Research on Dynamic Integration of Multi-objective Data in UI Color Interface

Ling-wei Zhu and Feng Zhai(✉)

Xi'an Eurasia University, Xi'an 710065, China
{zhulingwei541,zhaifeng5431}@sina.com

Abstract. On the traditional method of dynamic integration of multi-objective data in UI color interface, because of the single integration algorithm, it is easy to lose the target data when there is too much target data. Therefore, based on the use characteristics of UI color interface, a new integration method of multi-objective data is proposed. This method obtains the sampling target through deep web data, detects and tracks the target image, optimizes according to the multi-objective integration, realizes the optimal path multi-objective equilibrium integration. Experimental results: the proposed detection method is fully in place in data integration, the occupancy rate of arm is 0%, the load line of DSP is 20%, the system maintains reliable real-time, and achieves the ideal state of UI color interface operation. However, the traditional data integration method of SLR is not in place; it can be seen that the traditional integration method is not suitable for the requirements of UI color interface with large target data.

Keywords: Artificial intelligence · Network public opinion · Data abnormal behavior · Detection method

1 Introduction

With the development and evolution of multimedia technology in the digital era, UI interface design, as a means and way to enhance brand image, product value and human-computer interaction experience, gradually makes people feel the important role of interface design. In recent years, UI interface design has become a design research direction that both design field and computer field pay attention to [1]. In UI design, communication and communication are the essence of user interface, while in visual communication, 83% of human's reception of external stimuli is the function of visual media. UI interface is all the information sources that users touch when they use the software. Among these visual factors, color can form the effect of clarity, contrast and reflection, which has an impact on human visual system, can make people obtain a kind of impact in the first time, can also emphasize and convey information, as well as express the feelings and emotions of things. How to apply and organize these data has become the key of UI color interface processing.

This paper takes it as the research topic to analyze the dynamic integration of multi-objective data. By finding the candidate page of the data source, the query interface is obtained in the page; the USB camera of the ZC0301 microchip is used to collect images, and the moving targets and new targets newly appearing in the scene are

S. Liu and L. Xia (Eds.): ADHIP 2020, LNICST 348, pp. 245–254, 2021.
https://doi.org/10.1007/978-3-030-67874-6_23

detected and extracted from the video image sequence. Detect and extract; and use Mean-Shift algorithm to track the target and conduct experimental analysis.

2 Dynamic Integration Method of Multi-objective Data in UI Color Interface

2.1 Deep Web Data Acquisition Sampling

Data source discovery refers to the discovery of accessible databases in resources, which is generally divided into two steps: finding candidate data source pages and finding their query interfaces from the obtained pages [2]. The solution to the first step is to transform the technology that has focused on crawling. The goal of focused crawling is to identify those hyperlinks that are more likely to reach the target web page in the process of crawling, so that only part of the web pages that are closely related to the search topic can be crawled and the resource yield can be improved. A common focused crawler is usually composed of three parts: Web collector, web classifier and crawling queue. The key problems in focused crawling are the measurement of the correlation between the obtained web page and the target topic, and the ranking of crawling priority queues. Aiming at the former, this paper mainly studies the topic classification algorithm of web pages. For the latter, there are many different crawling scheduling strategies. One kind of method calculates the link value according to the link relationship between pages, such as algorithm, class algorithm, etc [3]. Another method, such as an algorithm, evaluates links according to the correlation between link information and subject area. The feedback information is used to train the classifier online and incrementally, so as to adjust the priority of the links in the candidate queue.

The second step is to find the query interface from the obtained home page. The current method is to lock the query interface based on the source code of the rule analysis home page. Through a large number of observations, starting from the home page of the website, crawling page links based on the principle of width first, the page where the query interface is located will not exceed the layer, and the query interface will not exceed the layer [4]. Based on the characteristics of query interface, three rules are put forward to judge whether there is a query interface in a page. Firstly, there should be a label in the page, secondly, there must be an input control in the label, thirdly, there should be at least one keyword in a group, such as “query”, “search”, etc. This method can achieve at least accuracy in its experiments.

2.2 Detection and Tracking of Target Image

Target image input this topic uses the USB camera of zc0301 microchip to collect image.

In this paper, the task of moving target tracking is to detect and extract such two types of targets in the video image sequence: moving targets and new targets appearing in the scene. Therefore, this task can be completed in two steps, the first is target detection, the second is target extraction [5]. The so-called target detection is to detect whether the monitored scene image in the video sequence image changes. If the image changes, it means that there are new targets, otherwise it means that there are no new targets. Target extraction is to segment and extract the target from the video sequence when the target detection algorithm detects the presence of the target, so as to provide data for the next target tracking. In the process of target tracking, one of the key technologies to track the moving target in detection [6]. The accuracy of target detection will have an important impact on the follow-up steps. In the experiment process, if the target detection is successful, the target detection module will know whether there is a moving target from the image sequence. If there is a moving target detection, the position and size of the moving target in the video image will be given through the algorithm, and then submitted to the target tracking module. The target tracking module will establish the tracking mode and the moving target template according to the given target position and size After extraction, the next step is target tracking. The first information needed to track the target is the target position information. The purpose of target tracking is to analyze the image sequence obtained by the camera, calculate the two-dimensional position coordinates of the target on each frame of the image, and associate the same moving target in different frames of the image sequence according to different characteristic values to get the complete motion track of each moving target, That is to establish the corresponding relationship of moving objects in the continuous video sequence. In short, it is to find the exact location of the target in the next image. In moving target tracking, the main work is to select good target features and use practical search algorithm.

The target tracking system designed in this paper adopts mean shift algorithm.

Mean shift algorithm is a local optimal search algorithm. By calculating the probability density index of the similarity between the candidate target and the target module directly, and then using the direction of probability density gradient decline to obtain the best path for matching search, accelerate the positioning of moving target and reduce the search time [7].

Mean shift algorithm starts from kernel density estimation (also known as Parzen window estimation), which is d relatively popular density estimation method at present. Given that the kernel density estimation of multidimensional variables of n sample data $x_i, i = 1, 2, \ldots, n$ in A-dimensional space R^d can be written as follows: Mean-Shift:

$$\widehat{f_{h,k}}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} |H|^{-1/2} k\left[\left\|\frac{x - x_i}{H}\right\|^2\right] \tag{1}$$

Where $c_{k,d}$ is the normalized constant, $k(x)$ is the kernel function or profile function, and H is the bandwidth matrix of $d \times d$ dimension. A complete representation of H parameters will increase the complexity of the algorithm. In practice, the diagonal matrix $H = diag\left[h_1^2, h_2^2, \ldots, h_n^2,\right]$ or $h^2 I$, I is usually the $d \times d$ - unit matrix. For simplicity, using the latter representation, the kernel density estimate can be written as:

$$\widehat{f_{h,k}}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^{n} k\left[\left\|\frac{x - x_i}{h}\right\|^2\right] \tag{2}$$

Among them, kernel function $k(x)$ must satisfy the following conditions: 1) $k(x)$ is nonnegative; 2) $k(x)$ is monotonically decreasing from the center to the outside, if $0 \leq a \leq b$, then $k(a) \geq k(b)$; 3) $k(x)$ is bounded. When $k(x)$ is differentiable, the gradient can be obtained from Eq. (5–11):

$$\nabla \widehat{f_{h,k}}(x) = \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k' \left[\left\| \frac{x - x_i}{h} \right\|^2 \right] \tag{3}$$

Definition $g(x) = -k'(X)$, obtained from the above formula:

$$\begin{aligned} \nabla \widehat{f_{h,k}}(x) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^{n} (x - x_i) k' \left[\left\| \frac{x - x_i}{h} \right\|^2 \right] \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x \right] \end{aligned} \tag{4}$$

The second half of the formula is the mean shift algorithm:

$$\nabla m_{k,g}(x) = \frac{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x, \widehat{f_{h,g}}(x) \frac{c_{k,d}}{nh^d} \left[\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] \tag{5}$$

Where $c_{g,d}$ is the normalization constant, the above formula can be written as:

$$\nabla \widehat{f_{h,k}}(x) = \widehat{f_{h,g}}(x) \frac{2c_{k,g}}{h^2 c_{g,d}} \nabla m_{h,g}(x) \tag{6}$$

There are:

$$\nabla m_{h,g}(x) = \frac{1}{2} h^2 c \frac{\nabla \widehat{f_{h,k}}(x)}{\widehat{f_{h,g}}(x)} \tag{7}$$

Among them, $c = c_{g,d} I c_{k,d}$. The above formula shows that the local mean value moves towards the dense area of nearby data samples, so there is an iterative formula:

$$y_{t+1} = y_t + \nabla m_{h,g}(y_t) \tag{8}$$

Among them, y_t represents the sample data of step t, y_{t+1} represents the sample data of step $t + 1$. After replacement and simplification, the iteration formula of mean shift algorithm is obtained:

$$y_{t+1} = \frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \tag{9}$$

There are:

$$y_{t+1} = y_t + \lambda_t \bullet d_t \tag{10}$$

Among them, $\lambda_t = h^2 c \Big/ 2\widehat{f_{h,g}}(y_t > 0); d_t = \nabla \widehat{f_{h,k}}(y_t)$ The above formula shows that the mean shift algorithm iterates along the ladder direction, so that each point to be processed "floats" to the local maximum point of the distribution density function, and its step size λ_t Changes adaptively with the iteration process, that is, when the current data is relatively low density, the iteration step size is larger; near the local maximum, the iteration step size is smaller.

2.3 Multi Objective Integration

At present, many multi-objective data integration algorithms only consider the current scheduling, without considering the use of data information before task execution, which is easy to cause too many tasks to be allocated to some advantage information, making the advantage information too busy to become disadvantage information, at the same time causing some information to be idle, some information to be busy, resulting in load imbalance, which is not conducive to the pursuit of UI color interface Find the goal of target data integration [8]. Therefore, in order to achieve load balancing, this algorithm adds a load factor to the scheduling, calculates the load factor when a scheduling cycle is completed, and updates the control information accordingly, as shown in formula (1):

$$v = U_c / U_{sun} \tag{11}$$

Among them: v is the load factor, U_c is the completed task amount, and U_{sun} is all the tasks assigned to the resource.

To enhance the cooperation among targets is helpful to find the optimal solution. A single target is easy to converge to the local optimal, and the communication between targets mainly depends on the diffusion of information [9]. The amount of information is related to the distance of the information source. The information diffuses to the surrounding area with the radius of r as the center of O points of the information source. Assuming that the amount of information of O points is τ_{max}, the formula for calculating the total amount of information of any point 8 in the circular area is as follows (12):

$$\tau_p = (1 - L/r)\tau_{max} \tag{12}$$

In addition, suppose that target k has selected nodes a and b, and the distance between them is L_{ab}. For any node 9 adjacent to nodes a and b, if it is within the scope of target c diffusion information, the amount of information that target c diffuses to the nodes can be calculated.

Set $\tau_{max} = v \bullet \Delta\tau_{ab}^{k}$, $r = \psi \bullet L_{ab}, \psi = L_{ab}/L'$. Where: k is the previously calculated load factor, 14 is the amount of information about the path from target k to nodes a and b, and L‘ is the average distance from all nodes to target k. Then the total amount of information about the path from target k to nodes a, c and b, c is calculated as follows:

$$\Delta\tau_{ac}^{k} = \begin{cases} \frac{vQ}{l_k}\left(1 - \frac{L_{ac}L'}{L_{ac}^2}\right), L_{ac} \quad < r \\ 0 \end{cases} \tag{13}$$

$$\Delta\tau_{bc}^{k} = \begin{cases} \frac{vQ}{l_k}\left(1 - \frac{L_{bc}L'}{L_{bc}^2},\right) L_{bc} < r \\ 0 \end{cases} \tag{14}$$

Where: $\frac{vQ}{l_k}$ is the current pheromone strength, l_k is the distance of target k from the start node to the current node. In this way, formula (13)–(14) can calculate all the information of each target left in the relevant path in each cycle. Each activity of the target will not only affect the information of the path it passes through, but also affect the information of all the paths within the range of its adjacent information diffusion. Through such a local updating method of pheromone, the communication cooperation ability between the targets can be greatly improved, and the algorithm can be enhanced Because of the participation of load threshold coefficient, the load balance performance of the interface can be improved.

2.4 Optimal Path Multi-objective Equilibrium Integration

The ultimate goal of the optimal path multi-objective balanced integration is to simplify the process of the objective, improve the efficiency of implementation, and at the same time minimize the cost of the objective. Therefore, in the integration of the objective process, only considering from the whole can this objective be achieved, so it is necessary to adopt the global optimization. According to the requirements of the efficiency and expenditure of the target object, these two indicators are regarded as the final implementation target of the target integrated calculation [10]. At the same time, considering the credibility and reliability of the target object, the final model is established as follows:

$$MinF(P) = (T(P), C(P)) \tag{15}$$

The constraint conditions of path P subject to reliability and reputation are:

$$\mathrm{Re}\, p(P) \geq \mathrm{Re}\, p_0 \tag{16}$$

$T(P), C(P)$ is the calculation formula of each parameter, which is established by defining different target execution models. By taking the whole as the objective of optimization and adopting constraints, this algorithm can be applied to the calculation of N kinds of target set integration algorithms.

Taking genetic algorithm as an example, the integration process path of each goal is set as a gene chromosome [11–14]. Through gene operations such as cross mutation and recombination of chromosomes, the next generation of chromosomes will be generated. The new chromosome is closer to the set target value, and the next generation of chromosomes will be continuously generated through gene operation method, and finally the set target value will be achieved Convergence. Let the target population be medium gene population P_1 and limited population P_2, the algebra of chromosome evolution is T, and the final path result is P^*. The operation flow of gene evolution target is shown in Fig. 1.

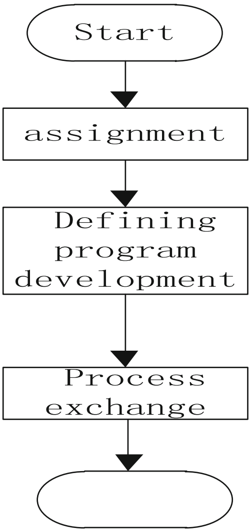

Fig. 1. Operation flow of target evolution

Each chromosome target evolution process needs an initial evolution path. In this calculation, a set of initial target paths is randomly generated, and the initial target paths are constrained by the set constraint algorithm to obtain an optimal target path generated by random numbers. The optimal target path method generated by random numbers is shown in Fig. 2.

According to the theory of gene evolution, in the same way, only each generation selects the highest quality target data to enter the next generation of gene genetic operation can get the optimal solution data, so it is necessary to judge the quality of the target data.

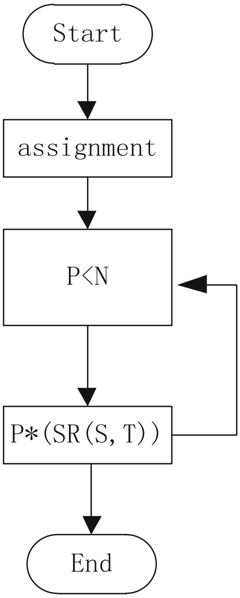

Fig. 2. Optimal target path of random number generation

3 Experiment and Analysis

In order to verify the reliability and efficiency of the proposed integration method, traditional detection methods are selected and applied to the test of dynamic integration method for detecting multi-target data. The proposed method is used as experimental group A and traditional method as experimental group B. Taking the video information collected by the front-end CCD camera as the input of the video image, the algorithm processing of moving target detection and tracking is carried out on the development board, and then the processed video image is transmitted to the large screen display.

3.1 Experiment Preparation

We collect CIF (352×288 resolution) images in YUV422 format to Da Vinci platform in real time, and start the moving target detection and tracking system. The first 50 frames are used to get the position of the moving target, and then the tracking and integration are started. Two integration methods are used to detect and integrate the target data sequence. Get and analyze the experimental test results.

3.2 Experimental Results and Analysis

Under the condition of keeping other experimental conditions unchanged, the two data integration methods are compared, and the target tracking performance during the inspection period is shown in the following table:

Table 1. Comparison of target tracking performance statistics

	Experimental B group (traditional methods)	Experimental A Group (Methods)
Frame rate	21	18
DSP load	24%	20%
ARM load	1%	0%
Time(s)	9	8

According to Table 1, the real-time performance index of the system when the moving target tracking integration algorithm is used, it can be seen that the integration method designed in this paper is fully in place in data integration, the occupancy rate of arm end is 0%, the load line of DSP is 20%, the system maintains reliable real-time performance, and achieves the ideal state of UI color interface operation. On the contrary, the traditional method of data integration is not in place; Under the same conditions, the time used is 1 s less than the traditional method, which improves the integration efficiency. It can be seen that the traditional integration method is not suitable for the requirements of UI color interface with large target data. Comprehensive experimental results show that the proposed multi-objective data integration method can perform data integration in the UI color interface with large target data.

4 Conclusion

Because traditional integration methods cannot meet the needs of UI color interfaces with large target data. The proposed multi-target data integration method can perform data integration in a UI color interface with larger target data, and has high practical applicability. This paper analyzes the multi-objective data integration method of UI color interface. Combined with the technical characteristics of UI color interface, the relevant algorithm is used to optimize parameters and improve the integration rate of multi-objective data. It is hoped that the multi-objective data integration method of UI color interface designed in this paper can provide theoretical basis for the application of UI color interface in China.

References

1. Zeng, J., Dou, L., Xin, B.: Multi-objective cooperative salvo attack against group target. J. Syst. Sci. Complex. **31**(1), 244–261 (2018)
2. Zhang, X., Tan, Y., Yang, Z.: Resource allocation optimization of equipment development task based on MOPSO algorithm. J. Syst. Eng. Electron. **30**(6), 1132–1143 (2019)
3. Gao, K., Cao, Z., Zhang, L., et al.: A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. IEEE/CAA J. Automat. Sin. **6**(4), 904–916 (2019)
4. Cheng, S., Lei, X., Lu, H., et al.: Generalized pigeon-inspired optimization algorithms. Sci. China (Inf. Sci.) **62**(7), 120–130 (2019)

5. Hu, Y., Wang, J., Liang, J., et al.: A self-organizing multimodal multi-objective pigeon-inspired optimization algorithm. Sci. China (Inf. Sci.), **62**(7), 73–89 (2019)
6. Yan, L., Qu, B., Zhu, Y., et al.: Dynamic economic emission dispatch based on multi-objective pigeon-inspired optimization with double disturbance. Sci. China (Inf. Sci.) **62**(7), 108–119 (2019)
7. Yang, Yu., Gao, S., Wang, Y., et al.: Global optimum-based search differential evolution. IEEE/CAA J. Autom. Sin. **6**(2), 379–394 (2019)
8. Shuai, L., Gelan, Y.: Advanced Hybrid Information Processing, pp. 1–594. Springer, USA. https://doi.org/10.1007/978-3-030-36402-1
9. Liu, A., Deng, X., Ren, L., et al.: An inverse power generation mechanism based fruit fly algorithm for function optimization. J. Syst. Sci. Complex. **32**(2), 634–656 (2019)
10. Sun, J., Ling, B.: Software module clustering algorithm using probability selection. Wuhan Univ. J. Nat. Sci. **23**(2), 93–102 (2018)
11. Gong, D.W., Sun, J., Miao, Z.: A set-based genetic algorithm for interval many-objective optimization problems. IEEE Trans. Evol. Comput. **22**(99), 47–60 (2018)
12. Bradford, E., Schweidtmann, A.M., Lapkin, A.: Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. J. Glob. Optim. **71**(2), 1–33 (2018)
13. Liu, S., Bai, W., Liu, G., et al.: Parallel fractal compression method for big video data. Complexity **2018**, 2016976 (2018). https://doi.org/10.1155/2018/2016976
14. Ben Elghali, S., Outbib, R., Benbouzid, M.: Selecting and optimal sizing of hybridized energy storage systems for tidal energy integration into power grid. J. Mod. Pow. Syst. Clean Energy **7**(1), 113–122 (2019)
15. Liu, S., Lu, M., Li, H., et al.: Prediction of gene expression patterns with generalized linear regression model. Front. Genet. **10**, 120 (2019)