Improved Stereo Matching Algorithm Based on Adaptive Grid for Fixed-Wing UAVs

Hua Xia, Hongyuan Zheng(✉), and Xiangping Zhai

College of Computer Science and Technology/College of Artificial Intelligence,
Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
11122067@qq.com

Abstract. Modern unmanned aerial vehicles (UAVs) rely on binocular ranging modules to complete tasks such as obstacle avoidance, 3D reconstruction, and terminal strikes. However, the limited computing resources and high flight speed all put forward requirements for the real-time performance of the ranging algorithm. Thus, in this paper, we focus on how to make the algorithm dynamically allocate computing resources according to the changes of the UAV attitude to improve the system efficiency. To this end, we propose an improved semi-global matching method based on adaptive grid for fixed-wing UAVs. Experimental results demonstrate that the proposed method can effectively adapt to the changes in the speed and attitude of the UAV, and improve the real-time performance of the ranging module.

Keywords: UAV · Stereo matching · Adaptive grid

1 Introduction

The booming development of UAV technology makes it rapidly heat up in the civilian and military fields. More advanced flight controllers, vision modules, and energy systems promote the development of UAVs in a multi-purpose and intelligent direction such as transportation, inspection, reconnaissance, and terminal strikes.

Stereo matching technology has become the core of binocular vision algorithms because it can generate dense disparity maps, and has been widely used in 3D reconstruction, robot obstacle avoidance, and automatic driving [1]. Stereo matching algorithms are divided into local [2], global [3], semi-global [4,5] and deep learning-based algorithms [6]. The semi-global stereo matching algorithm is an improvement of the global algorithm. By transforming the two-dimensional image optimization problem into a one-dimensional optimization problem with multiple paths, the computational complexity is reduced while preserving the precision of the algorithm. Li et al. [7] used the census transform [8] to calculate the matching cost, and used the image pyramid to perform 8-path SGM [9] processing on each layer, and established a coarse-fine parallel stereo matching algorithm to reduce the matching time.

Q. Liang et al. (Eds.): Artificial Intelligence in China, LNEE 854, pp. 368–375, 2022.
https://doi.org/10.1007/978-981-16-9423-3_46

This paper proposes a stereo matching algorithm to solve the problem of the poor real-time performance of fixed-wing UAVs to obtain scene depth information. Specifically, we divide the image into several grids, and the stereo matching algorithm will adjust the calculation method according to the attitude of the UAV, and different calculation methods are assigned to different grids.

2 Problem Formulation and Preliminary Definitions

Classical stereo matching algorithms such as PatchMatch and SGM are not specifically optimized for UAV platforms. For example, the real-time changes in the attitude of the UAV are not reflected in the image processing stage. This paper is interested in making the UAV maintain a high-frequency perception of obstacles when its attitude changes. Some assumptions are used to simplify the problem:

- The obstacle is stationary.
- The gimbal is mounted on a fixed-wing UAV.

According to the kinematic constraints on fixed-wing UAVs, we introduce three influencing factors, namely forward velocity $v(t)$, yaw rate $y(t)$, and pitch rate $p(t)$. We are interested in finding a strategy to make the UAV subject to the constraints of these three factors during image processing and produce the corresponding output. At the same time, we divide the image into nine grids, as shown in Fig. 1, each of which represents a receptive field. We define the system as the following states: locked state LS, forward state FS, yaw state YS, pitch state PS. The state of the system is derived from the following equation:

$$state(t) = f(v(t), y(t), p(t)) = \begin{cases} LS, & v(t) = 0 \\ FS, & v(t) > 0 \\ YS, & v(t) > 0, y(t)/Y_{max} > \sigma_1 \\ PS, & v(t) > 0, p(t)/P_{max} > \sigma_2 \end{cases} \tag{1}$$

where Y_{max} represents the maximum yaw rate, P_{max} represents the maximum pitch rate, σ_1 and σ_2 are hyperparameters and need to be adjusted according to the parameters of the UAV flight controller.

When the system is in the FS state, the computing resources are concentrated in the 5th receptive field. In the YS state, the computing resources are concentrated in the $\{1, 4, 7\}$ or $\{3, 6, 9\}$ receptive field, which depends on the yaw direction. In the PS state, the computing resources are concentrated in the $\{1, 2, 3\}$ or $\{7, 8, 9\}$ receptive field. The system can be in multiple states at the same time.

3 Improved Semi-global Matching Based on Adaptive Grid

The stereo matching algorithm is generally divided into 4 steps: cost initialization, cost aggregation, disparity selection, and disparity optimization. The

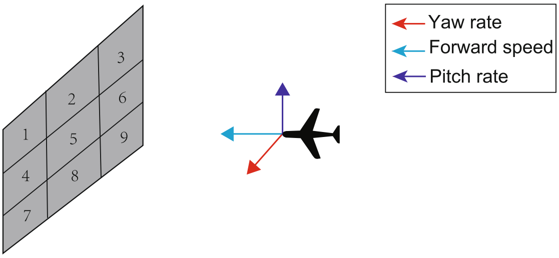


Fig. 1. The image is divided according to the receptive field of the UAV, and the forward speed, yaw rate, and pitch rate determine the system state.

cost initialization of semi-global matching adopts the mutual information algorithm but requires a relatively large amount of calculation. We use a simpler Census transform to replace this method. At the same time, we added a module before cost aggregation to dynamically calculate the grid size. This module reads data from the UAV's sensors and provides grid width information to the cost aggregation stage before each frame. The cost aggregation stage is also carefully designed, multi-path cost aggregation is used to approximate global matching. The algorithm flow chart is shown as in Fig. 2.

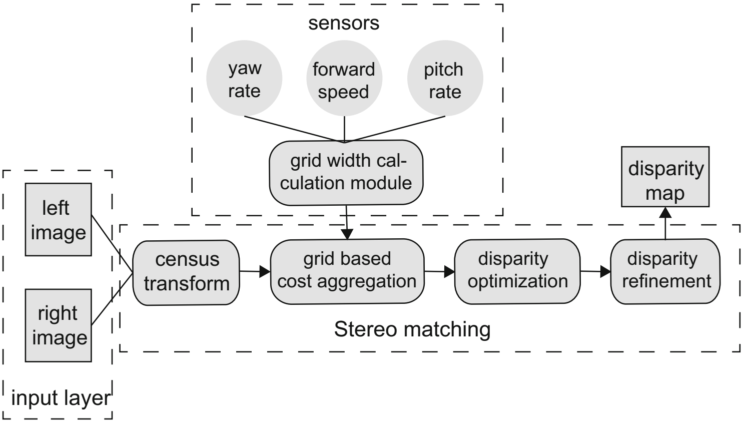


Fig. 2. Summary of processing steps of Sect. 3.1, 3.2, 3.3, and 3.4.

3.1 Cost Initialization

Census transform compares the pixel gray value in the center pixel domain window (the window size is $n*m$, n and m are odd) with the gray value in the center of the window, and maps the boolean value obtained from the comparison to a bit string. Finally, the bit string value is used as the Census transform value C_s of the center pixel, as shown in the following equation:

$$C_s := \bigotimes_{i=-n}^{n} \bigotimes_{j=-m}^{m} \xi(I(u,v), I(u+i, v+j)) \tag{2}$$

where u, v are pixel coordinates, and ξ operation is defined by

$$\xi(x,y) = \begin{cases} 0, x \leq y \\ 1, x \geq y \end{cases} \tag{3}$$

The matching cost calculation method based on census transform is to calculate the Hamming distance of the census transform value of the two pixels corresponding to the left and right images, that is, the number of different corresponding bits of the two-bit strings. The calculation method is to carry out NOR operation on two-bit strings, and then count the number of bits that are not equal to 1 in the result of NOR operation as the initial matching cost. The equation is as follows:

$$C(u,v,d) := Hamming(C_{sl}(u,v), C_{sr}(u-d,v)) \tag{4}$$

where $C(u,v,d)$ is the cost value of pixels $(u,\ v)$ under disparity d.

3.2 Adaptive Grid Size

The fixed grid size cannot adapt to the speed changes of the aircraft. We hope that the UAV's attitude changes can be reflected in the algorithm. We have designed a module specifically for this purpose. This module obtains data from the plane's sensors and aims to provide width and height information of the grid for subsequent calculations. We calculated the values of the three factors introduced in Sect. 2 based on the UAV's accelerometer, magnetometer, gyroscope, and equipped GPS module, and determined the system state, and finally calculated the size of the grid.

When the system is in the FS, YS, or PS state, the width w and height h of the 5th grid are adaptive and calculated by the following equation:

$$\begin{aligned} w &= W - 2v(t)/V_{max} * W \\ h &= H - 2v(t)/V_{max} * H \end{aligned} \tag{5}$$

where W and H represent the width and height of the input image, V_{max} represents the maximum forward speed. Once the system state and grid size are determined, we will use this information in the cost aggregation stage to improve the real-time performance of the algorithm.

3.3 Grid Based Cost Aggregation

The cost of the pixel p is aggregated from the multi-directional path cost, such as 16-path cost aggregation, 8-path cost aggregation, and 4-path cost aggregation. The calculation method of the path cost of pixel p along a certain path r is as follows:

$$L_r(p,d) = C(p,d) + \min \left\{ \begin{array}{c} L_r(p-r,d) \\ L_r(p-r,d-1) + P_1 \\ L_r(p-r,d+1) + p_1 \\ \min\limits_i L_r(p-r,i) + P_2 \end{array} \right\} - \min_i L_r(p-r,i) \qquad (6)$$

Among them, the first term is the matching cost value C, the second term is the smoothing term, the third item is to ensure that the new path cost value L_r does not exceed the upper limit of the value. The value accumulated to the path cost takes the least cost value in the three cases of no penalty, P_1 penalty, and P_2 penalty. P_1 penalty is less severe and punishes the situation where the disparity change of adjacent pixels is very small (1 pixel). P_2 penalty is more severe, which punishes the situation where the disparity of adjacent pixels varies greatly (greater than 1 pixel). To protect the discontinuity of disparity in the real scene, P_2 is often dynamically adjusted according to the gray difference of adjacent pixels, as shown below:

$$P_2 = \frac{P_2'}{|I_{bp} - I_{bq}|}, P_2 > P_1 \qquad (7)$$

where the P_2' is the initial value of the P_2 and is generally set to a value far greater than that of the P_1, I_{bp} and I_{bq} represents pixel intensity.

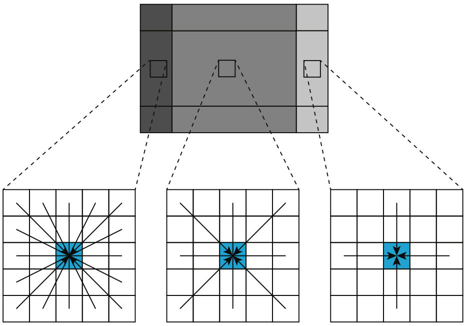


Fig. 3. When the UAV yaws to the left, the left grid adopts 16-path cost aggregation, and the calculations consumed by the middle and right grids only account for 1/2 and 1/4 of the left.

We decide to use several path cost aggregations based on the system state and the location of the pixel. As shown in Fig. 3, when the aircraft is in the

YS state and yaws to the left, the $\{1, 4, 7\}$ receptive field uses 16-path cost aggregation to obtain more accurate depth information, while $\{3, 6, 9\}$ uses 4-path cost aggregation, which makes the calculation amount of the right area only 1/4 of the left. Our method not only ensures the flight safety of the UAV but also improves the real-time performance of the algorithm due to the reduction of the overall calculation amount.

3.4 Disparity Optimization and Disparity Refinement

The disparity optimization step uses the winner-take-all method to calculate the optimal disparity map. WTA selects the disparity corresponding to the smallest cost value as the final disparity according to the DSI (Disparity Space Image) generated after cost aggregation.

We use a variety of methods to refine the disparity. First, the left-right consistency method, the elimination of small connected regions, and the uniqueness detection are used to reduce the mismatch rate, then the quadratic curve fitting is used to calculate the sub-pixel disparity, and finally, the 3×3 median filter algorithm is used to suppress the noise.

4 Experiments

We evaluated our method on Middlebury [10] datasets. We also used the AirSim [11] simulation environment to evaluate the influence of different system states and different flight speeds on the performance of the algorithm.

4.1 Datasets and AirSim Simulation Environment

Since the Middlebury dataset is not a picture taken by a UAV in a real scene, we have pre-defined the parameters of the aircraft. The maximum forward speed of the UAV supports 20 m/s, the maximum yaw rate is 10 rev/s, and the maximum pitch rate is set to 8 rev/s. Select Djembe, Hoops, Piano, and Teddy as the test pictures. AirSim is a cross-platform open-source simulator for drones and other autonomous mobile devices built on Unreal Engine. We use the built-in virtual binocular camera of AirSim to test the performance of the algorithm. The size of the image generated by AirSim is $256 * 144$, and the test scene is CityPark.

4.2 Experimental Results and Analysis

According to Fig. 4, when the system is in the FS state, as the speed of the UAV increases, the size of the 5th grid decreases, the proportion of 16-path cost aggregation decreases, resulting in a decrease in the running time of the algorithm. Our improved method is approximately 25% faster than the SGM algorithm at maximum speed. Table 1 shows the processing time for Teddy pictures when the system is in different states.

AirSim's experimental results show that when the UAV is in the YS state, the left grid adopts 16-path cost aggregation, and the objects in the red box show clearer disparity, as shown in Fig. 5.

Table 1. The running time of our method on Teddy image. We fixed the forward speed to obtain the comparison result.

State	Speed (m/s)	Time (ms)
FS	10	663
YS	10	788
PS	10	764

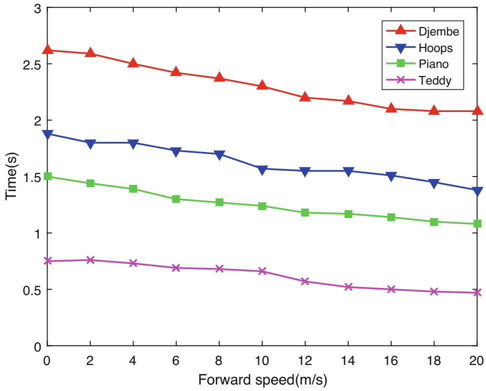
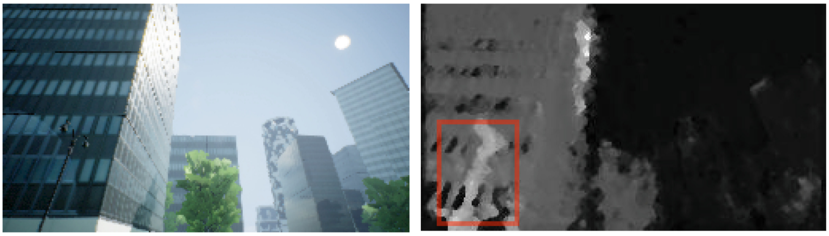


Fig. 4. The test results of the Middlebury dataset show that the running time of the algorithm can be reduced by up to about 25% as the forward speed increases.



(a) (b)

Fig. 5. The disparity map generated in AirSim. (a) Reference image; (b) Disparity map. Our method can well adapt to the attitude change of UAVs. The object in the red box shows high definition when the system is in the YS state.

5 Conclusions

We propose an improved stereo matching algorithm for fixed-wing UAVs. This method is based on the adaptive grid, which can dynamically allocate computing resources to different grids according to the attitude changes of UAV. Our experimental results on the Middlebury dataset show that this method is faster than the SGM algorithm at maximum speed. The experimental results in AirSim show that our method has good robustness in a simulated environment.

Acknowledgements. This work was supported by National Defense Science and Technology Innovation Zone Foundation under Grant No. 19-163-16-ZD-022-001-01.

References

1. Zhao, C., Li, W., Zhang, Q.: Research and development of binocular stereo matching algorithm. J. Front. Comput. Sci. Technol. **14**(7), 1104–1113 (2020)
2. Zeglazi, O., Rziza, M., Amine, A., Demonceaux, C.: A hierarchical stereo matching algorithm based on adaptive support region aggregation method. Pattern Recogn. Lett. **112**, 205–211 (2018)
3. Ma, N., Men, Y., Men, C., Li, X.: Segmentation-based stereo matching using combinatorial similarity measurement and adaptive support region. Optik **137**, 124–134 (2017)
4. Chai, Y., Yang, F.: Semi-global stereo matching algorithm based on minimum spanning tree. In: 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 2181–2185. IEEE (2018)
5. Hernandez-Juarez, D., Chacón, A., Espinosa, A., Vázquez, D., Moure, J.C., López, A.M.: Embedded real-time stereo estimation via semi-global matching on the GPU. Procedia Comput. Sci. **80**, 143–153 (2016)
6. Zbontar, J., LeCun, Y., et al.: Stereo matching by training a convolutional neural network to compare image patches. J. Mach. Learn. Res. **17**(1), 2287–2318 (2016)
7. Li, Y., Zheng, S., Wang, X., Ma, H.: An efficient photogrammetric stereo matching method for high-resolution images. Comput. Geosci. **97**, 58–66 (2016)
8. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 151–158. Springer, Heidelberg (1994). https://doi.org/10.1007/BFb0028345
9. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 328–341 (2007)
10. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision **47**, 7–42 (2002). https://doi.org/10.1023/A:1014573219977
11. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: high-fidelity visual and physical simulation for autonomous vehicles. In: Hutter, M., Siegwart, R. (eds.) Field and Service Robotics. SPAR, vol. 5, pp. 621–635. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67361-5_40