

HASIL ANALISIS

GLOBAL PLANNER

1. Greedy Best-First Search (GBFS):

Karakteristik: GBFS memilih jalur berdasarkan heuristik yang memperkirakan jarak terpendek ke tujuan, tanpa mempertimbangkan biaya yang telah ditempuh.

Analisis: Dalam animasi, GBFS cenderung memilih jalur yang tampak langsung menuju tujuan. Namun, karena tidak mempertimbangkan biaya total, jalur yang dihasilkan mungkin tidak optimal dan dapat terjebak dalam area dengan hambatan, mengakibatkan jalur yang lebih panjang atau tidak efisien.

2. Dijkstra:

Karakteristik: Algoritma ini mencari jalur terpendek dengan mempertimbangkan biaya total dari titik awal ke setiap node, tanpa menggunakan heuristik.

Analisis: Animasi menunjukkan Dijkstra mengeksplorasi area secara ekstensif, menghasilkan jalur yang optimal dalam hal biaya total. Namun, eksplorasi yang luas dapat menyebabkan waktu komputasi yang lebih lama, terutama pada peta besar atau kompleks.

3. A*:

Karakteristik: A* menggabungkan pendekatan Dijkstra dengan heuristik, mempertimbangkan biaya total dan perkiraan jarak ke tujuan.

Analisis: Dalam animasi, A* menunjukkan eksplorasi yang lebih efisien dibandingkan Dijkstra, dengan fokus pada area yang lebih mungkin mengarah ke tujuan. Hal ini menghasilkan jalur optimal dengan waktu komputasi yang lebih efisien.

4. Jump Point Search (JPS):

Karakteristik: JPS adalah optimasi dari A* yang mengurangi jumlah node yang dieksplorasi dengan melompati titik-titik tertentu dalam grid.

Analisis: Animasi JPS menunjukkan eksplorasi yang lebih cepat dan jalur yang lebih lurus, terutama pada grid dengan banyak ruang terbuka. Ini meningkatkan efisiensi komputasi tanpa mengorbankan optimalitas jalur.

5. D*:

Karakteristik: Dirancang untuk lingkungan yang berubah, D* memperbarui jalur secara dinamis saat informasi baru tersedia.

Analisis: Animasi D* menunjukkan kemampuan algoritma untuk menyesuaikan jalur saat hambatan baru muncul, memastikan robot tetap menuju tujuan meskipun ada perubahan di lingkungan.

6. Lifelong Planning A* (LPA*):

Karakteristik: LPA* adalah varian dari A* yang efisien dalam memperbarui jalur ketika ada perubahan kecil di lingkungan.

Analisis: Animasi LPA* menunjukkan kemampuan untuk memperbarui jalur dengan cepat saat terjadi perubahan, mengurangi kebutuhan untuk perhitungan ulang secara menyeluruh.

7. D* Lite:

Karakteristik: Versi sederhana dari D*, D* Lite menawarkan efisiensi dalam perencanaan ulang jalur di lingkungan yang dinamis.

Analisis: Animasi D* Lite menunjukkan respons cepat terhadap perubahan lingkungan, dengan perencanaan ulang jalur yang efisien dan efektif.

8. Voronoi:

Karakteristik: Membagi ruang menjadi wilayah berdasarkan jarak terdekat ke titik tertentu, menghasilkan jalur yang umumnya menghindari hambatan.

Analisis: Jalur yang dihasilkan cenderung menjaga jarak aman dari hambatan, namun mungkin tidak optimal dari segi panjang atau kelengkungan.

9. Theta*:

Karakteristik: Theta* memungkinkan jalur dengan sudut bebas, tidak terbatas pada gerakan grid, menghasilkan jalur yang lebih alami.

Analisis: Animasi Theta* menunjukkan jalur yang lebih halus dan langsung, mengurangi panjang jalur dan meningkatkan efisiensi perjalanan.

10. Lazy Theta*:

Karakteristik: Variasi dari Theta* yang menunda pengecekan jalur yang jelas hingga diperlukan, untuk efisiensi.

Analisis: Jalur yang dihasilkan lebih langsung, dengan kelancaran pada sudut, dan pengurangan biaya komputasi jika dibandingkan dengan Theta* biasa.

11. S-Theta*:

Karakteristik: Algoritma ini memadukan Theta* dengan heuristik yang memperbaiki kelancaran jalur pada lingkungan grid.

Analisis: Menghasilkan jalur yang lebih halus dan terarah dibandingkan Theta*, cocok untuk jalur dengan pembatasan kelengkungan.

12. Hybrid A*:

Karakteristik: Menggabungkan A* dengan pertimbangan dinamika kendaraan.

Analisis: Jalur yang dihasilkan dapat dieksekusi oleh kendaraan nyata karena memperhatikan batasan gerak, tetapi membutuhkan lebih banyak komputasi dibandingkan A* biasa.

13. Rapidly-Exploring Random Tree (RRT):

Karakteristik: RRT adalah algoritma sampling-based yang membangun pohon secara acak untuk menemukan jalur dari awal ke tujuan.

Analisis: Animasi RRT menunjukkan eksplorasi cepat di area luas, namun jalur yang dihasilkan mungkin tidak optimal dan memerlukan post-processing untuk perbaikan.

14. RRT*:

Karakteristik: Peningkatan dari RRT, RRT* mencari jalur optimal dengan mengoptimalkan pohon secara iteratif.

Analisis: Animasi RRT* menunjukkan jalur yang lebih optimal dibandingkan RRT, meskipun dengan waktu komputasi yang lebih lama.

15. Informed RRT*:

Karakteristik: Versi RRT* yang mengoptimalkan eksplorasi area yang relevan untuk mencapai tujuan.

Analisis: Mengurangi waktu komputasi dengan mengarahkan pencarian ke wilayah yang lebih informatif, menghasilkan jalur lebih cepat.

16. RRT Connect:

Karakteristik: Modifikasi RRT yang membangun dua pohon dari titik awal dan tujuan, kemudian menghubungkannya.

Analisis: Menemukan jalur lebih cepat dalam area terbuka, namun tidak optimal untuk lingkungan sempit.

17. Ant Colony Optimization (ACO):

Karakteristik: ACO meniru perilaku semut dalam mencari jalur terpendek melalui feromon, digunakan untuk menemukan jalur optimal.

Analisis: Animasi ACO menunjukkan konvergensi bertahap menuju jalur optimal, dengan jalur yang dihasilkan menjadi lebih efisien seiring waktu.

18. Genetic Algorithm (GA):

Karakteristik: Menggunakan proses evolusi untuk menemukan jalur optimal melalui seleksi dan mutasi.

Analisis: Dapat menghasilkan jalur optimal, namun memerlukan banyak iterasi dan waktu komputasi, terutama untuk lingkungan yang kompleks.

19. Particle Swarm Optimization (PSO):

Karakteristik: PSO menggunakan populasi partikel yang bergerak dalam ruang solusi untuk menemukan jalur optimal.

Analisis: Animasi PSO menunjukkan partikel-partikel yang berinteraksi dan berkumpul menuju jalur optimal, dengan konvergensi yang dipengaruhi oleh parameter algoritma.

LOCAL PLANNER

1. PID (Proportional-Integral-Derivative):

Karakteristik: Algoritma kontrol sederhana yang mengontrol kecepatan dan arah dengan mempertimbangkan error.

Analisis: Cocok untuk jalur langsung dengan perubahan kecil, tetapi tidak ideal untuk lingkungan dengan belokan tajam.

2. Linear Quadratic Regulator (LQR):

Karakteristik: Menggunakan kontrol optimal untuk menstabilkan jalur pada sistem linier.

Analisis: Menghasilkan pergerakan yang halus namun memerlukan tuning, dan kurang efektif untuk lingkungan nonlinier atau dengan batasan ketat.

3. Dynamic Window Approach (DWA):

Karakteristik: Perencanaan lokal yang mempertimbangkan kecepatan dan dinamika untuk menentukan jalur terdekat.

Analisis: Efektif dalam lingkungan dinamis, namun terbatas pada jangkauan lokal dan membutuhkan algoritma perencanaan global.

4. Artificial Potential Field (APF):

Karakteristik: Menggunakan potensi atraktif ke tujuan dan potensi repulsif dari hambatan.

Analisis: Cepat dan sederhana namun rentan terhadap masalah minimum lokal, dapat membuat robot terjebak.

5. Rapid Path Planner (RPP):

Karakteristik: Algoritma berbasis sampling untuk menemukan jalur di lingkungan besar.

Analisis: Cepat pada lingkungan terbuka, tetapi hasilnya cenderung lebih kasar dan membutuhkan pasca-pemrosesan untuk kelancaran.

6. Timed Elastic Band (TEB):

Karakteristik: Memodelkan jalur sebagai pita elastis yang meminimalkan biaya waktu dengan mempertimbangkan hambatan.

Analisis: Cocok untuk perencanaan dinamis dan menghindari hambatan secara real-time, namun membutuhkan tuning parameter.

7. Model Predictive Control (MPC):

Karakteristik: Menggunakan prediksi untuk menghasilkan jalur optimal yang mempertimbangkan dinamika sistem.

Analisis: Sangat efektif untuk kontrol kendaraan otonom, tetapi memerlukan komputasi intensif.

8. Lattice:

Karakteristik: Menggunakan grid yang memungkinkan manuver dengan mempertimbangkan dinamika kendaraan.

Analisis: Jalur yang dihasilkan sesuai dengan batasan kinematik kendaraan, cocok untuk kendaraan besar atau jalur di ruang terbatas.

CURVE GENERATION

1. Polynomial:

Karakteristik: Menggunakan fungsi polinomial untuk merencanakan jalur yang halus dan teratur.

Analisis: Menghasilkan jalur yang lancar, ideal untuk aplikasi yang membutuhkan perubahan arah yang halus.

2. Bezier Curve:

Karakteristik: Menggunakan kurva Bezier untuk menghasilkan jalur yang mulus.

Analisis: Memberikan kontrol tinggi atas bentuk jalur, cocok untuk robot dengan batasan kelengkungan.

3. Cubic Spline:

Karakteristik: Menggunakan splines kubik untuk menghasilkan jalur yang halus.

Analisis: Memberikan transisi yang mulus antar titik kontrol, cocok untuk perencanaan jalur dengan banyak titik.

4. B-Spline:

Karakteristik: Membuat jalur mulus dengan fungsi basis yang menawarkan fleksibilitas tinggi.

Analisis: Sangat fleksibel dan dapat disesuaikan dengan berbagai kendala.

5. Dubins Path:

Karakteristik: Digunakan untuk kendaraan dengan batasan radius putar minimum.

Analisis: Cocok untuk kendaraan seperti mobil, menghasilkan jalur yang sesuai dengan batasan arah dan kelengkungan.

6. Reeds-Shepp Path:

Karakteristik: Memperluas Dubins dengan kemampuan untuk mundur, cocok untuk kendaraan dengan putar balik.

Analisis: Menghasilkan jalur yang praktis untuk kendaraan yang membutuhkan manuver mundur atau putar balik di ruang sempit.