



压缩域目标跟踪算法在小型化 DSP 平台上的研究与实现

程卫亮, 王向军*, 万子敬, 郭志翼

天津大学微光机电教育部重点实验室, 天津 300072



摘要: 本文对基于压缩感知的压缩域目标跟踪算法进行了研究, 为满足特定的应用场合要求, 针对原算法的不足进行了改进, 同时基于小型化低成本目标位置探测器设计思想及需求, 设计并实现了以 TMS320DM6437 数字信号处理器为核心的实时图像跟踪处理平台, 对算法在该 DSP 平台进行了实现与优化。仿真和实验结果表明, 经过结合卡尔曼滤波器、融合 LBP 特征以及添加自适应学习速率更新策略等措施, 算法的鲁棒性得到提高; 对算法在 DSP 中的实现, 经过一系列优化措施, 对分辨率为 960×960 的视频图像, 当取目标窗口为 80×80 时, 处理速度可达 25 f/s, 能够满足实时性跟踪要求。系统能够对选定的运动目标进行连续、稳定地跟踪, 能够满足特定应用场合下的目标位置探测与跟踪需求, 具有一定的实用性, 同时也对该类目标跟踪方法在嵌入式平台的研究与应用具有一定的参考价值。

关键词: 目标跟踪; DSP; 压缩域; 算法优化; 实时跟踪

中图分类号: TP391.41

文献标志码: A

Research and implementation of target tracking algorithm in compression domain on miniaturized DSP platform

Weiliang Cheng, Xiangjun Wang*, Zijing Wan and Zhiyi Guo

MOEMS Education Ministry Key Lab, Tianjin University, Tianjin 300072, China

Abstract: The target tracking algorithm in compression domain based on compression perception is studied. To meet the specific application requirements, the shortcomings of original algorithm are improved. At the same time, based on the design idea and demand of miniaturized target position detector, a real-time image processing platform with TMS320DM6437 digital signal processor as the core is designed and implemented, and the algorithm is implemented and optimized on the DSP platform. The simulation and experiment results show that after the combination of Kalman filter, LBP feature and adding adaptive learning rate update strategy, the stability of the algorithm is improved. For the implementation in DSP, after a series of optimizing measures, as for an image with resolution of 960×960, taking the target window of 80×80 into account, the computation speed can be up to 25 fps, which can meet the requirement of real-time tracking. The embedded tracking system can track the selected moving objects continuously and stably, and can meet the target localization and tracking requirements under specific applications, which has a real practical value. Moreover, the method in this paper has a certain reference value for the research and applications of this kind of target tracking method in the embedded platform.

Keywords: target tracking; DSP; compression domain; algorithm optimization; real-time tracking

DOI: 10.3969/j.issn.1003-501X.2017.10.005

Citation: *Opto-Elec Eng*, 2017, 44(10): 972-982

收稿日期: 2017-06-04; 收到修改稿日期: 2017-09-14

*E-mail: xdocuxjw@vip.163.com

1 引言

目标跟踪是计算机视觉系统的关键技术,在军事和民用的各个领域都有着重要的应用价值,如武器引导、智能交通系统、医用图像系统以及虚拟现实等^[1]。目标跟踪的实质就是在视频的连续帧中确定目标的位置^[2]。经典的视频目标跟踪思路基于代表目标物体的表面模型,将问题转化为求解模型与候选分布相似度系数最大化的问题:搜索图像感兴趣区域(region of interest, ROI)并找到与目标模型分布最相似的分布。该类方法往往计算量较大,在资源有限的计算平台上,目标跟踪的鲁棒性、精度与实时性难以达到较好的平衡。基于判决模型的跟踪方法是当前目标跟踪领域的研究热点^[3-5],它把目标区域的定位视为二元分类问题,通过设计分类器有效区分背景和目标,进而实现跟踪目的^[6]。该类方法往往能够达到较高的帧速度,但是在嵌入式处理平台的研究与应用还不够完善。

基于压缩感知的压缩域目标跟踪算法融合了这两类目标跟踪思路,通过将目标与候选样本特征映射到压缩域中进行处理,并使用在线学习的朴素贝叶斯分类器分类,具有较小的计算量、较高的精度和稳定性,是实时跟踪领域内研究的热点模型,但是对一些场合下的要求以及在嵌入式系统中的应用,仍然存在一些问题有待改进。Yang 等^[7]利用在线向量增强(online vector boosting feature selection, OVB)的思想,通过使用欧氏距离(Euclidean distance)以及设计不同的包(bag)来容纳不同大小的图像块进而选择出最具有区分性的特征,并采用特征模板在线更新策略,有效减少了目标在发生较大形变时算法的漂移,但是该方法计算复杂度高,消耗资源量较大,在低负载计算平台上难以实现实时效果;Chan 等^[8]添加使用光照不变特征局部敏感直方图(locality sensitive histograms, LSH),提升了对光照变化的抵抗力,同时分类函数引入彩色属性进行加权来弥补类哈尔(Haar-like)特征的不足,但是该方法不适用于需要采用单色图像传感器的场合下,应用受限;Yao 等^[9]通过结合运动模型有效地减少目标在运动模糊、形貌迅速变化场景下的漂移,但是对较为常见的光照变化迅速的应用场合适应性不强。本文针对实际应用场合需求结合嵌入式 DSP 系统的处理能力,对原算法进行了一定的改进,在算法仿真验证的基础上,对改进后的压缩域目标跟踪方法在低负载 DSP 平台的应用进行了研究与实现,通过一系列的优化措施,在保证系统有效性和稳定性的基础上,有效

提高了算法的实时性。

2 基于压缩感知的压缩域目标跟踪算法

2.1 压缩域目标跟踪算法框架

压缩域目标跟踪算法中,首先利用符合压缩感知等距有限性质(restricted isometry property, RIP)的随机感知矩阵对多尺度图像特征进行降维,然后在降维后的特征上采用朴素贝叶斯分类器进行分类^[5]。

该跟踪算法利用基于压缩感知的公式提取特征:

$$V = PX, \quad (1)$$

其中: $X \in R^{n \times 1}$ 为原始信号,大小为 $n \times 1$ 维的矩阵, $P \in R^{k \times n}$ ($k \ll n$) 为测量矩阵, $V \in R^{k \times 1}$ 为压缩后的数据。根据 Johnson-Lindenstrauss 引理,测量矩阵可以采用一个非常稀疏的形式^[10]:

$$r_{ij} = \sqrt{\rho} \times \begin{cases} 1 & P = 1/2\rho \\ 0 & P = 1-1/\rho, \\ -1 & P = 1/2\rho \end{cases} \quad (2)$$

当 $\rho = O(m)$ 时,该矩阵渐近满足正态分布。取 $\rho = m/4$ 能够得到一个非常稀疏的随机矩阵,对 P 每一行,只需要 ≤ 4 数目的内容需要计算。

实际特征提取如下:

$$v_i = \sum_{k=1}^{Nr} \frac{P_{i,k} Rects_{i,k}}{\sqrt{Nr}}, \quad (3)$$

其中: $Rects$ 为在样本区域随机选取的图像块, Nr 为图像块数目($2 \sim 4$), $P_{i,k}$ 值在 1 与 -1 等概率随机选取,它在同一图像块中是相同的。由上式可知该特征表现为几何区域的加权和,也即一种 Haar-like 特征,可借助积分图来实现快速计算^[5],如图 1 所示。

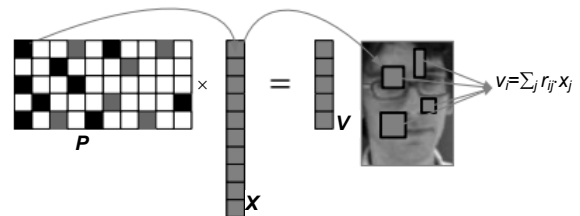


图 1 特征压缩。

Fig. 1 Feature compression.

对新一帧图像,在上一目标位置的邻近区域选取样本,通过朴素贝叶斯分类器进行分类、判别。Diaconis 等人证实高维随机向量的随机投影几乎是高斯分布的^[11],因此,分类器中的条件分布认为是高斯

分布,以4个参数 $(\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0)$ 表示:

$$H(v) = \log\left(\frac{\prod_{i=1}^k p(v_i | y=1)p(y=1)}{\prod_{i=1}^k p(v_i | y=0)p(y=0)}\right) = \sum_{i=1}^k \log\left(\frac{p(v_i | y=1)}{p(v_i | y=0)}\right), \quad (4)$$

其中 $\begin{cases} p(v_i | y=1) \sim N(\mu_i^1, \sigma_i^1) \\ p(v_i | y=0) \sim N(\mu_i^0, \sigma_i^0) \end{cases}$.

选取 $H(v)$ 最大的候选区域作为新一帧的目标区域,并对分类器参数进行更新:

$$\begin{cases} \mu_i^1 \leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1 \\ \sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \end{cases}. \quad (5)$$

2.2 改进的压缩域目标跟踪算法

基于压缩感知的目标跟踪算法框架直接基于图像灰度值计算 Haar-like 特征,在光照变化较为迅速的场合下不够稳定;由于压缩矩阵在整个跟踪过程中固定,对于形貌迅速变化的目标,算法适应性较弱,容易发生漂移甚至丢失目标;原算法中贝叶斯分类器采用固定的学习速率,在目标被短暂遮挡的情况下,参数的过快更新使去除遮挡后目标很容易丢失。此外,贝叶斯分类器在最优位置处的置信度有时候并不高,在这样的情况下,目标位置很大程度上由负样本决定,从而目标位置的精度受到影响,容易产生漂移,使整个跟踪系统不够稳定。针对这些不足,提出了如下改进措施:

2.2.1 融合 LBP 与 Haar-like 特征的样本压缩特征

针对原算法在光照变化较为迅速时不够稳定的缺点,将局部二值模式(local binary pattern, LBP)特征融

合到原 Haar-like 特征计算框架中进行改进。LBP 是一种用来描述图像局部纹理特征的算子,具有灰度不变性等显著优点,对光照变化较为免疫。对每一个像素点,基本的 LBP 特征值编码计算以该像素处的灰度值为阈值,与其八个邻域像素分别进行比较,较小则记为 0,否则记为 1;获得各自的编码后,再以当前像素左侧的像素为起点逆时针旋转编码,即可得到该像素处的 LBP 特征值编码,如图 2 所示,但由于 LBP 特征是点的局部特征,直接进行对应点匹配稳定性很差^[12]。

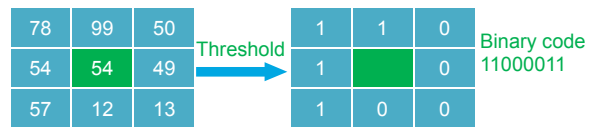


图 2 LBP 特征值编码。
Fig. 2 LBP feature encoding.

本文将 LBP 特征与 Haar-like 特征进行融合,在没有大量增加计算负担下有效地融入原算法框架中:对一帧图像,计算其各像素点处的 LBP 特征值编码,以该编码作为该点处的“灰度值”,生成一幅 LBP 图,并以此 LBP 图为基础,代替原来的灰度图,生成积分图,进而完成后续计算。同时,由于这时 Haar-like 特征是对区域中各像素点的 LBP 特征值求和,避免了点对点匹配 LBP 特征值的不稳定性。图 3 为基于灰度值(gray level, GL)编码的 Haar-like 特征和改进后的基于 LBP 特征值编码的 Haar-like 特征的求解示意图。

2.2.2 融合卡尔曼滤波器的位置校正

针对形貌快速变化的运动目标,本文融入了卡尔曼滤波器进行改进。卡尔曼滤波基于贝叶斯滤波框架,

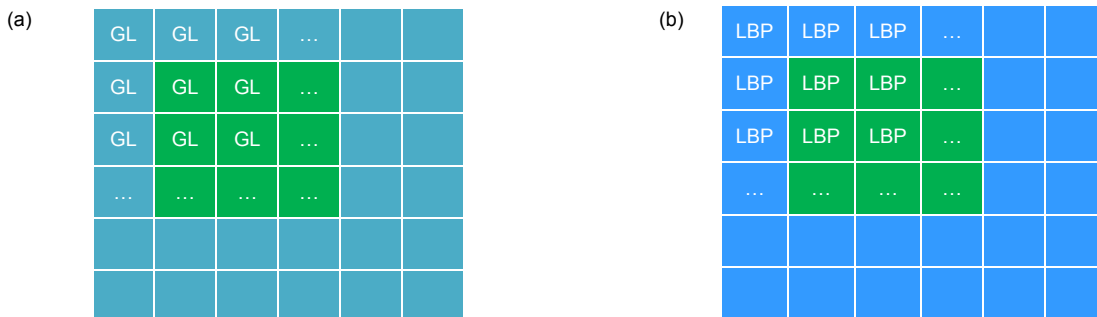


图 3 基于不同编码算法的 Haar-like 特征。(a) 基于灰度值编码的 Haar-like 特征。(b) 基于 LBP 值编码的 Haar-like 特征。

Fig. 3 Haar-like features based on different coding algorithm. (a) Haar-like features based on gray level. (b) Haar-like features based on LBP value.

是一种对线性运动模型、高斯分布的状态变量进行状态估计的有效方法。它是一种递归的估计，即只要获知上一时刻的状态估计值以及当前状态的观测值就可以计算出当前状态的估计值，不需要记录观测或估计的历史信息^[13]。运动目标的位置以及速度可以被线性状态空间描述如下：

$$X_t = (P_x, P_y, V_x, V_y), \quad (6)$$

其中： P_x, P_y, V_x, V_y 分别表示运动目标在 X 轴和 Y 轴方向的位置和速度。

在相邻两帧间隔，将运动目标视为匀速运动，定义系统状态转移矩阵 A 为

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

由系统状态和观测状态的关系，观测矩阵 H 为

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \quad (8)$$

协方差矩阵可以根据实际情况确定。由 Kalman 滤波器的预测，得到帧间目标确定性状态转移矢量为

$$\begin{cases} \Delta X = P_x(t) - P_x(t-1) \\ \Delta Y = P_y(t) - P_y(t-1) \end{cases}. \quad (9)$$

在本文的实现过程中，以上一帧中目标的位置作为先验状态值，由状态转移方程得到当前帧位置的预测值，通过贝叶斯分类器得到当前帧目标位置后，以该位置值作为观测值，通过校正方程对预测值进行校正得到含有目标位置的后验状态值，迭代过程如图 4 所示。

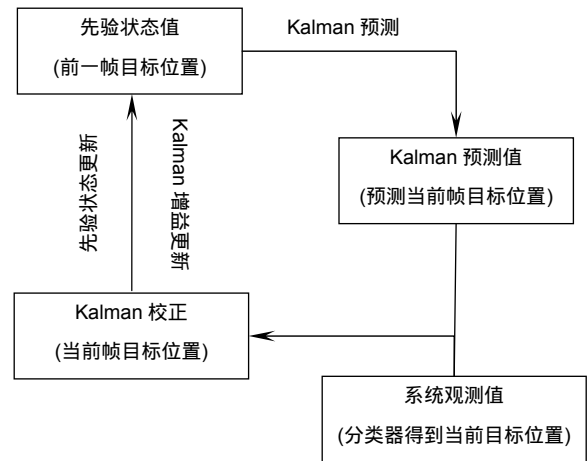


图 4 融合原算法与 Kalman 滤波器。
Fig. 4 Fusion of Kalman filter and original algorithm.

另一方面，在算法实现过程中，可将采样区域以上一帧中的目标位置为中心分为四个象限，根据卡尔曼滤波器的预测值与中心的相对位置，对不同象限进行不同密度的采样，如可设置不同的径向采样迭代步长，从而提高计算效率。

2.2.3 自适应学习速率更新

针对原算法中分类器固定学习速率带来的不足，在本文中，以动态分配的数组记录过去的 N (参数可调) 帧中每一帧的目标置信度并求取平均值，以该平均值为自适应阈值。对当前帧计算的置信度，若其低于当前的自适应阈值，则调低当前的分类器学习速率，反之，赋予当前的分类器较高的学习速率，其伪代码如下所示：

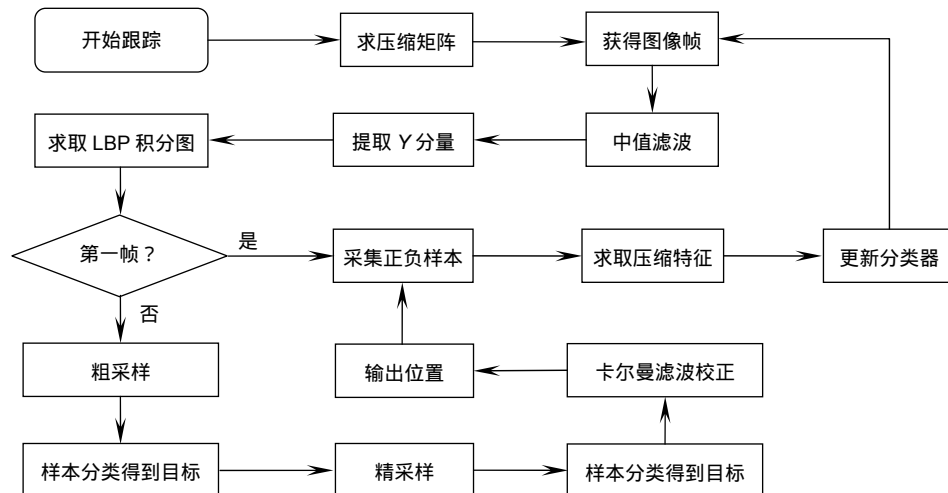


图 5 DSP 算法流程图。

Fig. 5 DSP algorithm flow diagram.

输入:置信度数组 $\{Cfd_i\}_{i=0}^{N-1}$, 当前帧帧序号 $M = (M) \% N$;

- 1) 当前贝叶斯分类器得到的最大样本置信度: R_{es} ;
- 2) 更新置信度数组: $Cfd_M = R_{es}$;
- 3) 初始化置信度和: $Cfs = 0$;
- 4) 求 N 帧置信度之和:

For $k=0$ to $N-1$ do

$Cfs += Cfd[k]$;

- 5) 求当前帧置信度阈值:

$Cf_th = Cfs / N$;

- 6) 更新分类器学习速率 clr :

If $(R_{es} < Ir_th)$ $clr = \text{lower-learning-rate}$;

Else $clr = \text{higher-learning-rate}$;

输出 clr .

通过该步骤设置动态更新的分类器学习速率,能够避免目标在低置信度条件下所带来的参数过快更新的问题,有利于减少跟踪算法的漂移,在目标被短暂遮挡的情况下具有更好的抵抗能力。

融入上述措施后,改进后的压缩域目标跟踪算法流程图如图5所示。

3 嵌入式 DSP 图像处理平台硬件设计

为实现对运动目标的实时识别与跟踪,本文以 DM6437 芯片为核心,结合其达芬奇系统,以视频信号的采集、存储、处理与分析以及交互信号的输入输出为主线进行系统硬件设计^[14]。DM6437 芯片是 TI 生产的 C6000 系列达芬奇数字信号处理器,采用 C64x+

内核与哈佛结构,是一款高速、低功耗的 32 位定点 DSP 芯片,其工作主频高达 600 MHz,采用 8 级流水线,并配备视频处理子系统(VPSS),专门应用于多媒体信号处理。系统从功能上可分为视频信号采集模块、视频信号传输与变换模块、视频处理和分析模块、视频信号存储模块和控制信号交互模块,其结构框图如图6所示。

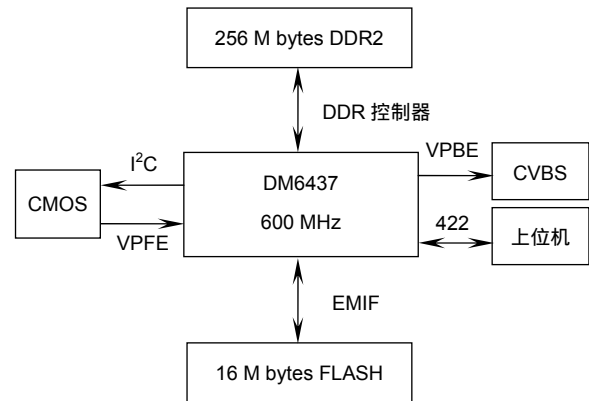


图6 系统硬件结构。

Fig. 6 System hardware structure.

视频信号采集模块采用的 CMOS 数字图像传感器具有 1280×960 的有效成像像素阵列,同时集成了自动曝光控制、片上开窗等功能。CMOS 与 DSP 的 I²C 总线相连,通过接收参数完成配置。摄像头输出的图像信号通过 12 位的数据总线与行、场同步信号、像素时钟与 DM6437 的 CCD 控制器(CCD controller,CCDC)模块进行连接^[15]。

视频传输与变换模块基于 DM6437 VPSS 实现:CCDC 模块从 CMOS 获得一帧图像后(Bayer 类型),交

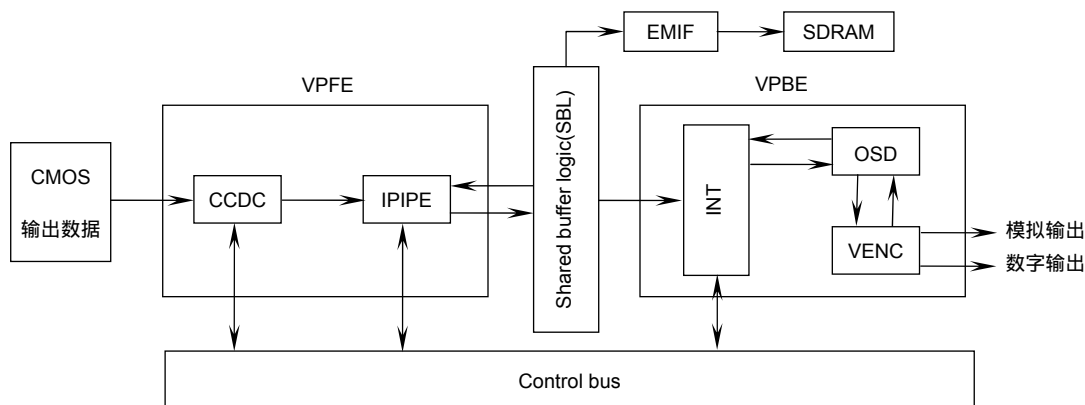


图7 视频采集与显示系统。

Fig. 7 Video capture and display system.

给预览引擎图像管道(IPIPE)模块, 由其对图像进行数据格式变换与一定的预处理。具体地, 通过 CFA 插值, 将 Bayer 格式的 RGB 图像数据转换成 RGB 图像数据, 再进一步完成 RGB 色彩空间到 YCbCr 色彩空间的转换。IPIPE 中 4:2:2 下采样模块可以将图像数据最终转变为便于传输和处理的 YUV 格式; 此外, 按照人为设定的不同亮度、对比度数值对图像进行一定的调整, 如图 7 所示。

视频处理和分析模块基于 DM6437 处理器实现。通过从图像缓存队列中提取图像帧并对其进行处理与计算, 得到目标在每一帧中的位置, 用矩形框标注并通过 CVBS 接口传输到显示器。

视频信号存储模块基于系统外扩存储器实现。DM6437 的 DDR2 映射空间大小为 256 M, 地址范围为 0x80000000—0xFFFFFFFF, 通过专门的外设模块 DDR2 控制器连接外部存储器。本文中系统采用 2 片 64 M×16 bit 的 DDR2, 组合成 32 位数据总线、13 位地址总线作为系统主存与 DDR2 控制器相连, 结合 3 位 BANK 选择信号达到 256 M 的寻址空间。此外, 需要使用 DM6437 外部存储器接口(external memory interface, EMIF)空间来映射非易失性存储器, 以存储程序实现系统上电自举。本文中使 FLASH 容量为 16 M Bytes, 具有 24 位地址总线, 8 位数据总线, 通过连接 CS2 片选映射到 DM6437 0x42000000—0x42FFFFFF 的空间范围内。

4 算法在 DSP 平台的实现与优化

4.1 基于队列的循环图像缓存管理

在视频前端(VPFE)与 CPU 之间维护了两个队列进行图像数据传输, 分别为待填充图像队列(A1)与就

绪队列(A2)。初始化时将 4 个视频帧缓存头地址入队。过程中若 A1 元素不为空, 则将队头处的帧头地址作为当前图像数据传输的目的地址, 通过 EDMA 通道将数据从前端接口向缓存区进行搬移。一帧图像搬运完毕后, 进入 EDMA 中断, 将该帧头地址从 A1 出队, 进入 A2, 此时, CPU 程序监测到就绪队列中存在元素, 将该帧图像提取出来进行处理。

视频后端同样维护了两个队列(B1 与 B2), B1 用于接收 CPU 处理之后的待输出图像, 该队列中元素非空时将触发 EDMA, 通过 EDMA 通道把图像数据从 DDR2 中的缓存区搬移至图像输出接口。B2 用于将完成传输的待输出图像进行显示。之后, 后端的图像帧被传回前端, 实现缓存区的循环利用, 如图 8 所示。

4.2 算法在 DSP 平台的实现

在视频流处理控制框架搭建完成的基础上, 算法使用 C 语言结合 TI 官方经过线性汇编优化的代码库完成, 实现过程中一些注意要点如下:

1) 对每一帧图像进行中值滤波预处理, 从而去除孤立的噪声点, 提高图像信噪比的同时有利于跟踪算法的稳定性。考虑到实时性的要求, 本文中采用 1×5 的一维模板来进行。

2) 对每一帧图像, 每个像素处的积分图数值最大可达 $255 \times 960 \times 960 = 235008000$, 32 位编译器下 short 型数据最大可表示的数值为 65535, 故需利用无符号 32 位整形数组进行积分图存储。

3) 由于 DM6437 为定点处理器, 算法中的浮点数计算如果直接利用 ANSI C 中的运算语句会消耗大量的时钟周期, 需利用 TI 的 IQmath 库来达到速度与精度的权衡^[16], 但许多情况下精度的折损不可避免。其实质是通过把浮点数转化成 Q 格式的定点数进行计

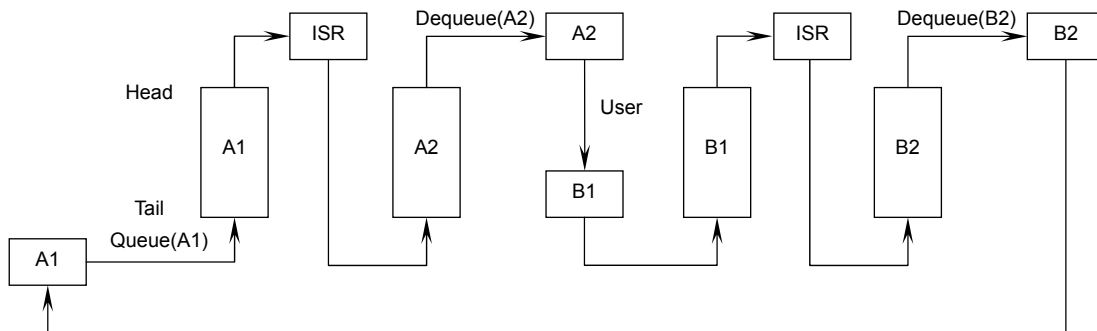


图 8 图像帧缓存管理。

Fig. 8 Image buffers management.

算,每个Q值格式的数据有不同的数值范围与精度。为了充分保证算法整体的精度,在设置全局Q值为9的基础上,对局部关键计算采用局部Q值进行处理以达到更高的精度。

4.3 算法在DSP平台的优化

本文所使用的DSP平台DM6437的DSP内核主频高达600MHz,采用哈佛结构与先进的超长指令字(VLIW)架构,并具备八级流水线与丰富的核内逻辑资源,包括64个32位通用寄存器、6个32位累加器与2个32位硬件乘法器等。采用两级缓存结构,包括80kB的一级数据缓存L1DSRAM、32kB的一级程序缓存L1PSRAM和128kB的二级缓存L2SRAM,均可灵活配置为缓存(cache)或普通的片内存储器^[17]。通过优化,能够充分释放处理器在指令级并行、数据级并行的能力,使视频处理帧率获得有效的提升,达到实时处理的效果。

本文中算法优化采取的主要措施有:

1) 循环展开、多路并行与软件流水

视频、图像处理属于计算密集型、数据密集型的场合,而算法中主要的计算量集中于各个循环体,因此,对循环体运行速度的优化具有至关重要的意义。

软件流水是一种重要的指令调度技术,通过并行执行来自不同循环体的指令来加快循环程序的执行速度,在前一个循环体未结束前启动下一个新的循环体。循环展开将循环体中的指令进行多份复制,减少了循环的重复次数,降低了循环转移中跳转指令、变量状态转移等指令的开销。在软件流水中应用循环展开可以使软件流水实现分数值的启动间距^[18],同时基于展开的优化技术结合多路并行可以降低程序的资源需求和关键路径的长度。

本文的优化过程中,主要通过结合编译器选项、编译器编译输出信息以及手动调整代码的方式来实现更好的软件流水效果与指令级的并行运行。通过编译器选项设置-O2及以上、预处理指令#pragma MUST_ITERATE()的使用,能够将大量一重循环以及内循环为常数次的二重循环展开。

通过编译器设置-k选项,能够获得编译器对循环体的编译信息:对过大的、编译器难以完成软件流水编排的循环体进行分离,通过分离为多个较小的循环体从而实现原大循环体的软件流水;通过restrict关键字的使用,能够消除编译器对不同数据域可能存在依赖性的顾虑,减少了循环携带依赖性(loop carried

dependency)与存储别名(memory aliasing)等造成的依赖性,更加有利于指令的并行编排与运行;通过对内核逻辑资源(累加器、乘法器等)使用情况的观察,对循环体内部数据操作进行适当的调整(如乘法器使用过载、而加法器有所空闲的情况下,可以把部分乘法操作转换为加法操作),能够更加充分合理地利用处理器硬件资源,获得更佳的指令级并行效果。经过这一步全局性的优化,算法消耗时钟数由初始的98MHz降低至51MHz,速度将近提升了一倍,效果较为显著。

2) VLIB代码库的使用

VLIB是TI提供的针对C6x内核的视觉代码库,经过汇编级的深度优化,合理使用能够节省大量的时钟周期数^[19]。在本文的算法中,对每一帧图像都需要计算其全图积分图,由于全图循环以及其存储基于32位的大位宽数据,耗时较为严重。而利用VLIB下的函数接口VLIB_integralImage8能够对像素数据为8位的灰度图像实现高效的全图积分图计算,相比原始代码缩减了高达2/3的运算时间。

3) 存储结构的优化

DM6437具有二级片内存储结构,如图9所示,其各级存储参数在本节开头已经进行了描述。本文在算法实现与优化的过程中,通过实验在不同缓存参数配置下的效果进行对比,最终确定设置L1P为32kB cache,L1D为32kB的SRAM和48kB cache,L2设置为80kB的cache和48kB的SRAM。同时,需要通过MAR寄存器设置访问频繁的程序段和DDR2上的大数据量数组为可被缓存、代码编写过程中注意局部性问题、堆上分配内存与cache line大小相符等措施来充分利用其缓存机制。此外,为充分利用片内SRAM,将频繁使用的数据结构如压缩矩阵、动态分配的堆数据等放置在片内,减少大量访存时间。

4) 针对数据级并行进行的优化

数据级并行优化主要利用的是处理器的单指令流多数据流(SIMD)技术,可以在一条单独的指令周期中同时执行多个运算操作,从而增加处理器的吞吐量,有效提高数据密集型应用场合下的系统处理效率。对于本文中使用的DSP处理器,主要通过两种途径来进行这一优化:第一种是在遵循尽量使用小位宽数据原则的基础上,通过手动对数据域按照处理器要求进行对齐的基础上,设置一定的编译选项(-O2及以上)后,由编译器自动生成SIMD处理的指令流;第二种方法是完全手动地、显式地使用该DSP提供的SIMD指令

集相关指令完成代码逻辑,具体地,在该 DSP 中,使用提供的内联函数集(intrinsics)通过数据打包实现在单个指令周期内的多个数据同时操作,例如通过 `_amem4_const(Uint8*)` 可以一次获取 4 个 8 位的无符号整型数据,并以此为基础,利用 `add2(Uint32*, Uint32*)` 指令可以一次完成 4 个 8 位无符号整型数据相加。出于开发效率的考虑,实现过程中以第一种方法为主。

此外,本文在实现过程中使用了移位运算代替一些乘除法运算,对计算每个样本压缩特征的函数进行内联以防止流水线被频繁打断并且减少函数调用开销

等。经过这些优化措施,算法速度实现了明显的提升,在样本选取数量特别是粗采样样本数量较为充分的情况下,算法处理速度能够达到 25 f/s,满足实时性处理要求。

优化前后的性能对比如图 10 所示,其中,计算积分图即对分辨率为 960×960 的 8 位数据灰度图计算全图积分图,样本分类计算包括对样本在压缩域特征的提取、利用分类器对其进行分类等,其他操作包括计算 LBP 特征图、抽取正负样本、分类器参数更新等。

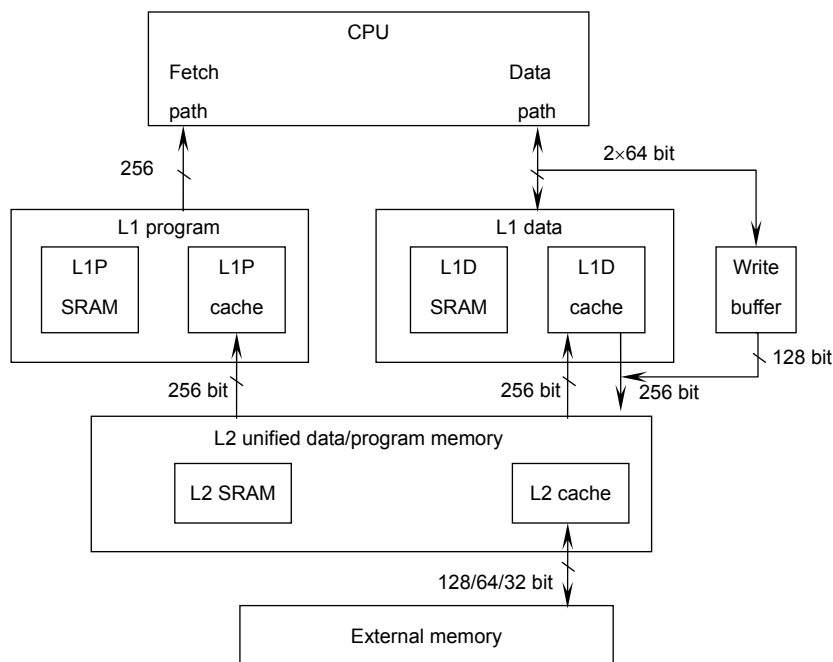


图 9 DM6437 片内存储结构图。

Fig. 9 DM6437 on-chip memory structure.

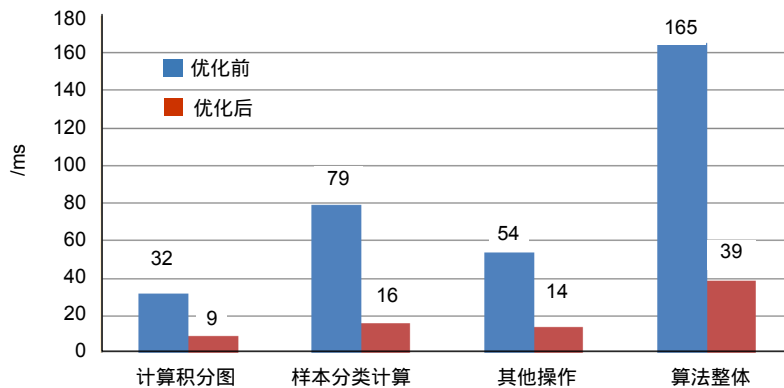


图 10 优化前后各部分耗时对比图。

Fig. 10 Time contrast sections consume.

5 系统测试与实验结果

5.1 算法在 PC 平台的仿真验证

在 PC 平台对基于压缩感知的压缩域目标跟踪算法的改进措施进行了仿真验证(对彩色图像测试序列,先将其转换为灰度图像进行处理,这与原算法及 DSP 平台的处理方法是一致的,之后显示结果仍叠加在原彩色图像上),结果如图 11~图 15 所示(对相同帧序号的图像,左侧图像为改进之前的算法效果,右侧图像为改进之后的算法效果,对图 15, FCT 为改进之前的算法, FCT Kalman 为改进之后的算法)。其中,图 11 中的 David-indoor 序列主要是在目标表面光照迅速变化的场合下对改进算法的稳定性进行测试;图 12 中的

Kite-surf 序列主要针对目标形貌迅速变化以及快速运动的场景中对改进算法的稳定性进行测试;图 13 中的 Box 序列主要对目标在被短暂遮挡的情况下对改进算法的鲁棒性进行测试;图 14 中的 coke11 序列则在上述因素同时存在的情况下对改进后跟踪方法的鲁棒性进行了测试;图 15 的测试利用了 Kite-surf 序列和 bolt 序列的已有标注结果,对算法改进前后每帧的跟踪误差进行了对比。

由上述测试序列的仿真结果可以得到:

1) 由 David-indoor 序列的测试结果可知,在面对光照条件变化迅速的环境时,改进后的融合了 LBP 与 Haar-like 特征的方法具有更好的鲁棒性。

2) 由 Kite-surf 序列与 bolt 序列的仿真测试结果可

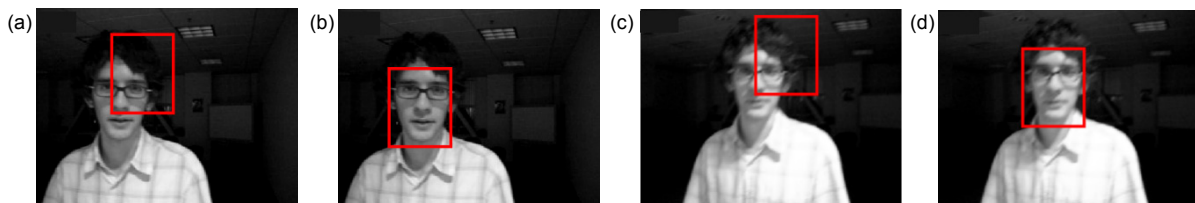


图 11 David-indoor 序列测试结果. (a) 改进前第 313 帧. (b) 改进后第 313 帧. (c) 改进前第 329 帧. (d) 改进后第 329 帧.

Fig. 11 David-indoor sequence test result. (a), (c): The 313th frame (a) and the 329th frame (c) before algorithm improvement. (b), (d): The 313th frame (b) and the 329th frame (d) after algorithm improvement.

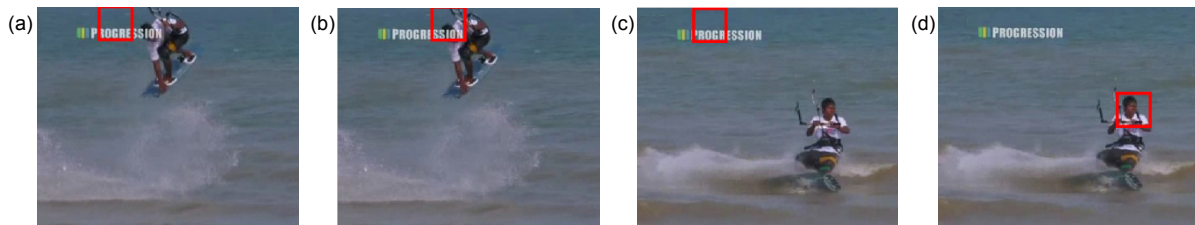


图 12 Kite-surf 序列测试结果. (a) 改进前第 38 帧. (b) 改进后第 38 帧. (c) 改进前第 65 帧. (d) 改进后第 65 帧.

Fig. 12 Kite-surf sequence test result. (a), (c): The 38th frame (a) and the 65th frame (c) before algorithm improvement. (b), (d): The 38th frame (b) and the 65th frame (d) after algorithm improvement.

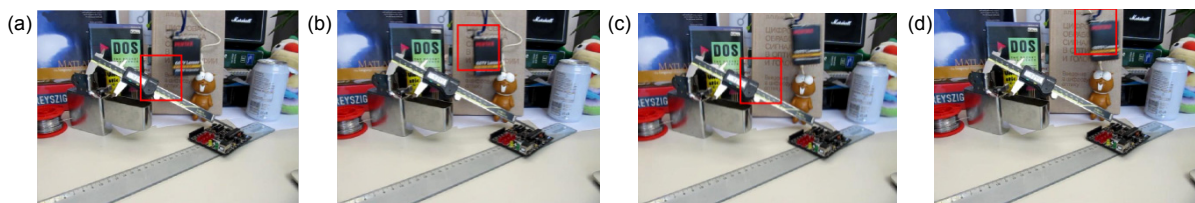


图 13 Box 序列测试结果. (a) 改进前第 336 帧. (b) 改进后第 336 帧. (c) 改进前第 344 帧. (d) 改进后第 344 帧.

Fig. 13 Box sequence test result. (a), (c): The 336th frame (a) and the 344th frame (c) before algorithm improvement. (b), (d): The 336th frame (b) and the 344th frame (d) after algorithm improvement.

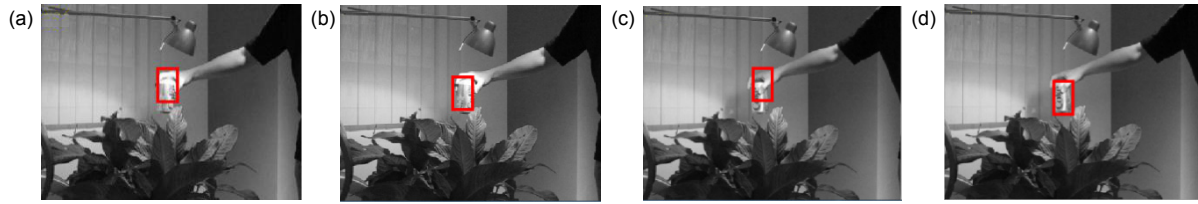


图 14 Coke11 序列测试结果. (a) 改进前第 24 帧. (b) 改进后第 24 帧. (c) 改进前第 85 帧. (d) 改进后第 85 帧.
Fig. 14 Coke11 sequence test result. (a), (c): The 24th frame (a) and the 85th frame (c) before algorithm improvement. (b), (d): The 24th frame (b) and the 85th frame (d) after algorithm improvement.

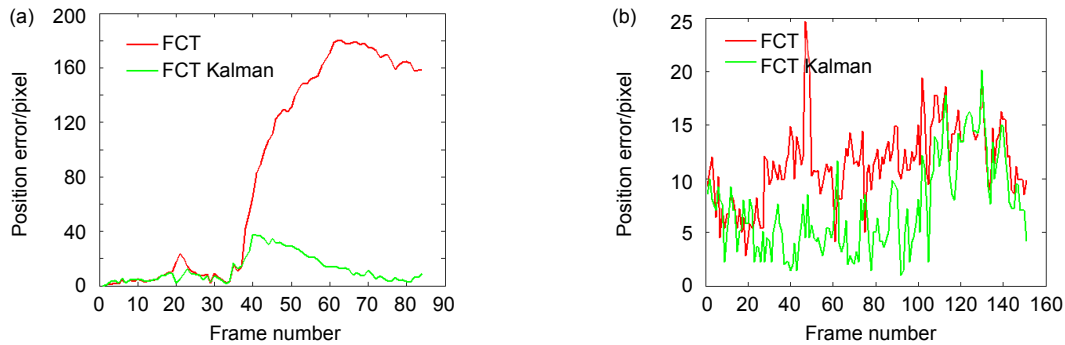


图 15 算法改进前后目标跟踪误差比较. (a) Kite-surf 序列测试结果. (b) Bolt 序列测试结果.
Fig. 15 Comparison of target tracking errors (before and after algorithm improvement). (a) Kite-surf sequence test result. (b) Bolt sequence test result.

知融合了卡尔曼滤波器的改进方法对目标物体的快速运动与形貌变化有了更好的跟踪鲁棒性与精度。

3) 由 Box 序列的仿真测试结果可知, 添加了自适应学习速率更新策略之后的跟踪方法对目标短期遮挡后轻易丢失的缺点有了较好的改进。

4) 通过对 coke11 序列的测试结果可知, 在光照变化、目标形貌变化和短期遮挡同时存在的情况下, 改进后的方法也能够取得较好的跟踪效果。

5.2 DSP 目标跟踪系统测试与结果

实验过程中, 为适配显示器分辨率, 输出 720×480 的灰度图像进行观察。初始目标框设置为图像中心位

置 80×80 像素区域, 以目标为中心, 采集以 2 像素为步长、半径 15 像素以内的样本作为粗采样正样本, 半径范围介于 16 到 18 像素之间的样本作为粗采样负样本; 采集以 1 像素为步长、半径 4 像素以内的样本作为精采样正样本, 半径范围介于 5 到 8 像素之间的样本作为精采样负样本, 学习速率初始化为 0.7。在室内进行实验, 通过 CCS3.3 读取输出图像缓存区的图像数据(因显示图像分辨率要求, 其大小为 720×480)并进行显示, 实验结果如图 16 所示。

初始时将待跟踪的目标物体置于视频图像中心的位置, 之后通过串口向跟踪系统发送开始跟踪指令, 跟踪算法开始对目标物体进行持续地跟踪、锁定。目



图 16 DSP 端算法实验结果.
Fig. 16 DSP algorithm experiment result.

标物体在视频图像中自左向右移动的过程中,通过物体的旋转改变其形貌、物体姿态的变化改变其光照情况以及物体移动速度的变化来测试跟踪系统的跟踪性能。实验结果显示算法能够实现稳定的跟踪,改进后的方法移植到 DSP 系统中后对形貌变化、光照变化和快速运动都具有了较好的适应能力。通过 CCS3.3 的性能剖析工具测得处理每帧图像消耗的时钟数为 24 MHz,即大约耗时 39 ms,处理帧率可达 25 f/s。说明改进后的算法在 DSP 端同时具有较好的实时性,能够满足特定应用场合下的视频目标实时跟踪需求。

6 结论

本文针对小型化嵌入式图像跟踪系统的需求,设计并实现了基于 DM6437 的目标跟踪图像处理平台,并在该平台上实现了改进后的基于压缩感知理论的目标跟踪算法。仿真和实验结果表明,经过对算法的改进与优化实现,算法能够在跟踪鲁棒性、精度和实时性上满足一些应用条件下的目标跟踪需求,具有一定的实用价值和参考意义。今后的工作将主要集中在复杂背景下跟踪算法的研究、目标丢失处理策略、目标模型更新的策略优化及其在嵌入式系统中的实际应用等。

基金项目

国家自然科学基金资助项目(51575388)。

参考文献

- Jia Guimin. Research of target tracking based on local information[D]. Tianjin: Tianjin University, 2008.
贾桂敏. 基于物体局部信息的跟踪算法研究[D]. 天津: 天津大学, 2008.
- Wang Rui, Wang Lin, Jiang Zhiwei. Active visual system for moving object intelligent tracking based on DSP[J]. *Opto-Electronic Engineering*, 2009, **36**(2): 6–10.
王睿, 王林, 姜志威. 基于 DSP 的主动视觉运动目标跟踪策略及实现[J]. *光电工程*, 2009, **36**(2): 6–10.
- Grabner H, Grabner M, Bischof H. Real-time tracking via on-line boosting[C]// *Proceedings of British Machine Vision Conference*, 2006, **1**: 6.
- Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(7): 1409–1422.
- Zhang Kaihua, Zhang Lei, Yang M H. Real-time compressive tracking[C]// *Proceedings of the 12th European Conference on Computer Vision*, 2012.
- Li Qingwu, Zhu Guoqing, Zhou Yan, et al. Object compressive tracking via online feature selection[J]. *Acta Automatica Sinica*, 2015, **41**(11): 1961–1970.
李庆武, 朱国庆, 周妍, 等. 基于特征在线选择的目标压缩跟踪算法[J]. *自动化学报*, 2015, **41**(11): 1961–1970.
- Liu Qingshan, Yang Jing, Zhang Kaihua, et al. Adaptive compressive tracking via online vector boosting feature selection[J]. *IEEE Transactions on Cybernetics*, 2015, *PP*(99): 1–13, doi: 10.1109/TCYB.2016.2606512. (in Press)
- Chan Sixian, Zhou Xiaolong, Li Junwei, et al. Adaptive compressive tracking based on locality sensitive histograms[J]. *Pattern Recognition*, 2017, **72**: 517–531.
- Yao Xuan, Zhou Yue. Real-time compressive tracking with a particle filter framework[C]// *International Conference on Neural Information Processing*, Cham, 2014: 242–249.
- Johnson W B, Lindenstrauss J. Extensions of Lipschitz mappings into a Hilbert space[J]. *Contemporary Mathematics*, 1984, **26**: 189–206
- Diaconis P, Freedman D. Asymptotics of graphical projection pursuit[J]. *The Annals of Statistics*, 1984, **12**(3): 793–815.
- Pan Qiuping, Yang Wankou, Sun Changyin. License plate detection algorithm based on Haar and MB-LBP features[J]. *Journal of Southeast University (Natural Science Edition)*, 2012, **42**(Z1): 74–77.
潘秋萍, 杨万扣, 孙长银. 基于 Haar 与 MB-LBP 特征的车牌检测算法[J]. *东南大学学报(自然科学版)*, 2012, **42**(Z1): 74–77.
- Fang Wentao, Wang Xiangjun, Tang Qijian. Object tracking system for MUAUV based on particle filter[J]. *Laser & Infrared*, 2012, **42**(7): 841–844.
房文涛, 王向军, 汤其剑. 基于粒子滤波的机载目标跟踪系统设计[J]. *激光与红外*, 2012, **42**(7): 841–844.
- Liu Xiaoning, Chen Xiaodong, Yu Daoyin. Design of moving target recognition and tracking system based on DSP[J]. *Video Application & Project*, 2010, **34**(11): 107–110.
刘小宁, 陈晓冬, 郁道银. 基于 DSP 的运动目标识别与跟踪系统的设计[J]. *电视技术*, 2010, **34**(11): 107–110.
- Feng Yu, Wang Xiangjun, Chen Wenliang. DSP implementation of relative position and attitude calculation[J]. *Chinese Journal of Sensors and Actuators*, 2016, **29**(1): 35–39.
冯禹, 王向军, 陈文亮. 相对位姿测量解算的 DSP 实现[J]. *传感技术学报*, 2016, **29**(1): 35–39.
- Peng Zhiming, Li Lin. Fixed-point DSP algorithm based on IQmath library[J]. *Microcontrollers & Embedded System*, 2010(9): 39–41.
彭志明, 李琳. 基于 IQmath 库的定点 DSP 算法设计[J]. *单片机与嵌入式系统应用*, 2010(9): 39–41.
- Zhu Wei, Han Jufeng, Zheng Yayu, et al. Real-time detection of multi-object based on DSP processor for panoramic video[J]. *Opto-Electronic Engineering*, 2014, **41**(5): 68–76.
朱威, 韩巨峰, 郑雅羽, 等. 基于 DSP 的全景视频多目标实时检测[J]. *光电工程*, 2014, **41**(5): 68–76.
- Li Wenlong, Liu Li, Tang Zhizhong. Loop unrolling optimization for software pipelining[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2004, **30**(11): 1111–1115.
李文龙, 刘利, 汤志忠. 软件流水中的循环展开优化[J]. *北京航空航天大学学报*, 2004, **30**(11): 1111–1115.
- Zhu Zhongke. Optimized for vehicle wheelset wear detection algorithm based on DM6437[D]. Hangzhou: Hangzhou Dianzi University, 2013.
祝中科. 基于 DM6467 的车轮对磨损检测算法的优化实现[D]. 杭州: 杭州电子科技大学, 2013.