is always assigned to be the transporter of the i^{th} . In this case GTMS perform the fastest. If $\alpha < 0$, there is no transporter to be assigned. In this case GTMS performs as standard mean shift algorithm. Therefore, mean shift algorithm is considered as a special case of the GTMS.

The value of α can be adapted in each iteration. Because of the nature of a mode seeking algorithm, the data which belongs to a difference mode will be shifted far from each other when the number of iteration is increased. Moreover, we also provide a technique for adjusting α in this paper. Let α_0 be an initial value and α_r be a maximum acceptable threshold at iteration r . The value of α can be assigned as

$$\alpha = \min(\alpha_r, \alpha_0 + \frac{\alpha_r - \alpha_0}{\tau}i) \quad (12)$$

where i denotes the number of iteration. The value of α linearly be increased from α_0 to α_r when the number of iteration raises up to iteration τ .

Obviously, GTMS process much faster than mean shift algorithm, because the trailers are excluded for the next iteration. The time complexity of the GTMS is $O(nm \sum_{i=1}^p q_i)$, where $q_p \leq \dots \leq q_2 \ll q_1 = n$ and q_i denotes the number of active data point at iteration i . The space complexity of the GTMS is $O(nm)$.

IV. EXPERIMENTAL RESULTS

The GTMS was tested on clustering and image segmentation. In general, density estimator based algorithm need a desired bandwidth which is still an open problem and is beyond scope of this paper. So we select some suitable bandwidth for comparison. The algorithms were tested on same parameter and environment. There are two approaches of mean shift algorithm. In our experiment, MS1 denotes the standard mean shift algorithm as in [8]. MS2 denotes the standard mean shift algorithm with exclude converge points for the next iteration. GTMS1 denotes the fastest GTMS by choosing $\alpha > 1$ and GTMS2 denotes a GTMS with $\alpha_0 = 0$, $\alpha_r = 0.5$ and $\tau = 10$. The tolerate threshold is set to 10^{-7} for all experiments.

A. Clustering

We conduct experiments to evaluate the accuracy of the proposed algorithm. The algorithm was tested using real world data set. Several data sets from UCI repository database were selected, such as iris, soybean and blood. The bandwidth of the density estimator was selected by repeating applied standard mean shift algorithm with difference bandwidth until the number of cluster produced by the algorithm equal to the real one. Then every algorithm was performed using the same bandwidth. The average accuracies of the clustering results are shown in Table I and the speed up were shown in Fig. 3. We can observe that, GTMS algorithm not only was a fast algorithm but also achieved more accurate clustering results than the standard mean shift algorithm, especially the GTMS1 which was the fastest algorithm. Although for blood data set, every algorithm achieved the same accuracy result, GTMS1

performed faster than the other algorithms. For soybean data set GTMS1 archived 100% clustering accuracy with speeded up 8.31 times faster than MS1.

TABLE I. AVERAGE ACCURACY RESULTS ON 10 EXPERIMENTS OF THE CLUSTERING

Data set	Accuracy (%)			
	MS1	MS2	GTMS1	GTMS2
iris	62.67	62.67	84	82
soybean	98.87	98.87	100	98.87
blood	76.47	76.47	76.47	76.47

■ GTMS1 vs. MS1 ■ GTMS1 vs. MS2 ■ GTMS2 vs. MS1 ■ GTMS2 vs. MS2

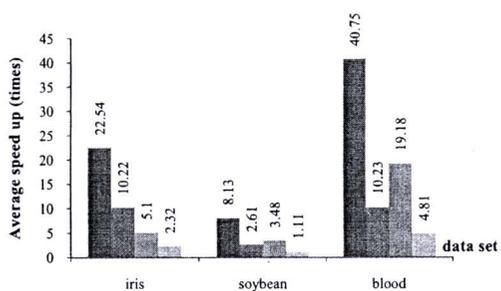


Figure 3. The averages speed up on 10 experiments of GTMS algorithm compare with MS1 and MS2, respectively.

B. Image segmentation.

The algorithms were evaluated on image segmentation using 5 well known image data sets, (1) Hand, (2) Lena, (3) Hawk, (4) House, and (5) vegetable [13]. A pixel of the first four images was decomposed to a 5-dimension vector (x, y, R, G, B). The first two dimensions were normalized to range [1,100] in order to reduce the effect of image's size. The last image was decomposed as in [13], (x, y, G-R, G-B, H, S, I) with (x, y) were normalized to range [1, 20]. The average time-consuming on 10 experiments and the speed up of GTMS with difference bandwidth of density estimator were shown in Table II.

From Table II, we can see that GTMS1 achieved speed up for 130-333 times faster than MS1 and 8-53 times faster than MS2, respectively. Moreover, GTMS2 which performed result approximate to MS, achieved speed up rising to 139 and 23 times faster than MS1 and MS2, respectively. Because the vegetable image consists of 33,908 pixels and the progression of MS1 was very slow, we did not perform this problem. Consider Fig. 4, for vegetable image, the number of active data points of GTMS1 in the second iteration remained only 15 data points. This means that there were huge trailers which were shifted to the same direction of the transporter. GTMS algorithm excluded those points for the next iteration while MS still computed a shift direction of those. MS2 all points remained since there were no converged points in this iteration. The number of active data points of the first five iteration of all image using different algorithm were shown in Table III. It can

be seen that GTMSI gave the fewest active data points at iteration 5 for all images. The results of the image segmentation were shown in Table IV. MS1 and MS2 performed the same final results which were shown in the row of MS. Finally, we can observe that GTMSI significantly reduced unnecessarily computation when the iteration increases. Less than 10% of data points of all images remained at the end of the second iteration using GTMSI. For Vegetable image, there were only 0.04% of data points still active at the end of the second iteration.

V. CONCLUSION

In this paper, we have proposed an unsupervised simultaneous mode seeking and clustering algorithm, called GTMS, based on the idea of transportation. The data points which are shifted through the same trajectory are transported by the same transporter and excluded for the next iteration. The GTMS algorithm considers the direction of the transporter and the trailer shift vector in order to determine a reasonable relationship. Also, GTMS is flexible by adjusting the acceptable direction threshold (δ). Mean shift algorithm can be viewed as a special case of GTMS algorithm for $\alpha < 0$. Experimental evidences show that, GTMS not only performs faster but also achieve more accuracy than MS for clustering.

For image segmentation, GTMS performs faster than MS1 for 3-333 times, providing a useful tool for image processing. In addition, the active data points are rapid reduce when the iteration increase for GTMSI. The algorithm should perform faster than this. Particularly the last image, it speed up only 8.81 time while the data point remain only 15 from 33,908 points. We are currently exploring the ability for more faster algorithm.

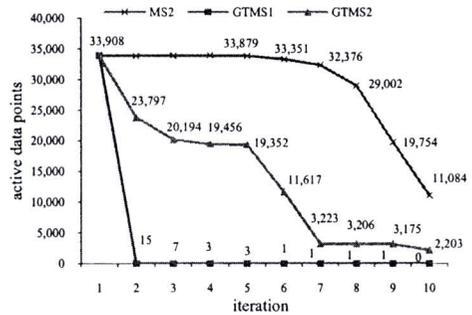


Figure 4. The scatter plot of the number of the active data points at each iteration for Vegetable image with bandwidth 100 is shown.

TABLE II. THE AVERAGE TIME-CONSUMING OF IMAGE SEGMENTATION ON 10 EXPERIMENTS AND THE SPEED UP OF GTMS ALGORITHM BE COMPARED WITH MS1 AND MS2, RESPECTIVELY.

Data set	σ	The average time (second)							
		MS1		MS2		GTMS1		GTMS2	
		Time	Time	Time	Speedup	Time	Speedup	Time	Speedup
Hand	500	747.73	73.19	2.98	250.53	24.52	12.92	57.84	5.66
	200	643.89	64.35	3.15	204.13	20.40	9.20	69.98	6.99
Lena	500	304.60	49.20	1.84	165.51	26.74	6.30	48.33	7.81
	200	577.83	57.29	2.42	238.41	23.64	6.97	82.85	8.22
Hawk	500	1442.80	134.22	4.32	333.81	31.05	12.05	119.73	11.14
	200	1477.17	252.16	4.72	312.62	53.37	10.62	139.00	23.73
Houses	500	500.00	75.61	3.82	130.66	19.76	13.20	37.85	5.72
	200	1215.16	154.81	4.15	292.38	37.25	12.11	100.29	12.78
Vegetable	100	-	5327.93	604.82	-	8.81	1822.47	-	2.92
	50	-	8499.35	606.04	-	14.02	1525.82	-	5.57

TABLE III. THE NUMBER OF ACTIVE DATA POINTS OF THE FITST FIVE ITERATION FOR THREE ALGORITHMS, THOSE ARE MS2, GTMS1 AND GTMS2 ARE SHOWN.

Data set	σ	MS2	GTMS1	GTMS2
Hand	500	3100,3100,3100,3100,3100	3100,82,19,10,7	3100,1822,1352,1224,1094
Lena	500	2500,2500,2500,2500,2500	2500,211,42,17,6	2500,1556,985,720,593
Hawk	500	3750,3750,3750,3750,3750	3750,157,29,16,9	3750,1432,991,829,749
Houses	500	3750,3750,3750,3750,3750	3750,104,20,13,10	3750,2163,1254,1066,955
Vegetable	100	33908,33908,33908,33908,33879	33908,15,7,3,3	33908,23797,20194,19456,19352

TABLE IV. THE RESULTS OF IMAGE SEGMENTATION USING BANDWIDTH AS IN TABLE III

	1. Hand 62x50	2. Lena 50x50	3. Hawk 75x50	4. Houses 75x50	5. Vegetable 176x173
Original image					
MS					
GTMS1					
GTMS2					

REFERENCES

- [1] K. Fukunaga, L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. 21, pp. 32–40, 1975.
- [2] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, August 1995.
- [3] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in Proc. Int'l Conf. Computer Vision, 1999, pp. 1197–1203.
- [4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
- [5] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," in Statistical Methods in Video Processing Workshop, Copenhagen, Denmark, 2002.
- [6] M. Á. Carreira-Perpiñán, "Fast nonparametric clustering with Gaussian blurring mean-shift," Proc. of the 23rd Int. Conf. on Machine learning table of contents, Pennsylvania, 2006.
- [7] P. Padungweing, S. Chiewchanwattana, and K. Sunat, "Resilient Mean Shift Algorithm," 13th Int. Conf. on Systems, Signals and Image Processing, Sep 21-23, Hungary, 2006.
- [8] K. Zhang, James T. Kwok, and M. Tang. "Accelerated Convergence Using Dynamic Mean Shift," In the 9th European Conf. on Computer Vision, Graz, Austria, May 2006.
- [9] M. Á. Carreira-Perpiñán, "Acceleration Strategies for Gaussian Mean-Shift Image Segmentation," Proc. of the 2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2006.
- [10] Xiao-Tong Yuan, Bao-Gang Hu, and Ran He, "Agglomerative Mean-Shift Clustering via Query Set Compression," Proc. of the Ninth SIAM Int. Conf. on Data Mining, Nevada, 2009.
- [11] M. Á. C. Perpiñán, "Mode finding for mixtures of Gaussian distributions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, Nov. 2000.
- [12] R.E. Tarjan, "Depth first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146-160, 1972.
- [13] Z. Liying, Z. Jingtao, W. Qianyu, "Mean-shift-based color segmentation of images containing green vegetation," *Computers and Electronics in Agriculture*, vol. 65, pp. 93–98, 2009.

ประวัติผู้เขียน

นางพนิดา ผดุงเวียง เกิดเมื่อวันที่ 21 กันยายน 2521 สำเร็จการศึกษาระดับปริญญาตรีจาก ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อ พ.ศ.2543 และระดับปริญญาโท จากสาขาวิทยาศาสตร์ศึกษา คณะศึกษาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อ พ.ศ.2545 และได้ศึกษาต่อในระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น เมื่อ พ.ศ.2551



