Computer Science and Application 计算机科学与应用, 2021, 11(1), 75-83
Published Online January 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.111009

Hans汉斯

哈希变换在信息处理中的应用研究

王连亮[1]，焦　璐[2]

[1]西南电子技术研究所，四川 成都
[2]成都飞机设计研究所，四川 成都
Email: wlley@163.com

收稿日期：2020年12月24日；录用日期：2021年1月18日；发布日期：2021年1月26日

摘　要

介绍了哈希映射技术的基本原理，并针对C++编程语言，给出了一种基于数组的哈希映射实现方法。然后，针对目标关联应用，对哈希映射技术进行了数值模拟，结果表明，该技术可显著提高数据关联的处理效率。最后，对哈希映射技术在信息处理中的典型应用进行了探讨。

关键词

哈希变换，数据关联，数据融合，信息处理

The Research of Hash Translation Application in Information Processing

Lianliang Wang[1], Lu Jiao[2]

[1]Southwest China Institute of Electronic Technology, Chengdu Sichuan
[2]Chengdu Aircraft Design & Research Institute, Chengdu Sichuan
Email: wlley@163.com

Received: Dec. 24th, 2020; accepted: Jan. 18th, 2021; published: Jan. 26th, 2021

Abstract

The basic principle of Hash mapping technology is described, and an array-based method of Hash mapping about C++ program language is provided. Secondly, a value simulation is provided about Hash mapping technology in the application of target association, and the result shows that the method proves evidently the efficiency of data association. Finally, a discussing about the typical application of Hash mapping technology in information processing is provided.

文章引用: 王连亮, 焦璐. 哈希变换在信息处理中的应用研究[J]. 计算机科学与应用, 2021, 11(1): 75-83.
DOI: 10.12677/csa.2021.111009

Keywords

Hash Translation, Data Association, Data Fusion, Information Processing

1. 前言

信息处理技术是现代作战平台中不可或缺的处理环节，通常具有高时效要求；特别对于高对抗的战斗机平台，对信息处理的时效性要求更为严苛。

数据融合技术是重要的信息处理技术，包括数据时空配准、数据关联、状态估计、属性识别等过程[1] [2]，通常存在大量的数据匹配处理过程，如传感器数据由极坐标系转到地理坐标系时，需要本机导航提供的位姿信息，从而需要传感器数据与本机导航数据在时间上保持一致，需要通过匹配搜索使传感器的时间戳与导航数据保持一致或接近。又如在目标航迹管理中，当对某一特定批号的航迹信息进行更新时，需要通过搜索过程快速查找到该条航迹以更新相应信息。当目标航迹数较多时，上述搜索过程将会降低处理的时效性。

当某一特定数据元素在数据存储区中的存储位置是随机的，且与数据元素的关键字之间不存在确定关系时，在数据存储区中查找某一特定数据元素时需要一系列搜索比较过程。查找的效率依赖于查找过程中所进行的比较次数。哈希变换技术是基于一种查找表的查找技术，在提高数据查找效率方面具备较好的性能，在数据挖掘、数据检索、网络路由、信息安全等领域有着广泛的应用[3] [4] [5] [6] [7]。

常用的哈希算法是局部敏感哈希算法[8]，把相似度高的数据对象尽可能地哈希到同一个数据桶中，并且数据相似度越高哈希到同一个数据桶中的概率也越高。在进行数据搜索时，以同一个数据桶中的数据对象作为候选对象，依次计算候选对象与查询对象的距离。在软件编程实现中，通常用数据队列、多维数组等方法来实现对数据桶的存储，该方法将会存在较大的存储空间浪费。当硬件存储条件受限时，需要设计更加优化的哈希实现方法。

本文结合哈希变换技术原理，设计了基于动态溢出表的哈希表示方法，并对信息处理领域中若干与数据查找相关的问题进行探讨。

2. 哈希变换简介

哈希映射技术是将无序存储的数据元素表映射为有序存储的哈希表的过程，如图 1 所示。首先按照一定的映射规则将无序存储的数据元素映射到数据空间中，再将每一个空间单元对应的数据元素按照一定的数据结构存储为哈希表。当两组数据元素进行匹配时，只需要对各对应空间单元及其附近空间单元中的数据进行比较即可，从而提高了数据查找匹配的效率。

哈希变换技术实现主要包括哈希函数构造与哈希造表等过程[9] [10]。

1) 哈希函数构造

哈希函数的作用是将一组关键字映象到一个有限的连续的地址集上，并以关键字在地址集中的“象”作为记录在表中的存储位置，这种表便称为哈希表。哈希函数的形式为：

$$y = f(x) \tag{1}$$

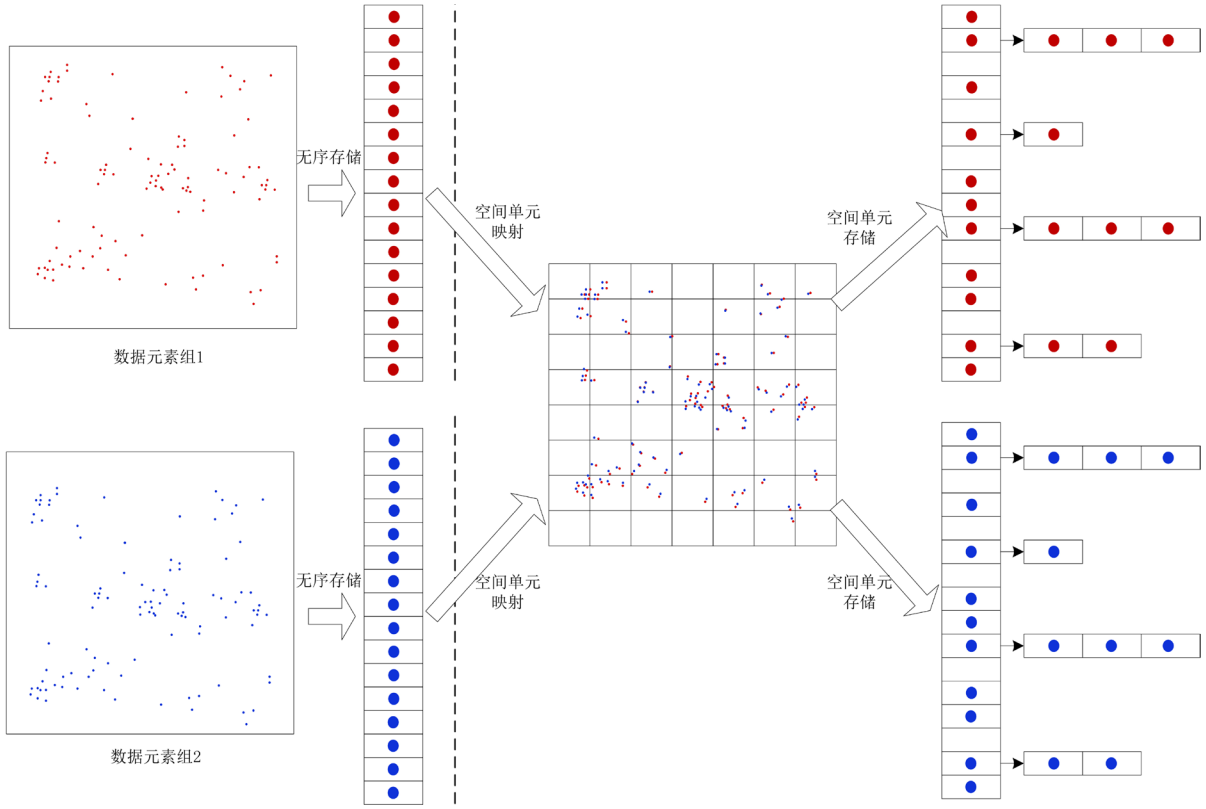


Figure 1. The sketch map of Hash translation and data searching
图 1. 哈希变换与数据查找示意图

式中，x 为查找表中数据元素的关键字，y 称为哈希地址，表示对应于 x 的数据元素在哈希表中的存储位置。

构造哈希函数的方法包括直接定址法、基数转换法、平方取中法、折叠法、除留余数法、随机数法、提取法等。实际工作中需视不同的情况采用不同的哈希函数。通常，考虑的因素有：

a) 计算哈希函数所需时间；

b) 关键字的长度；

c) 哈希表的大小；

d) 关键字的分布情况。

2) 哈希造表

根据哈希函数，对查找表中顺序存储的每条数据记录计算哈希地址，存入哈希表中。由于哈希地址冲突的存在，哈希表往往由基本表和溢出表构成。

基本表内存储的是溢出表的表头，溢出表为数据元素的单向链表，由数据元素与后继地址构成。基本表的存储长度 M 由哈希函数的取值范围确定；溢出表中各队列的长度 l 由可能分配给该队列的最大数据元素个数确定。假设查找表中最大存在 N 条数据记录，则溢出表队列长度 l 最大取值为 N。

图 2 为哈希函数为 $f(x) = x \bmod 13$ 的数据元素的基本表以及溢出表。

在 Vxworks 等嵌入式操作系统中，利用 C++程序语言实现溢出表时，需要事先定义固定大小的数组，即用二维数组 Q [M] [N]表示，共占用 M × N 个存储单元。实际上，数组 Q [M] [N]中最多只使用了 N 个存储单元，从而将浪费(M − 1) × N 个存储单元。当 N 与 M 值很大时，浪费的存储单元将不可忽视。

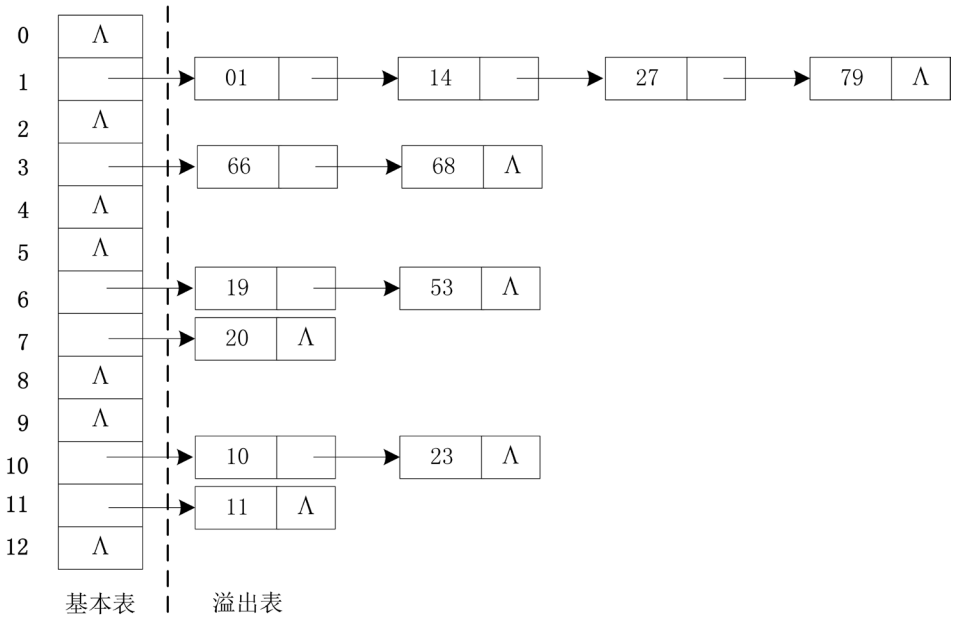


Figure 2. The sketch map of the data structure of hash table
图 2. 哈希表数据结构示意图

为了最大化利用存储空间，基于 C++程序语言的特点，本文设计了一种基于两个独立一维数组的哈希表表示方法，将二维数组的静态溢出表表示替换为由一维数组的动态溢出表表示。

3. 基于动态溢出表的哈希表表示方法

由上一节可知，基本表的本质是建立查找表中的数据记录与哈希函数值的对应关系；溢出表的本质是存储具有相同哈希函数值的数据记录的集合，其实际存储单元数目等于总的数据记录数目，因此，溢出表可用与查找表相同长度的一维数组表示。

1) 哈希表构造方法

假设 N 个数据元素的查找表用一长度为 N 的 d 维数组表示 V[N]，d 为数据元素的维数；哈希地址的取值范围为[0, M − 1]，即哈希表中基本表的长度为 M，用长度为 M 的一维数组 B[M]表示；哈希表的溢出表用长度为 N 的一维数组表示 O[N]，如图 3 所示。数组 B[M]、数组 V[N]以及数组 O[N]共同构成了若干个单项链表。其中，B[M]中存储链表表头地址，“∧”表示无效链表；O[N]中存储后继节点，“∧”表示无后继节点，即链表尾节点；V[N]中存储原始数据记录。

对于 $r \in [0, M-1]$，$s \in [0, N-1]$，$t \in [0, N-1]$，若 s = B[r]，则表示 V[s]的哈希函数值为 r；若 t = O[s]，则表示 V[s]，与 V[t]对应同一哈希地址，且为同一单向链表的两个相邻节点，V[t]为 V[s]的后继节点。

利用图 3 数据结构，对查找表 V[N]中的数据建立哈希表的过程如表 1 所示。

2) 存储量分析

基于动态溢出表的哈希表所占用的最大存储空间为(N + M) × L Bytes，其中，L 为基本表数组和溢出表数组中单个元素的字节数，取决于 N 的大小。

当利用哈希表对 V[N]中的某条数据记录进行检索时，首先利用哈希函数计算在 B[M]中的位置，然后在 O[N]中搜索相应的链表队列，以确定给定数据记录的存储位置。特别地，当 M 足够大而使得 V[N]中存储的数据记录与 B[M]中的有效地址数据一一对应时，则不需要哈希溢出表 O[N]，此时对 V[N]中的数据查询效率最高。

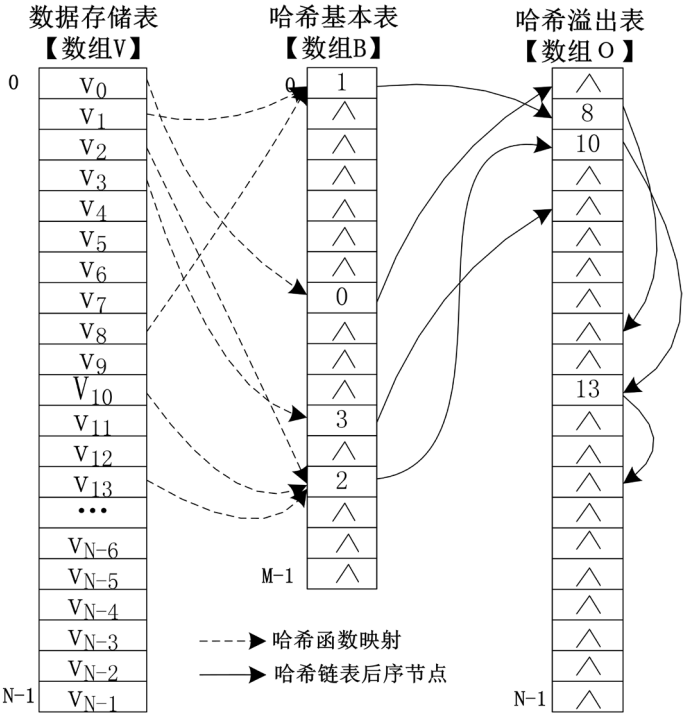


Figure 3. The implement principle figure of hash table data structure
图 3. 哈希表数据结构实现原理

Table 1. The building process of hash table
表 1. 哈希表建立过程

1) 初始化 B[M]与 O[N]中的每个存储单元为无效地址值 InvalidAdrr (令 InvalidAdrr = -1);
2) 遍历 V[N]中的每个条数据记录 V[i] (0 ≤ i < N)：
{
　　计算 V[i]的哈希函数值 h(V[i])，记为 j，则考察 B[j]值：
　{
　　若 B[j]=InvalidAdrr，则将 B[j]赋为 i；否则考察 O[k]值：
　　{
　　若 O[k]=InvalidAdrr，则将 O[k]赋为 i；
　　否则进行如下循环：
　　While(O[k]≠InvalidAdrr)
　　{
　　k =O[k]；
　　}
　　将 O[k]赋为 i。
　　}
　}
}

3) 计算复杂度分析

假设 V[N]的哈希表包含 m 个哈希子表，第 j 个哈希子表的长度为 $n_j\left(1 \le j \le m\right)$，则有：

$$\begin{cases} \sum_{j=1}^{m} n_j = N \\ n_j > 0 \end{cases} \tag{2}$$

对 V[N]中所有 N 个数据记录进行检索，所用的比较次数 C 为：

$$C=\sum_{j=1}^{m}\frac{n_j\left(1+n_j\right)}{2}=\frac{1}{2}\left(N+\sum_{j=1}^{m}n_j^2\right) \tag{3}$$

由式(2)、式(3)知，当 $m=1$ 时，$C=\frac{1}{2}\left(N^2+N\right)$，此时即为没有进行哈希变换时的搜索次数；当 $m=N$ 时，$C=N$，此时只有基本表，没有溢出表，搜索次数最少；其余情况，C 将介于二者之间。

4. 数值模拟分析

针对两种数据源观测值之间匹配搜索过程所占用的时间进行数值模拟分析，以说明哈希变换技术对数据查找匹配处理中的时效性。

假设存在 N 个战场目标，平均分布在观测平台周围 360 度、距离 300 km 范围内；观测平台搭载传感器 A 与传感器 B，分别对这 N 个目标进行观测，获得 N 个观测数据 A[N]与 B[N]，每个观测数据均包括方位角 θ 与距离 d 信息；传感器 A 与传感器 B 的方位角精度均为 $\sigma_\theta=0.5$ 度，距离精度均为 $\sigma_d=200$ m。

在本例中，根据方位角 θ 与距离 d，构建哈希变换表。构建哈希函数如下：

$$y=f\left(\theta,d\right)=\frac{\theta}{\Delta\theta}+\frac{\mathrm{d}}{\Delta\mathrm{d}}\times\frac{360}{\Delta\theta}$$

式中，y 为哈希地址；$\Delta\theta$ 为方位向的分区宽度，Δd 为距离向的分区宽度。

设定不同的匹配方法如表 2 所示。M0 为不采用哈希技术的全遍历匹配方法，M1~M3 为采用哈希技术的哈希遍历匹配方法，但设置了不同的哈希变换参数。

Table 2. Account for the four matching methods
表 2. 四种匹配方法说明

方法标识	方法说明	哈希函数参数
M0	全遍历匹配方法	——
M1	本文哈希匹配方法	$\Delta\theta=16^\circ$, $\Delta d=20$ km
M2		$\Delta\theta=8^\circ$, $\Delta d=8$ km
M3		$\Delta\theta=4^\circ$, $\Delta d=4$ km

在 VC++环境下，依据表 2 中不同遍历方法，考察传感器 A 观测值 A[N]与传感器 B 观测值 B[N]的关联处理耗时如表 3 所示。

由表 3 可知，随着目标数的增大，哈希匹配方法与遍历匹配方法相比，在时效性上的优势越来越明显，耗时缩短了 1~3 个数量级；当目标数相同时，通过哈希函数映射将原始数据集分的数据桶越精细，所耗时间越少。表 3 中四种匹配方法的时间 t (ms) 随目标个数 N 的变化曲线如图 4 所示。

由图 4(a)知，M0 方法的耗时 t 随目标数 N 呈指数增长；而采用了哈希变换技术的 M1~M3 方法的耗时 t 随目标数 N 呈近似线性增长关系。通过拟合，M0~M3 方法的计算复杂度如表 4 所示。

由图 4(b)可知，M1~M3 方法，随着哈希变化精细程度的增大，处理耗时 t 也呈减小趋势。

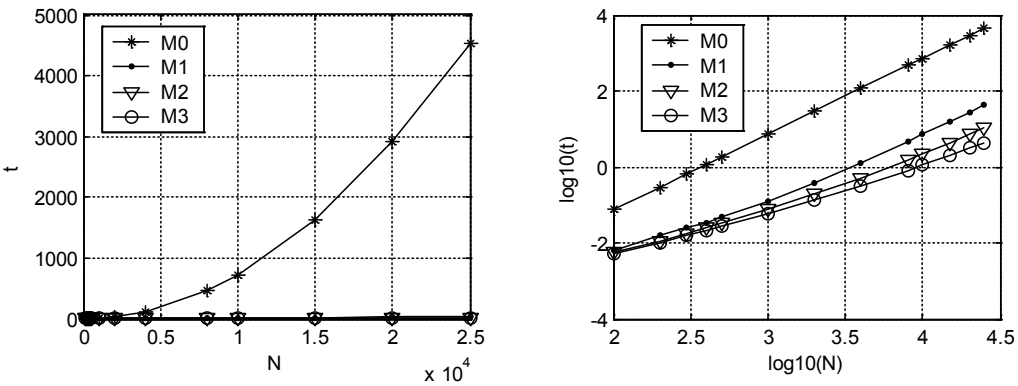
通过数值模拟可知，采用哈希遍历方法能够显著提高匹配处理的时效性，且哈希变换的精细程度越高，时效性提高的越大。

Table 3. The comparing of consuming time about the four matching methods
表 3. 四种匹配方法所用时间对比

目标数 N	M0 (ms)	M1 (ms)	M2 (ms)	M3 (ms)
100	0.078	0.0065	0.0056	0.0050
200	0.286	0.0159	0.0106	0.0102
300	0.649	0.0253	0.0175	0.0152
400	1.156	0.0351	0.0246	0.0211
500	1.812	0.0483	0.0328	0.0268
1000	7.19	0.1234	0.0766	0.0594
2000	29.1	0.3625	0.1906	0.1375
4000	117	1.278	0.5156	0.3265
8000	469	4.688	1.516	0.809
10,000	728	7.188	2.172	1.125
15,000	1640	15.68	4.329	2.015
20,000	2922	27.65	7.19	3.12
25,000	4540	42.34	10.78	4.37

Table 4. The computing complexity of the four matching methods
表 4. 四种匹配方法的计算复杂度

	M0	M1	M2	M3
计算复杂度	$O(N^2)$	$O(N^{1.59})$	$O(N^{1.37})$	$O(N^{1.23})$



(a) t~N 曲线　　(b) $\log_{10}t$~$\log_{10}N$ 曲线

Figure 4. The consuming time changing curve with target number changing about the four matching methods
图 4. 四种匹配方法耗时随目标数目的变化曲线

5. 哈希变换在信息处理中的应用探讨

在信息处理中，通常用到数据检索、匹配等数据查找问题，若采用遍历搜索查找的方法，在数据量大时，查找效率会很低。采用哈希变换技术进行优化处理，可以显著提高处理效率。下面针对信息处理中常遇到的一些查找问题，对哈希变换技术在信息处理中的应用进行探讨。

1) 数据关联

数据关联处理是传感器目标跟踪处理与多源传感器数据融合处理中最基本的处理过程，其基本原理就是确定两种或多种数据源之间特征最相近的目标观测值。通常所讲的“分区存储关联”方法实际上就是哈希变换技术的典型应用。

当目标数目较大时，多种数据源观测之间匹配搜索过程所占用的时间将会显著影响整个数据关联处理的时效性；哈希变换技术可显著改善此时数据关联处理的时效性。

2) 目标识别

目标识别通常是在目标识别特征库的支持下，利用目标的观测特征对未知目标的属性进行识别[11]。在这个识别过程中，通常需要将目标的观测特征值与目标识别特征库的特征识别模板进行匹配。

为了目标识别的准确性，需要对目标的各种姿态、各种环境以及传感器的各种模式等条件下建立目标的特征识别模板，从而使得一个目标的特征识别模板数量巨大；同时，目标特征识别库需要对不同的战场目标建立特征识别模板，已达到目标分辨的目的。因此，目标特征识别库中的目标特征识别模板数量庞大，在目标识别过程中，利用目标观测特征与所有特征识别模板进行匹配达不到目标实时识别的目的。

为了实现实时目标识别，需要对目标识别特征库中的特征模板的存储方式进行特殊设计，采用哈希变换技术的特征库设计方式是一种较好的选择，该方法的优点是不需要原有特征库的存储方式，只需额外增加存储相应的哈希表即可。

首先在目标特征识别模板中选定具有代表性的数据项构造哈希函数，根据哈希函数对各特征识别模板确定哈希地址，建立相应的哈希表，保存在目标识别特征库中；在目标识别时，只需要计算目标观测特征的哈希地址，根据该哈希地址索引到相应的目标识别特征集合，然后再进行识别匹配，获得相应的目标识别结果，从而达到实时识别的目的。

3) 图像配准

图像配准处理是将对同一地理空间的两幅独立拍摄的图像进行匹配处理，是图像融合、图像校正、图像对比分析等处理的基础。在图像配准处理过程中，需要对各图像分别独立地提取特征点，然后对图像之间的特征点进行匹配，确定两图像中的特征点的对应关系[12]。这个匹配过程就是两组特征点之间的匹配搜索问题，是哈希变换技术的典型应用问题。

采用哈希变换技术，对两组特征点的特征值建立哈希表，在哈希表基础进行匹配搜索将会显著提高图像特征点的匹配效率，从而改善图像配准的处理时效性。

4) 目标态势显示

在目标态势显示中，往往将目标按照不同类型、不同敌我属性、不同区域等多种要素进行重点或过滤显示。当目标较多时，采用遍历查找的方式选择给定目标要素的目标进行显示会影响目标显示效果。为了提高目标显示速度，分别针对目标类型、目标敌我属性、目标所处区域等因素，采用哈希变换技术建立哈希表，然后再根据相应的哈希表对目标进行显示，可显著改善目标显示效率。

5) 大数据处理

大数据处理是信息处理的发展趋势，而 Hadoop 是大数据处理的重要途径[13]。在 Hadoop 平台的 MapReduce 处理机制中，如有多个 reduce 任务，则每个 map 任务都会为每个 reduce 任务建立一个分区。分区由用户定义的分区函数控制，但通常默认的分区器是通过哈希函数实现的，这种方法很高效。

6. 结束语

哈希变换技术是一种常用的数据查找技术，其原理简单，易于实现，在工程中有着广泛的应用。在

信息处理领域中，数据查找是常见的处理过程，当数据量较大时，数据查找就会影响着整个信息处理的时效性，而哈希变换技术正是解决该问题的重要途径。

参考文献

[1] García, F., Martín, D., de la Escalera, A. and Armingol, J.M. (2017) Sensor Fusion Methodology for Vehicle Detection. *IEEE Intelligent Transportation Systems Magazine*, **9**, 123-133. https://doi.org/10.1109/MITS.2016.2620398

[2] Xiong, J., Shu, L., *et al.* (2017) A Scheme on Indoor Tracking of Ship Dynamic Positioning Based on Distributed Multi-Sensor Data Fusion. *IEEE Access*, **5**, 379-392. https://doi.org/10.1109/ACCESS.2016.2607232

[3] 钱江波, 胡伟, 陈华辉, 等. 基于学习型哈希的在线近邻查找算法[J]. 控制与决策, 2019, 34(12): 2567-2575.

[4] 罗旋, 李永忠. ModbusTCP 安全协议的研究与设计[J]. 数据采集与处理, 2019, 34(6): 1110-1117.

[5] 张庭. 面向时空数据流的分布式索引[D]: [硕士学位论文]. 杭州: 浙江工业大学, 2019.

[6] 徐盼盼. 基于哈希学习的图像检索方法研究[D]: [硕士学位论文]. 沈阳: 沈阳工业大学, 2019.

[7] 胡珂, 李成名, 沈建明. 分布式时空数据库去中心化负载均衡方法[J]. 测绘通报, 2018(12): 65-68.

[8] 李丹阳, 程晓荣. 一种基于哈希方法的相似性搜索[J]. 中国科技信息, 2018(13): 51-52.

[9] 李华, 李莉, 郭育艳. 哈希编码和 Kalman 滤波的目标跟踪算法[J]. 沈阳工业大学学报, 2019, 41(4): 406-411.

[10] 刘昊鑫, 吴小俊, 庾骏. 联合哈希特征和分类器学习的跨模态检索算法[J]. 模式识别与人工智能, 2020, 33(2): 160-165.

[11] 徐融, 赵飞, 周锦松. 空间点目标光谱探测与特征识别研究进展[J]. 光谱学与光谱分析, 2019, 39(2): 333-339.

[12] 贾镕, 季航, 拓浩男. 一种基于尺度不变的景像匹配方法[J]. 舰船电子工程, 2020, 40(1): 29-36.

[13] 刘贤康. 基于 Hadoop 生态圈的工业数据平台设计与研究[D]: [硕士学位论文]. 武汉: 华中科技大学, 2019.