Análisis de Errores de Posicionamiento en Formación de Patrones 2D Mediante Auto-ensamblado en Enjambres de Robots

Juan Eizaguerri Serrano

Universidad Politécnica de Madrid, España
Máster Universitario en Inteligencia Artificial, Sistemas Multiagente
juan.eizaguerri@alumnos.upm.es

Abstract. Este trabajo se analizan los errores de posicionamiento en la formación de patrones 2D mediante el autoensamblado de enjambres de robots. Basándose en el sistema multiagente descentralizado propuesto por Rubenstein et al., el trabajo investiga las causas de las deformaciones en los patrones formados por los Kilobots, atribuidas a errores acumulativos en la localización de los robots. Se desarrolla un entorno de simulación sobre el que se realizan distintas propuestas para solucionar las deformaciones, incluyendo reducir la velocidad de los robots, aumentar el rango de comunicación y mantener la localización después de que los robots se unan al patrón. Los resultados experimentales indican que mantener la localización tras la fijación de los robots a la figura reduce significativamente los errores, ofreciendo una solución práctica para mejorar la precisión de los patrones.

1 Introducción

La formación de patrones 2D mediante robots es un campo ampliamente investigado por sus posibles aplicaciones en diversas áreas. Las aportaciones de Rubenstein et al. [3] destacan en este ámbito por su gran escalabilidad y tolerancia a fallos. Utilizando tres algoritmos básicos, proponen un sistema multiagente descentralizado funcional en robots de capacidades muy restringidas como son los *Kilobots* [2]. Este método ha demostrado ser eficaz para formar una amplia variedad de figuras de diversos tamaños y formas, aunque con algunas limitaciones como se discute en [1]. Sin embargo, es apreciable a simple vista que las figuras formadas habitualmente se presentan ligeramente giradas o dobladas respecto al patrón que se quiere formar. Este problema es comentado brevemente en el artículo original y atribuido a errores acumulados, pero no se discuten en profundidad las posibles causas ni se proponen soluciones al problema.

El objetivo de este trabajo es por tanto, en primer lugar, crear un entorno de pruebas (Sección 2) y buscar el origen de la deformidad en las figuras (Sección 3), y por otro lado, proponer soluciones para eliminar o suavizar este fenómeno (Sección 4), analizar los resultados y realizar conclusiones (Sección 5).

2 Formación de patrones 2D mediante enjambres de robots

Las pruebas a realizar requieren un entorno de pruebas que replique el utilizado en [3]. Al no disponer de un entorno físico se ha implementado una simulación en Python siguiendo los algoritmos de formación de gradiente, seguimiento de borde, trilateración y comportamiento general del artículo original. Con el objetivo de simular un entorno realista, se modela un error en el desplazamiento y giro de los robots que da lugar a fenómenos observados en el entorno real como los "atascos de tráfico". De forma análoga se incluye error en la medición de distancia entre robots vecinos, que juega un papel importante en el algoritmo de trilateración de los robots. No se han tenido en cuenta posibles errores de pérdida de mensajes. Por simplicidad, se utilizan píxeles como unidad de medida del espacio. Se realizan pasos de simulación de 100ms. Se implementa también un algoritmo sencillo de detección y resolución de colisiones entre robots.

Al realizar pruebas en el entorno desarrollado, se observa una deformidad en las figuras muy similar a la documentada por Rubenstein et al. (Ver Figura 1a). Es fácil ver que este problema se debe a un aumento del error de localización en función del tiempo como el observado en las gráficas de la Figura 1.

3 Análisis del problema

Para argumentar la posible causa de este problema se ha tomado un enfoque experimental en el que se han eliminado marginalmente los posibles orígenes del error acumulado comprobando los resultados obtenidos.

Parece poco probable que el problema provenga del error de movimiento pues la localización de los robots depende completamente de la comunicación con los vecinos al no utilizarse odometría de ningún tipo. Por otro lado, es de esperar que el error en la medición de distancia entre vecinos juegue un papel importante en el error de localización, pues juega un papel importante en el algoritmo de trilateración utilizado para estimar la posición de los robots. Sin embargo, como se puede ver en la Figura 1d, en ninguna de estas pruebas se obtiene una mejora significativa, ni siquiera eliminando ambos errores al mismo tiempo.

Como última prueba, se evalúa la posibilidad de que el error acumulado provenga del algoritmo de trilateración. Se ejecuta la simulación utilizando información perfecta sobre las posiciones de los robots. Los resultados de las Figuras 1c y 1e muestran que efectivamente se elimina el error de localización y con ello el problema de la deformación de la figura.

Sería también lógico pensar que el error no aumenta por acumulación, sino por la distancia a la semilla de los robots que se van colocando. Para descartar esta posibilidad se diseña una prueba en la que se utiliza localización perfecta durante la primera mitad de la simulación y el algoritmo de trilateración durante la segunda. Si la distancia a la semilla fuese el problema, el error aumentaría de golpe una vez activada la trilateración, cosa que no ocurre, sino que empieza a acumularse desde ese punto (ver Figura 1c).

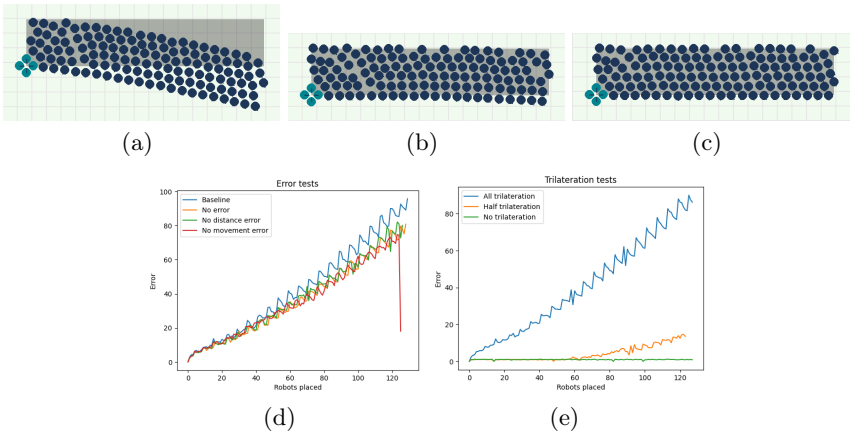

Fig. 1. Figura formada mediante (a) algoritmo de trilateración a lo largo de toda la ejecución, (b) localización perfecta en la primera mitad y trilateración en la segunda, y (c) localización perfecta durante toda la ejecución. Error de localización a lo largo del tiempo. (d) Sin errores, con errores, y sólo con error de movimiento o distancia, y (e) con y sin trilateración.

El método de trilateración propuesto por Rubenstein et al. es una solución iterativa a un problema de minimización. Debido a la naturaleza de este tipo de algoritmos, se requiere de múltiples iteraciones para llegar a una solución cercana a la óptima. Esta tarea se dificulta al realizarse mientras el robot está en movimiento. Analizando el vector formado entre la posición real de cada robot y su posición percibida mediante trilateración se puede observar que la estimación siempre se queda cierta distancia por detrás del robot a lo largo de su recorrido. Esto se convierte en un problema en el momento en el que el robot pasa al estado *joined_ shape* y se fija en la figura, dejando de localizarse y quedándose por tanto con su última posición algo por detrás de la real. Esta posición es utilizada como punto de referencia en la trilateración de los robots siguientes, aumentando así su error de localización acumulativamente.

4 Métodos propuestos

Se proponen las siguientes pruebas para tratar de eliminar o reducir el problema planteado.

1. **Reducir la velocidad de movimiento de los robots:** Dado que el problema proviene del hecho de que la posición calculada va siempre cierta distancia por detrás de la real, cabe la posibilidad de que reduciendo la velocidad de los robots se realicen suficientes iteraciones del algoritmo de trilateración como para cacular una solución con un error poco significativo. Se miden los resultados para distintos valores de velocidad v, modificando también proporcionalmente la velocidad angular w de los robots.

2. **Aumentar el radio de comunicación con los vecinos:** La posición estimada se calcula a partir de todos los vecinos con los que se comunica el robot. Aumentar el radio de comunicación (y por tanto el número de vecinos) puede acelerar la modificación de la posición, y por otro lado, actuar como elemento suavizador de los errores de localización de los vecinos. Se estudia el efecto de distintos radios de comunicación.
3. **Mantener la localización tras fijar robots:** Si el problema es generado simplemente por el error de localización que existe en el momento de fijar el robot dentro de la figura, permitir que siga realizando trilateración cuando el robot ya no se está moviendo podría permitir reducir en gran medida el error de localización.

5 Resultados y conclusiones

Se realizan y analizan las pruebas propuestas en la sección anterior. La Figura 2 muestra las gráficas de error en función del tiempo para los distintos valores probados de velocidad de movimiento y rotación, radio de comunicación de vecinos, y segundos de trilateración tras unirse a la figura. Por otro lado, en la Tabla 1 se encuentra un resumen de todas las pruebas realizadas, mostrando el número de robots final que forman la figura, el tiempo de simulación empleado, y el error medio de localización junto con su desviación estándar.

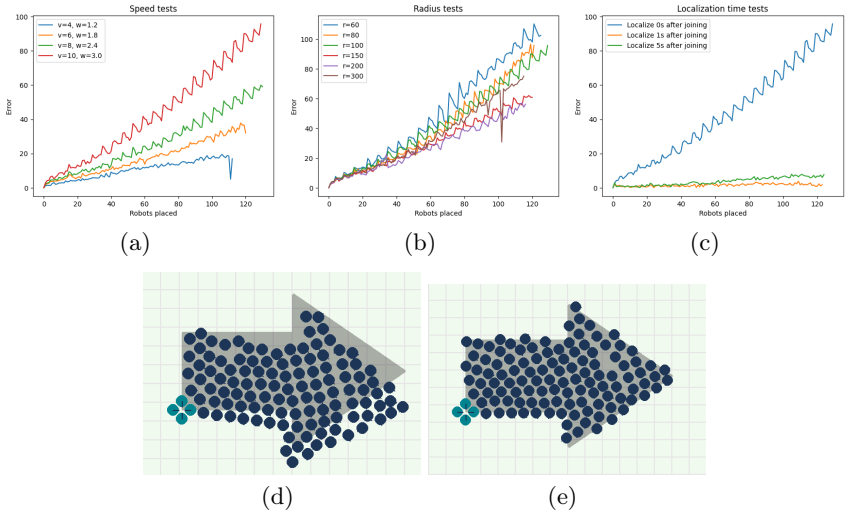

Fig. 2. Gráficas de error en función de (a) velocidad de desplazamiento y rotación, (b) radio de vecindad, (c) segundos de trilateración tras unirse a la figura. (d) Figura con algoritmo base, (e) Figura formada aplicando localización 1s.

Test	n robots	Tiempo (s)	Error medio	Std error
baseline	130	2414	43.24	26.83
trilateration_all	128	2268	40.75	25.64
trilateration_half	124	2263	4.01	4.07
trilateration_none	128	2301	0.96	0.20
no movement error	126	2272	35.29	20.76
no distance error	127	2302	37.20	22.74
no error	129	2208	36.86	22.77
v=4, w=1.2	113	4814	10.36	5.33
v=6, w=1.8	121	3451	17.43	10.05
v=8, w=2.4	131	2831	27.13	17.83
r=60	126	1556	49.75	32.26
r=80	122	1962	39.91	27.12
r=150	121	3399	30.71	17.88
r=200	117	4622	27.45	15.68
r=300	116	7291	33.10	21.21
localize 1s	124	2386	1.57	0.76
localize 5s	125	2243	3.76	2.26

Table 1. Resumen de los experimentos.

En primer lugar, cabe destacar que todas las pruebas han sido sorprendentemente consistentes en cuanto al número de robots empleados para formar la figura, con una desviación típica de tan sólo 4.95 robots. En cuanto al error de localización, se observa que reducir la velocidad de los robots disminuye el error aunque aumenta significativamente el tiempo de ejecución haciendo poco viable esta opción. Aumentar el radio de comunicación de vecinos también tiene ele efecto esperado, disminuyendo el error, aunque se deja de obtener mejora a partir de $r = 150$, incluso empeorando los resultados para valores más altos. Mantener la localización tras fijar los robots en la figura es sin duda la mejor opción encontrada, eliminando casi por completo el error acumulado (y con ello la deformación de la figura como se ve en la Figura 2e) cuando se mantiene durante un segundo. Para valores más altos las interacciones entre los robots ya fijados hacen que sus posiciones sean menos estables aumentando el error.

Como conclusión, se atribuyen las deformaciones observadas en las figuras en el algoritmo de formación de Rubenstein et al. a un error acumulativo fruto del método de localización iterativo utilizado, se ha estudiado el impacto de distintos parámetros sobre este error y se ha encontrado una solución al problema modificando el comportamiento de los robots de forma que continúen localizándose durante un corto tiempo tras fijarse a la figura. Como línea de trabajo futura, sería interesante aplicar esta propuesta en un sistema real para verificar su validez.

Referencias

1. Niazi, M.A.: Technical problems with" programmable self-assembly in a thousand-robot swarm". arXiv preprint arXiv:1708.03341 (2017)
2. Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: 2012 IEEE international conference on robotics and automation. pp. 3293–3298. IEEE (2012)
3. Rubenstein, M., Cornejo, A., Nagpal, R.: Programmable self-assembly in a thousand-robot swarm. Science **345**(6198), 795–799 (2014)