Proposed Algorithm Based on Inertial Measurement for Localization

Shouyi Zhang, Neeli Prasad, May Huang, Jian Cao, Xing Zhang

Outline

- Background of the project
- Research objectives and content
- Related technology introduction
- Proposed Algorithm
- Experiments and results
- Conclusion

Background of the project

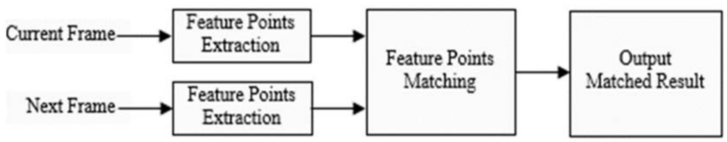
- Simultaneous Localization and Mapping (SLAM) is a technology for mobile robots to construct and generate maps of surrounding environments. The map generated by this technique determines the position of the robot in the environment and makes path planning for robot navigation.
- SLAM has the advantage of synchronously generating geometrically consistent environmental maps and locating robots' landmark locations, and has become a hot research topic in the field of autonomous mobile robots.
- The SLAM problem based on the RGB-D camera is called RGB-D SLAM.
- In the RGB-D SLAM process, the feature point matching process is time consuming, which makes the algorithm difficult to meet real-time performance.

Research objectives and content

- Objectives
  - Optimize the feature point matching process of RGB-D SLAM algorithm, and complete the design and implementation of image feature point prediction algorithm based on inertial measurement unit (IMU).
- Content
  - In the traditional algorithm, the IMU sensor is used to measure the angular velocity and linear acceleration of the camera from the current frame to the next frame, and the coordinates of the current frame image feature point in the next frame image are predicted according to the plane homography.
  - When matching feature points, there is no need to traverse the entire image, and only need to traverse within a small range near the predicted point to find matching feature points.

Related technology introduction

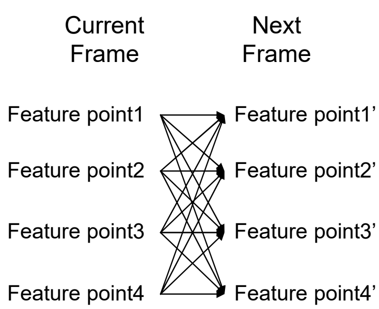
- Traditional feature point matching algorithm

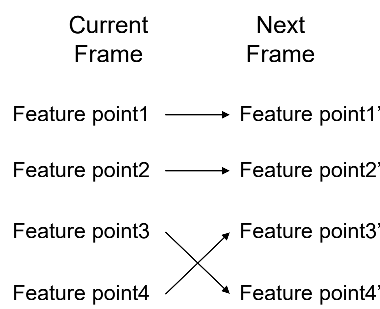


- Disadvantages:

For the two sets of feature points obtained by the feature point detection algorithm for the current frame image and the next frame image, it has to perform a feature point matching algorithm in the feature point matching process to determine whether it is a matching feature point, which is very time consuming.

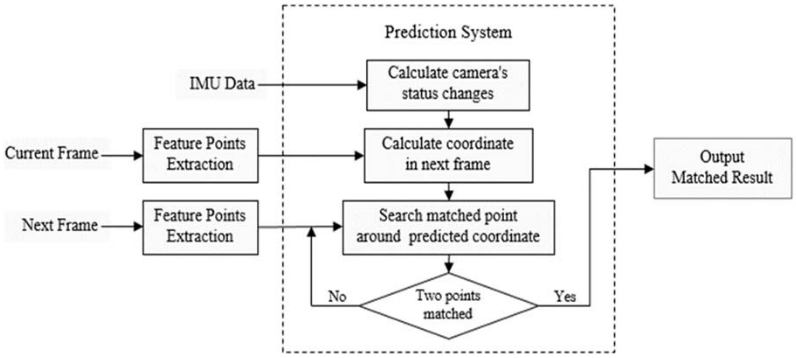
Related technology introduction





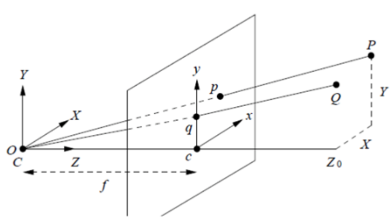
	Feature point1	Feature point2	Feature point3	Feature point4
Current Frame	(25, 40)	(100, 20)	(50, 70)	(40, 80)
Next Frame	(26, 38)	(99, 22)	(37, 79)	(49, 69)

Proposed Algorithm



Proposed algorithm flow chart

Proposed Algorithm



Pinhole camera model

$$u = \frac{x \cdot f_x}{2} + c_x$$

fx, fy are the focal length of the camera on the x and y axes

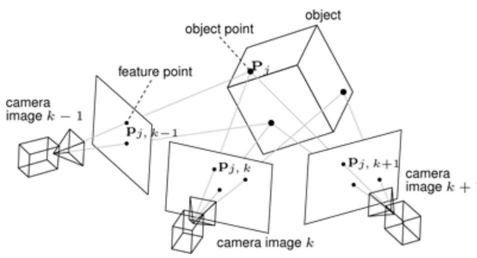
$$v = \frac{y \cdot f_y}{z} + c_y$$

cx, cy are the center of the camera's aperture

$$d = z \cdot s$$

s is the scaling factor of the depth map

Proposed Algorithm



Calculate camera's position changes

$$\Delta\theta = \omega \cdot \Delta t$$

$$\theta_C = \theta_L + \Delta\theta$$

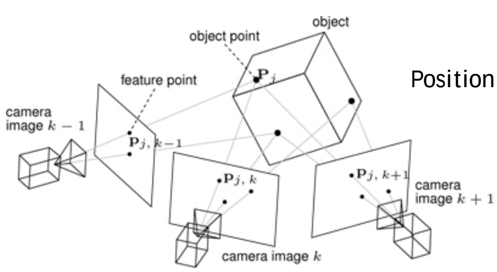
$$\Delta v = \mathrm{a} \cdot \Delta t$$

$$v_C = v_L + \Delta v$$

$$\Delta r = v_L + \frac{1}{2} \cdot \mathrm{a} \cdot \Delta t$$

$$r_C = r_L + \Delta r$$

Proposed Algorithm



Position prediction

Assume that (x, y, z) is the coordinate of the P_j point in the camera coordinate system at time k+1, and (x_1, y_1, z_1) is the coordinate of the P_j point in the camera coordinate system at time k, then there is the following correspondence:

$$[x, y, z]^T = R \times [x_1, y_1, z_1]^T + r$$

$$u = \frac{x \cdot f_x}{2} + c_x$$

$$v = \frac{y \cdot f_y}{z} + c_y$$

$$d = z \cdot s$$

Proposed Algorithm

Current Frame | Next Frame

Feature point1 → Feature point1'

Feature point2 → Feature point2'

Feature point3 → Feature point4'

Feature point4 → Feature point3'

Neighboring matching

Assume that the predicted coordinates of feature point 1 in the next frame image are (26, 39)

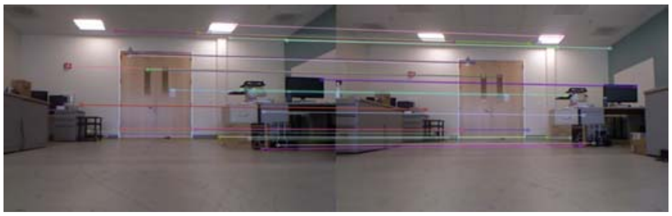
	Feature point1	Feature point2	Feature point3	Feature point4
Current Frame	(25, 40)	(100, 20)	(50, 70)	(40, 80)
Next Frame	(26, 38)	(99, 22)	(37, 79)	(49, 69)

Experiments and results

TurtleBot 2 robot used in the experiment

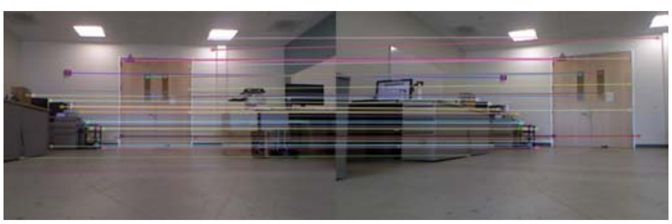
Experiments and results

Initial image and 1m forward image matching result

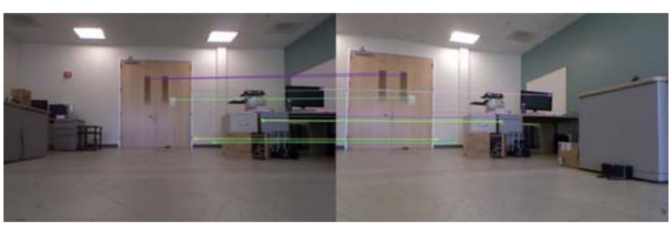


Initial image and back 1 meter image matching result

Experiments and results



Initial image and left turn 30 degree image matching result



Initial image and right turn 30 degree image matching result

Experiments and results

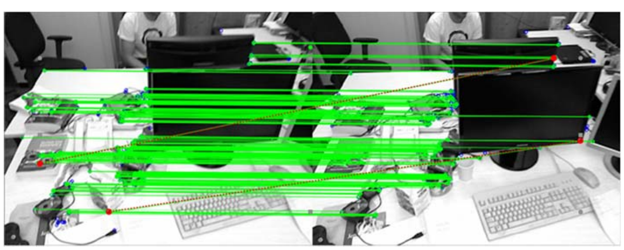
TABLE I. COMPARISON OF TIME CONSUMPTIONS

	SIFT	SURF	ORB	MonoSLAM	Our Algorithm
Front1m	6552.635	145.620	39.356	40.506	40.856
Back1m	8235.368	341.309	81.379	90.450	80.677
Left30	14523.506	495.597	104.860	102.756	100.042
Right30	7920.473	280.993	55.327	55.267	54.908
Average time	9307.996	315.88	70.231	72.245	69.121

Experiments and results

TABLE I. COMPARISON OF FALSE RATE

	SIFT	SURF	ORB	Mono-SLAM	Our algorithm
Average total number	298	180	82	50	25
Average false number	30	19	5	2	0
Average false rate	10.06%	10.56%	6.09%	4.00%	0%



Conclusion

- Result
  - This paper proposes a method that can reduce time consumption and error rate of feature points matching in visual SLAM system with a decrease of seeking feature points from whole image to surrounded area of a predicted position.
  - This paper puts forward a method of using camera's angle and displacement change to predict feature point's location in next frame. The prediction position can provide a reference coordinates for feature point matching process.
- Future
  - In RGB-D SLAM, implement Iterative Closest Point (ICP) algorithm is also time-consumed. The depth value of the RGB-D camera is sometimes invalid or uncertainty, which lead to the result of the ICP inaccurate.
  - Extending prediction algorithm to ICP algorithm for efficiency and accuracy improvement will be our next consideration and effort.

THANK YOU!