Proceedings of the 2003 IEEE
International Conference on Robotics,Intelligent Systems and Signal Processing
Changsha, China - October 2003

A Parallel Algorithm for Visual Tracking of Multiple Free-swimming Robot Fishes Based on Color Information

Junzhi Yu, Shuo Wang, Min Tan
Institute of Automation Chinese Academy of Sciences, Beijing, China, 100080
E-mail: yujunzhi@263.sina.com or jzyu@compsys.ia.ac.cn

Abstract

Obtaining information of both action of multiple free-swimming robot fishes and environment quickly and accurately from image sequence is the foundation of making decision and control in a MRFS (Multiple Robot Fishes cooperation System). The real-time [1] vision subsystem of MRFS and its tracking strategy for multi-target are introduced in this paper. Combining features of the robot fish and location background, an adaptive segmentation algorithm based on hue histogram and saturation histogram is proposed, which can adapt the changing environmental conditions. At the same time, integrating with computer parallel processing technology, the overall tracking algorithms are optimized by means of MMX and SSE instructions. The designed visual subsystem has been implemented in MRFS, and the results have shown its effectiveness by successfully tracking multiple free-swimming fishes and obstacles.

1 Introduction

With ceaseless advancements on biomimetism, underwater fish-like propulsion has become an important research in Autonomous Underwater Vehicle (AUV) domain over the last few years. Development of robot employing this technology, namely robot fish, provides new ideas for designing and making autonomous underwater vehicles with high efficiency, good maneuverability and stealthy capability [1-3]. The prospective applications of robot fish are in military detection, underwater operation, oceanic supervision, aquatic life-form observation, pollution search and so on. However, the capability of a single robot fish is limited. Thus, it will be incompetent for accomplishing missions in unconstructed and dynamic environment. In such cases, hence, it may be a feasible approach to select a cooperative multi-agent robotic system with high maneuverability and excellent tolerance. The establishment of *MRFS* based on the former application cases, will be convenient for research on single and multiple robot fishes' control.

In *MRFS*, vision subsystem is crucial for free-swimming fishes to sense environment. The image captured by CCD camera is effectively processed and analyzed, and then the useful information about the robot fishes' action and surrounding is extracted as an input of decision-making. In general, a high-speed image processing system is required for accurate recognition of multi-target. An important first step is to classify each pixel in a sampling image into one of a discrete number of color classes. The basic approaches to accomplishing this task include linear color thresholding, nearest neighbour classification, color space thresholding and probabilistic methods [4]. The inability to adapt the image segmentation process to real-world changes is one of the fundamental weaknesses of constant thresholding in a projected color space. To achieve robust performance, therefore, it is necessary to add the variant of environmental conditions to constant thresholding so that closed-loop segmentation is working. As a rule, the multi-target tracker at real-time rates is implemented in hardware (*e.g*, DSP and FPGA), but is potentially more expensive and complex than software-only approaches on general purposes hardware. The approach to be taken in *MRFS* is a combination of parallel processing technology in common PC, but with a special focus on efficiency issues. Our objective is to develop an efficient and practical real-time multi-target tracker.

The paper is organized as follows: A brief description of vision subsystem of *MRFS* is introduced in Section 2. An adaptive segmentation algorithm for specified color based on parameters H and S with HLS (Hue, Lightness, Saturation) color space is presented in Section 3. A parallel processing model is proposed and applied to the former identifying algorithms in Section 4. Finally, corresponding experimental results are given.

2 Scheme and Implementation of Vision Subsystem in *MRFS*

As shown in figure 1, *MRFS* is composed of four subsystems: robot fish subsystem, vision subsystem, decision-making and emulation subsystem and

[1] We define "real-time" as full frame processing at 25Hz (for PAL mode), 30Hz(for NTSC mode) or faster.

communication subsystem. A free-swimming, 4-link robot fish was developed by the Laboratory of Complex Systems and Intelligence Science in the Chinese Academy of Sciences and the Robotic Institute in Beijing University of Aeronautics and Astronautics, which is 375mm in length, 75mm in width and 40mm in thickness. The information of fishes and their surrounding captured by overhead CCD camera is effectively processed and sent to the decision-making module as an input, and then the output of the decision-making and emulation subsystem is transmitted to the single robot fish through the communication subsystem. So, the robot fishes work together to implement some tasks. In the *MRFS*, rapid and accurate visual tracking algorithm is important for global implement efficiency and precision. During a robot fish's control cycle, the faster the image identifies, the ampler time remains for decision-making in one control cycle, and the better robot fish's real-time decision may be.

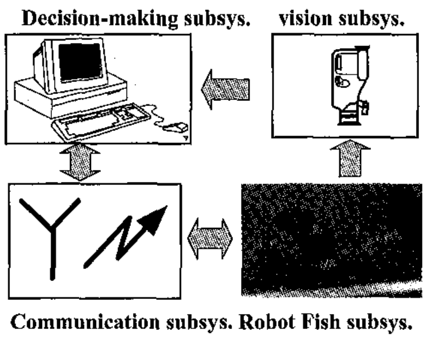


Fig.1 Architecture of MRFS

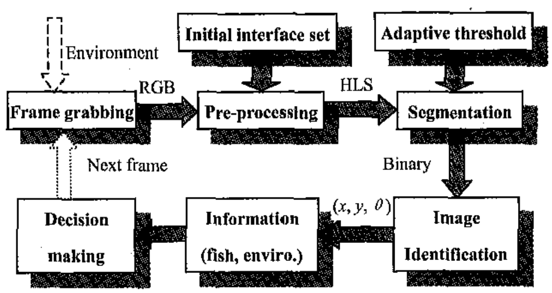


Fig. 2 Operation process of vision subsystem of MRFS

In *MRFS*, a CCD camera hanged over swimming pond acts as sensor that is in charge of capturing robot fish's moving and surrounding information. And the vision subsystem is primarily used to track many targets such as position and orientation of swimming fishes, ball (used for fish's playing ball) and obstacles. So, it is responsible for identifying obstacles in surroundings and tracking robot fish's swimming. Speediness and accuracy are its basic requirements. In the meantime, the anti-jamming measure avoiding the irregular effect of light must be taken in the course of systematic design and implementation.

The scheme of operation is described in block diagram (Fig. 2). The overall operation consists of six parts, which include frame grabbing, pre-processing, initial interface setting, adaptive segmentation, identifaction and information output. For each object is equipped with different color, an efficient color space thresholding is utilized. First, a color CCD camera takes a 24-bit image through sampling RGB signal. As is known, RGB is a representation of color which is not suitable for color visual applications, so RGB color space is converted to HLS color space in pre-processing module. After adaptive segmentation, the position and orientation of every object is then extracted employing the enhanced image. At the end of every processing cycle, state vectors which keep the information about the position and orientation are obtained and sent to decision-making and emulation subsystem.

3 Image Identification Algorithm Based on Color Information

In order to verify the feasibility and reliability of proposed algorithms in complex and dynamic environment, a variety of experiments such as playing ball and going through channel on free-swimming robot fishes were carried out. In a pond with 2000mm×1150mm, a floating ball (diameter 80mm) is regarded as a target, and two robot fishes swam to ball and pushed it to opponents' door respectively. As for experiment of going through the channel, multiple robot fishes swim through 80mm-wide channel in the specified order or sequence. In these cases, accordingly, the most important work for vision subsystem is to track robot fishes and identify their surroundings. How to design a fast and efficient identifying algorithm is the key to vision subsystem's implementation.

3.1 Adaptive Thresholding for Segmentation

Color information is the foundation of identification. In *MRFS*, ball, obstacles and robot fishes are equipped with specified colors. Considering surrounding background and characteristic of the robot fish, we use color tokens in multi-target tracking. As shown in Fig.3, each robot fish's color index consists of two rectangular bands: ID color and orientation color. The two color tokens are rectangle whose lognitudinal side is longer than transverse one along the fish body's axis, and ID color's lognitudinal side is larger than Orientation color's lognitudinal one. For diffferent robot fishes, Orientation color may be identical, while ID color must be different. In order to enhance identifying speed of system, the same Orientation color is applied to all fishes.

Fig.3 Color token of robot fish

Because R, G and B of image data are scatted and related, they are not suitable for color visual applications. Taking advantage of robustness of combining saturation and hue properties of the object, a normalized color space HLS is selected during segmentation [4-7]. In this color space, the lightness and the saturation values range from 0.0 to 1.0 inclusive, and the hue varies from 0° to 360° inclusive. To utilize efficiently histogram analysis in image operations, three components of HLS is normalized to range from 0 to 255. That is,

$[H\ L\ S]_{Norm} = [H'\ L'\ S'] = [H\times255/359\ \ L\times255\ \ S\times255]$.

Thus, three histograms for the hue, the saturation and the lightness respectively can be obtained from a color image after HLS transformation. Because very few objects have the same hue and saturation, moreover, experiments on the intensity property showed that it is very susceptible to illumination conditions, only the hue (H) and the saturation (S) are chose as the criteria of binarization [7]. Here, the value of threshold for H and S are determined by hue histogram and saturation histogram. The checked pixel that satisfies constraint condition (1) is regarded as goal point in binarization.

$$\begin{cases} H_{lowerthresh} < H < H_{upperthresh} \\ S_{lowerthresh} < S < S_{upperthresh} \end{cases} \qquad (1)$$

For the changing environmental conditions affect the appearance of an image the fixed threshold will malfunction. Also ripples in surface of water disturbed by free-swimming fish reflect light and lead to continuous variation of background. To achieve robust performance in real application, an adaptive thresholding has to be adopted.

Firstly, for every frame of image some fixed characteristic points (*e.g.*, N points scattering four corners of the scenario) reflecting background changes are sampled in HLS value: $PHLS_i=[H_i\ L_i\ S_i]$, $i=1,2,\ldots,N$. Consequently, the mean μ and the "sample" standard deviation σ for H and S of characteristic points are estimated. Then, by comparing σ with σ_{ideal}, a Boolean value is return to $B_{HLS}(t)$: if $\sigma \geqslant \sigma_{ideal}$, $B_{HLS}(t)$ returns TRUE; Otherwise, $B_{HLS}(t)$ returns FALSE; If $B_{HLS}(t)$ is TRUE, the variants of background: ΔH and ΔS are calculated and added to threshold respectively. Here, $\Delta H=\mu_H-H_{ideal}$, $\Delta S=\mu_S-S_{ideal}$, where, H_{ideal} and S_{ideal} are predefined values for background color got from histogram analysis. If $B_{HLS}(t)$ is FALSE, the thresholds keep same. Thereby a closed-loop thresholding is created in each segmentation cycle.

3.2 Smooth Processing

To eliminate spurious effects in digitization and transportation, the binary image after segmentation should be smoothed. An 8-point neighborhood averaging is used in actual implementation. Usually, most common enhancement filters will smooth the interior of regions at the cost of the edges. To lessen this effect, a threshold method is adopted to yield smoothing image. It assumes that $f(x, y)$ is initial image with $N\times N$ pixels, and $g(x, y)$ is processed image. Then $g(x, y)$ is determined by expression (2) [4]:

$$g(x,y)=\begin{cases} \dfrac{1}{M}\sum\limits_{(m,n)\in S} f(m,n) & \left|f(x,y)-\dfrac{1}{M}\sum\limits_{(m,n)\in S} f(m,n)\right| > T \\ f(x,y) & Otherse \end{cases} \qquad (2)$$

Where, $x, y=0,1,2,\ldots,N-1$, S is the set of neighboring points (of course, point (x, y) is excluded), M is the number of points in neighborhood area (here M=8), T is the specified threshold. In fact, for the input image is a binary image, the neighborhood averaging can be reduced to judge the value of $\Sigma f(m, n)$. That is to say, if $\Sigma f(m, n)>4$, the value of $g(x, y)$ is replaced with 1 (stands for the goal point), or otherwise, $g(x, y)$ keeps 0(stands for the background).

3.3 Feature Extraction of Tracking Objects

3.3.1 Centroid Identification of Objects

Once smoothing processing has been performed, the centroid of object can be calculated by searching the global image by means of moving window scanning. In implementation, a 24×24 mask window W is used to search maximum-goal-point window (W_{max}) in which the number of goal point is maximum. The position of window and the maximum-goal-point are recorded during the global search. If the final maximum-goal-point is larger than some threshold, tracking object is regarded as existent. Otherwise, tracking object is noise. On condition that tracking object is validated, we will define region R ($R \in W_{max}$ and $g(x, y)$=1)as the set of pixels which are part of the object. For the region R, centroids for x-coordinate and y-coordinate will be computed respectively, the equation is shown in formula (3).

$$x_c(R) = \frac{\sum_{(x,y)\in R} x}{N(R)}, \quad y_c(R) = \frac{\sum_{(x,y)\in R} y}{N(R)} \tag{3}$$

Where, $N(R)$ stands for the number of pixels in the region R.

Consequently, the centroid of the tracking object can be expressed as $C(R)=(x_c(R), y_c(R))$. By calculating its centroid, the position of the tracking object is identified.

3.3.2 Identification of Orientation

In *MRFS*, there are multiple free-swimming fishes to be tracked. For a single fish, a simple approach to identifying its orientation is calculating its slope between ID color and orientation color. However, two different colors are needed for identifying one fish. In such case, when there are several ones to be tracked at one time, the real-time control will be heavily destroyed due to enormous operation burden. Additionally, it is difficult to choose different colors for multiple fishes. So, an identifying algorithm for fish's orientation with ID color must be designed, while orientation color only serves as an aid. A detailed algorithm integrating geometric features of color token is stated as follows.

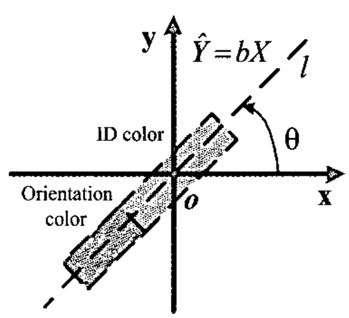


Fig.4 Calculation schematic of orientation of robot fish

As shown in Fig.4, for a box region R (ID color region dotted in yellow) composed of N discrete data points $P_i(x_i,y_i)$ ($0 \leqslant i \leqslant N$-1) in 2-D Cartesian coordinates plane xoy, a linear regression equation $\hat{Y} = a + bX$ can be created to describe distribution of multi-data. To facilitate description and inference, coordinate translation is adopted to make origin corresponds to the centroid of region $C(R)= (\frac{1}{N}\sum_{i=0}^{N-1} x_i, \frac{1}{N}\sum_{i=0}^{N-1} y_i)$. Let the perpendicular distance from point P_i to straight-line l be d_i. If the evaluative function to be minimized is chosen as $\sum_{i=0}^{N-1} d_i^2$, then a fitting straight line through a set of points will be determined. Since the straight line in fish's motion direction is through its centroid, moreover the centroid is current origin, a simple inference is a=0. After a fair bit of algebra, the result of slope b is:

$$b = \frac{\sum y_i'^2 - \sum x_i'^2 \pm \sqrt{(\sum x_i'^2)^2 + (\sum y_i'^2)^2 - 2\sum x_i'^2 \sum y_i'^2 + 4(\sum x_i' y_i')^2}}{2\sum x_i' y_i'} \tag{4}$$

Where, $\sum \cdot$ is short for $\sum_{i=0}^{N-1} \cdot$, x_i', y_i' are the values after the coordinates translation corresponding to x_i, y_i respectively. The detailed transformation is shown in formula (5). Of course, the above results can be extended into more general circumstances by means of reverse coordinates translation.

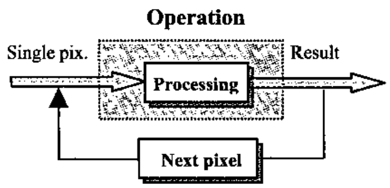
$$\begin{cases} x_i' = x_i - \frac{1}{N}\sum_{i=0}^{N-1} x_i \\ y_i' = y_i - \frac{1}{N}\sum_{i=0}^{N-1} y_i \end{cases} \tag{5}$$

By calculating according to formula (4), two different resolutions: b_1 and b_2 are got. In reality, only the value of b with minimal $\sum d_i^2$ is selected. Hence the orientation angle θ can be obtained using inverse tangent function. In addition, with the aid of ID color which is alignment with Orientation color, the orientation angle can be mapped into desirable intervals, like $[-\pi, \pi]$ and $[0, 2\pi]$.

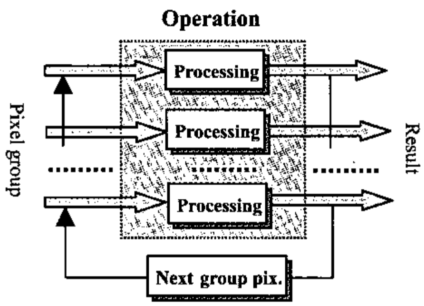
4 Parallel optimization of algorithms

In the above visual tracking, a variety of operations such as filtering, segmentation and feature extraction are performed. In such operations, the same processing rule usually applies large numbers of image data. Thus, there is a great need for parallel processing that can be easily and effectively used on a mass of data. Integrating inherent parallelism of image processing with characteristics of processing unit's length in PC, a parallel processing method is adopted in our implementation.

Considering independence between pixel and pixel in time and space, we maybe operate multiple pixels at one time. That is to say, a group of organized pixels are sent to operation unit. As Fig.5 shows, for normal operation, one pixel is handled at a time, whereas multiple pixels can be handled for parallel operation at a time. Apparently, with parallel operation better performance will be obtained on multi-data block processing.



(a) Normal operation



(b) Parallel operation

Fig.5 Schematic comparison between normal operation and parallel operation

Fortunately by exploiting the features added by the MMX/SSE technology, image processing can be parallelized conveniently. MMX/SSE technology is designed to accelerate multimedia and communications applications, yet maintain full compatibility with existing operating systems and applications. Its key is called Single Instruction Multiple Data (SIMD), meaning that an instruction can perform the same function on multiple pieces of data. In detail, a MMX instruction can perform 8, 4 and 2 data units with 8-bit, 16-bit and 32-bit integral type respectively, and a SSE instruction can perform 4 data units with 32-bit single-precision floating-point type [8, 9]. As a result, by taking advantage of parallelism of MMX/SSE instructions in data processing, the speed of image processing will double and redouble.

By creating corresponding data construction for various processing operators based on former algorithmic framework, primary modules of visual tracking are coded in MMX/SSE instructions. We use MMX instructions to optimize segmentation module and smoothing processing module, and SSE instructions for HLS transformation, centroid and orientation calculation. In consequence, the efficiency of vision tracking improves remarkably.

5 Experimental results

All the components of hardware and software for visual tracking are based on PC. All these are executed on a Pentium 4 PC with WINDOWS98, Visual C++ 6.0 and Service Pack5. It takes about 15 milliseconds to identify two swimming fishes and two obstacles in a 320×240-resolution image (24 bit). A real scenario and corresponding experimental data are given in Fig.6 and Table 1 respectively, and positional error keeps within 5%.

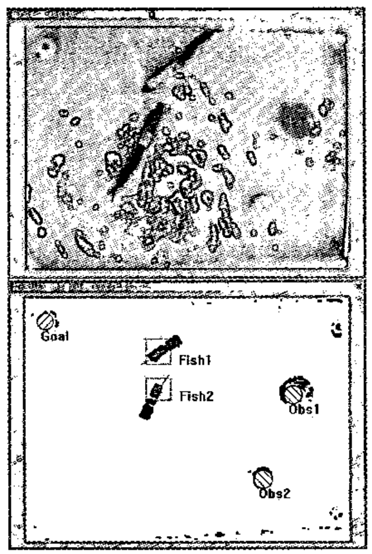


Fig.6 A real experimental scenario and results of tracking

Table 1. Testing results of tracking algorithm (Debug version, for 100 times)

Identifying parts	MMX/SSE average time (ms)
HIS transformation	5.849084
Adaptive thresholding	0.792829
8-point smoothing	0.454818
Features extraction	0.400618

When frame grabber works in PAL mode (capture rate is 25 frame/second), the interval of two continuous frames is 40 milliseconds, hence CPU has enough time to implement the decision making such as path-planning, obstacle avoidance, formation control and so on. Seen from the table 1, the most time-consuming operation is HLS transformation, however this transformation coded in C Language costs 28.551669ms. Apparently, the speedup ratio of MMX/SSE is up to 4.88. Therefore, the proposed tracking algorithm optimized by MMX/SSE can fully satisfy operation requirements of *MRFS*.

6 Conclusions

In this paper, a real-time visual tracking algorithm for *MRFS* is discussed. We give a description of operation process for vision subsystem, and propose an adaptive segmentation method based on color information. Integrating tracking algorithms with parallel processing technology based on PC, the presented algorithms are optimized with MMX/SSE instructions. As experimental results show, recognitions of free-swimming fish and obstacles are fast and accurate, which validate the feasibility of the algorithms. Perhaps, more advanced method for adaptive segmentation and fast feature extraction is required to improve the stability and robustness of visual tracking in the changing conditions. However, the methods integrating image processing with parallel processing technology are general in some circumstance, and provides valuable references for real-time vision's application.

Acknowledgement

This work is funded by research grants from the robotics subject of 863 Program (Grant No. 2001AA422370).

References

[1] Michael Sfakiotakis, David M. Lane, and J.Bruce C.Davies., "Review of Fish Swimming Modes for Aquatic Locomotion," *IEEE Journal of Oceanic Engineering*, pp.237-252, 1999.

[2] M.S. Triantafyllou and G.S. Triantafyllou, "An Efficient Swimming Machine," *Scientific American*, pp.64-70, 1995.

[3] J.M. Anderson and P.A. Kerrebrock, "The Vorticity Control Unmanned Undersea Vehicle (VCUUV)-An autonomous vehicle employing fish swimming propulsion and maneuvering," in *proc.10th Int. Symp. Unmanned Untethered Submersible Technology*, NH, Sept., 1997, pp.189-195.

[4] Ruan qiuqi, "*Digital Image Processing*," Beijing: publishing house of electronics industry, 2001 (in Chinese).

[5] R.Garcia, J.Batle, X.Cufi, and J.Amat, "Positioning an Underwater Vehicle through Image Mosaicking," *ICRA*, 2001, pp.2779-2784.

[6] Zhu shulong and Zhang zhanmu, "*Obtaining and analysis of remote sensing image*," Beijing: publishing house of science, 2000 (in Chinese).

[7] Rafael Garcia-Campos, Joan Batlle, and Bainer Bischoff, "Architecture of an Object-based Tracking System using Colour Segmentation," information available electronically at

http://www.unibw-muenchen.de/campus/LRT6/ps/iwsip96.pdf

[8] Qian xiaojie, "*Design of assembly program*," Beijing: publishing house of electronics industry, 2000 (in Chinese).

[9] Ping-Che Chen, "Introduction of SSE," information available electronically at

http://smart.iis.sinica.edu.tw/~pcchen/hive/index.html