

基于兴趣度的协作过滤技术研究

陈永平^{1,2}, 苏新², 毛万胜²

(1. 合肥工业大学, 安徽 合肥 230009; 2. 安徽工业大学, 安徽 马鞍山 243002)

摘要:随着互联网上的信息的迅速增长,协作过滤技术得到越来越广泛的应用。结合了显式和隐式计算兴趣度的方法,提出了一种新的计算用户兴趣度的方法。并在此基础上论述了基于兴趣度的协作过滤技术。该方法通过寻找相似用户群体,由相似用户群体来预测用户对某一WEB事务的喜好,并给出了相应的实现算法。实验结果表明,该方法能提供较好的协作推荐服务。

关键词:兴趣度;协作过滤;WEB挖掘;显式方式;隐式方式

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2008)01-0106-05

Research of Collaborative Filtering Technology Based on Interest

CHEN Yong-ping^{1,2}, SU Xin², MAO Wan-sheng²

(1. Hefei University of Technology, Hefei 230009, China;

2. Anhui University of Technology, Ma'anshan 243002, China)

Abstract: As the rapidly growing of information on Internet, collaborative filtering techniques have been more and more widely applied. Combines explicit and implicit computer users' degree of interest method together, putting forward a new method for user interest computing. And analyse the collaborative filtering technology based on the degree of interest. This method proves to find similar users group which can be used to expect the users' likes to a certain WEB item, this provides the relevant calculation method. Experiment result makes shown that this method can provide a better collaborative filtering recommendation service.

Key words: interest; collaborative filtering; WEB mining; explicit; implicit

0 引言

随着信息技术的高速发展和因特网络的不断普及,Internet上的信息资源呈指数增长。面对海量般的网上资源,人们往往感到无所适从,出现了所谓的“信息过载”和“信息迷向”现象,即人们很难找到自己感兴趣的信息,即使找到了一些,也经常混有很多“噪音”。为了帮助用户根据个人兴趣爱好,自动在网上查找相关的信息,屏蔽与用户毫不相关的信息,研究人员研究了一种用于解决“信息过载”的系统,即协作过滤系统。协作过滤系统解决信息过载问题是基于其它用户对一系列相关文档的观点而向用户作出推荐^[1]。

1 兴趣度

1.1 兴趣度的定义

兴趣度就是用户对某一网络结点或某一WEB事

务的兴趣强度弱。它反映了用户对网络结点或WEB事务的喜爱程度。可以通过用户对某一事务的浏览行为来计算和表示用户的兴趣度。目前,用来计算和表示用户的兴趣度的方式有两种:即显式方式和隐式方式^[2]。

1.2 兴趣度的计算和表示

显式方式:这种方式要求用户在浏览某一WEB事务时,根据其个人兴趣给出相应的评分,有三分制(若用户给出3分,则表示其对该事务非常感兴趣,2分表示对该事务的兴趣一般,1分表示其对该事务不感兴趣或兴趣很小),也有五分制(5分表示用户对其浏览的事务有很大的兴趣,4分的兴趣次之,3分表示用户对该事务的兴趣一般,2分表示用户对其浏览的事务兴趣很小,而1分表示用户对该事务不感兴趣)。根据用户的评分来判断其对浏览的WEB事务兴趣程度。显然这种在用户的参与下来表示用户对某一事务的兴趣是比较准确的,但是它要求用户的参与,增加了用户的负担,所以不能被一些用户所接受。

隐式方式:这种方式主要是通过用户在网上的浏览行为来计算用户的兴趣度。它不需要用户直接参

收稿日期:2007-03-23

基金项目:安徽省自然科学基金资助项目(KJ2007B245)

作者简介:陈永平(1969-),男,安徽合肥人,讲师,研究方向为数据挖掘。

与,是目前常用的计算用户兴趣的方式,但它所计算出的兴趣只能较准确表示用户的兴趣度。众所周知,用户上网时对自己感兴趣 WEB 事务通常会用较多的时间去浏览或者保存和下载到自己的计算机,也有可能通过打印机打印下来,若对自己浏览的事务不感兴趣可能用较少的时间去浏览。因此可能挖掘用户的浏览行为来计算出用户的兴趣度。

目前,用于计算用户兴趣度的公式比较多,主要如下:

公式 1:定义 USER 对事务页 PAGE 的兴趣度 $F(\text{USER}, \text{PAGE})$ 如下^[3]:

$$F(\text{USER}, \text{PAGE}) = \frac{\text{user 浏览 page 所用的时间}}{\text{user 的总浏览时间}} \times \frac{\text{page 字节数}}{\text{路径中总字节数}}$$

公式 2:定义 USER 对事务页 PAGE 的兴趣度 $F(\text{USER}, \text{PAGE})$ 如下^[1]:

$$F(\text{USER}, \text{PAGE}) = \frac{\text{user 浏览 page 所用的时间}}{\text{page 中字符数目}} \times \text{路径因子}$$

其中路径因子可以根据 PAGE 在用户会话有意义的访问路径 P 中的深度设定。例如:若 PAGE 是 P 的终点,则 PAGE 的路径因子设为大于 1 的值,否则,设为 1。

公式 3: $F(\text{USER}, \text{PAGE})$ 为用户 USER 对 WEB 事务页 PAGE 的兴趣度,用多元线性回归模型来描述用户兴趣与用户浏览 WEB 事务页时间和翻页 / 拉动滚动条次数(浏览时间和拉动滚动条次数为用户浏览兴趣事务页时的两种主要行为),其公式如下^[4]:

$$F(\text{USER}, \text{PAGE}) = AX_1 + BX_2 + C$$

其中: X_1 表示浏览时间, X_2 表示翻页 / 拉动滚动条次数, A, B, C 为一组常数(站点类型不同有不同的值,为一经验值)。

以上三公式都能较好地计算出用户的兴趣度。但也有不足之处。用户对某一事务页感兴趣与其浏览时间有关,可它们之间不一定存在正比例关系。有时由于网络阻塞等原因或用户打开事务页后去做别的事情,这时用户浏览事务页所用时间就会很长,按照以上三公式计算的兴趣度就会很大,所以应该设定一个浏览事务页时间的阈值,当浏览时间超过这一阈值,兴趣度就相应减小或为零。有时用户在浏览到自己喜爱的事务页可能会把该事务页下载或保存到自己计算机的存储器里,或者通过打印机打印出来,以上两公式在这方面没有反映。另外,不同的用户阅读速度有快有慢,他们对同一兴趣网页浏览时间有长有短,而在上面的三公式都没有较好地表示出来。

1.3 显式和隐式相结合的兴趣度的计算方法

由于计算用户兴趣度的显式方式和隐式方式各自的不足,文中采用了将两种方式相结合的方式来计算用户的兴趣度。其具体方法如下:

若用户浏览 WEB 事务页时给予评分了,则以用户的评分作为兴趣度的值,同时进行归一化处理,即用用户的评分减去 1 后除以该种评分制的最大值减去 1 的值(三分制的除以 2,五分制的除以 4)其计算公式如下:

$$F(\text{USER}, \text{PAGE}) = (\text{用户的评分} - 1) / (\text{该分制的最大值} - 1) \quad (1)$$

若用户浏览事务页时没有给予评分,则用隐显方式计算用户的兴趣度。另外因上述关于隐式方式计算兴趣度的缺陷,定义了一个时间区间 $[\min, \max]$ (\min 表示对某事务页感兴趣时浏览所用最短时间,它可以用事务页的总字符数除以阅读速度最快的人每分钟阅读的字符数的值表示。 \max 表示对某事务页感兴趣时浏览所用的最长时间,它可以用事务页的总字符数除以阅读速度最慢的人每分钟阅读的字符数的值表示),当用户浏览该事务页时间 T 若在 $[\min, \max]$ 范围内,则 $F(\text{USER}, \text{PAGE})$ 最大,文中为 1。当用户浏览时间 T 大于零而小于 \min , $F(\text{USER}, \text{PAGE})$ 随着时间增长而增长,当用户浏览时间 T 大于 \max 而小于 $2 * \max$ ($2 * \max$ 为文中设置的浏览时间阈值)时, $F(\text{USER}, \text{PAGE})$ 随着时间的增长而减小,当 T 大于等于 $2 * \max$ 时 $F(\text{USER}, \text{PAGE})$ 等于零。另外当用户下载、保存或打印该事务页时, $F(\text{USER}, \text{PAGE})$ 等于 1。因此可得到如下计算兴趣度的计算公式:

$$F(\text{user}, \text{page}) = \begin{cases} 1 & \text{用户打印、保存或下载所浏览的事务页} \\ 1 & T \in [\min, \max] \\ \frac{T}{\min} & T < \min \\ \frac{T - \max}{\max} & \max < T < 2 * \max \\ 0 & T \geq 2 * \max \end{cases} \quad (2)$$

综合式(1)和(2)可以得如下的计算兴趣度的计算公式:

$$F(\text{user}, \text{page}) = \begin{cases} 1 & \text{用户打印、保存或下载所浏览的事务页} \\ 1 & T \in [\min, \max] \\ \frac{T}{\min} & T < \min \\ \frac{T - \max}{\max} & \max < T \leq 2 * \max \\ 0 & T > 2 * \max \\ \frac{\text{用户评分} - 1}{\text{该分制的最大值} - 1} & \end{cases} \quad (3)$$

1.4 计算兴趣度的算法

根据 1.3 节可以得到以下关于计算兴趣度的算法。在这里用户若下载、保存或打印该 WEB 事务页统一用时间表示,其值为 min。

INPUT: 用户浏览事务页的浏览行为(可能是评分,用 x 表示用户的评分值,若其值为 0 表示没评分。或浏览时间,用 browtime 表示,下载、打印、保存时用 browtime 表示,并且其值为 min)。

OUTPUT: 用户浏览 WEB 页面的兴趣度。

FUNCTION PAGEINTEREST(browtime, x)

Begin

IF ($x \neq 0$) $F(\text{USER}, \text{PAGE}) = (x - 1) / (\text{该分制的最大值} - 1)$

ELSE

Begin

IF ($\text{browtime} > 0$ and $\text{browtime} < \min$)

$F(\text{USER}, \text{PAGE}) = \text{browtime} / \min$;

IF ($\text{browtime} \geq \min$ and $\text{browtime} \leq \max$)

$F(\text{USER}, \text{PAGE}) = 1$;

IF ($\text{browtime} \geq \max$ and $\text{browtime} < 2 * \max$)

$F(\text{USER}, \text{PAGE}) = (\text{browtime} - \max) / \max$;

IF ($\text{browtime} \geq 2 * \max$)

$F(\text{USER}, \text{PAGE}) = 0$; ($2 * \max$ 为浏览该网页的时间阈

值)

End

End

2 协作过滤技术

2.1 协作过滤的算法介绍

输入: 协作过滤算法的输入是用户的兴趣度值,用于表达用户对项目(WEB 页面等)的喜好。兴趣度的值是通过前面公式(3)计算得到。

处理: 协作过滤算法的目的是向人们推荐新事务或依据用户以前的喜好和其它与该用户有相同喜好用用户的观点为某一用户推荐相应的项目。

输出: 协作过滤算法的输出是项目的建议。它可以是以下两种方式^[5]:

(1) 预测: 通过计算得出用户 c_i 对项目 p_j 的兴趣度。

(2) 推荐: 一系列用户最喜爱的项目列表 $TP_r = \{T_{p1}, T_{p2}, \dots, T_{pm}\}$ 推荐的项目列表, 一般来说包括用户没有浏览过的项目。协作过滤算法的这种输出方式也称为 Top - N 推荐。

协作过滤算法把 $m * n$ 用户一项目数据表示成一个兴趣矩阵 A , 如下所示:

$$\begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ W_{m1} & W_{m2} & \cdots & W_{mn} \end{bmatrix}$$

矩阵 A 中的每一个元素 W_{ij} 是用用户 i 对项目 j 的兴趣度的值, 其值由上文的公式(3)计算得到。

协作过滤算法: 自动搜索整个用户数据库, 寻找当前用户的近邻用户(即那些在过去与当前用户表示出相近兴趣的用户群体), 并利用他们的喜好向当前用户进行推荐。直观地, 这一过程主要可以分为三步^[6]:

① 首先定义一个距离准则, 用于度量用户之间的相似性;

② 其次, 根据用户之间的相似性选取相似邻居集;

③ 最后, 按特定的方法, 综合邻居集中用户对目标项目的兴趣, 形成系统对目标用户兴趣的预测。

在协作过滤算法中, 通过对邻居集中的用户对某一项的兴趣度值, 以及相关的一系列权重来对目标用户的兴趣进行预测。假设目标用户 c 对项目 p 的预测兴趣为 Intr , 则

$$\text{Intr} = p_i \times \delta \quad (4)$$

其中: p_i 为用户 i 对项目 p 的兴趣度值, δ 为用户 i 和用户 c 的相似度的值。

2.2 协作过滤的实现

基于协作过滤的算法使用统计方法找出一系列用户, 称为邻居, 他们与目标用户有相类似的兴趣, 他们对不同的项目的喜好相似。一旦找到目标用户的相似用户群体, 系统使用不同的算法把邻居的喜好结合在一起为目标用户产生预测。将整个过程分为三个部分: 用户一项目数据库的表示、邻居的形成、推荐的产生^[5]。

(1) 获得用户的信息并形成用户数据库。

协作过滤算法的输入数据是 m 个用户对 n 个项目兴趣的集合。它通常被表示成一个 $m \times n$ 用户一项目矩阵 A :

$$\begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ W_{m1} & W_{m2} & \cdots & W_{mn} \end{bmatrix}$$

矩阵 A 中的每一个元素 W_{ij} 是用用户 i 对项目 j 的兴趣度的值, 其值由上文的公式(3)计算得到。其值若为 0 表示用户还没有浏览该项目。

(2) 邻居的形成。

基于协作过滤的推荐系统中最重要的一步是计算用户之间的相似性, 然后使用它来形成一个与目标用户有相似喜好的邻居集。邻居集形成的主要目的是: 对每个用户 c 找到一系列 L 个用户 $N = \{N_1, N_2, \dots, N_L\}$, 其中 $c \notin N$, 而且 c 和 N_1 的相似性最大, c 和 N_2 的相似性次之, 依此类推。用户之间的相似性计算及据此形成一个邻居集可以以不同的方式进行。目前

用于计算兴趣相同相似用户群体的方法有多种,文中采用目前常用的向量空间中两个向量之间的夹角余弦来得到相似度计算方法,其计算公式如下:

$$\cos(u_i, u_j) = \frac{|u_i \cdot u_j|}{||u_i|| \times ||u_j||} \quad (5)$$

上式中的 \cdot 表示两向量的点积,而两用户 u_i, u_j 被看做向量空间中的两个向量,可以通过计算两个向量的夹角的余弦来衡量用户之间的相似度,夹角越小,相似度越高。通过计算得到一个 $m \times m$ 相似矩阵 T , T 中的第 i 行表示第 i 个用户与所有其它用户的相似性,矩阵 T 中的对角线元素的设置成“0”,表示用户不会被选做自己的邻居。

在计算了矩阵 T 里的 $m \times m$ 相似性后,就可以形成相似用户群体(邻近区域)。相似用户群体的形成可以有多种方法,常用的有:基于中心、聚集邻居等方法。文中用基于中心方法形成相似用户群体。这种方法形成一个 d 个大小的邻近区域。即:对于一个用户 u , d 个与该用户最接近的用户被选为邻居。

(3) 预测和推荐。

这一步主要工作就是产生预测和进行推荐,产生预测就是通过相关相似性形成目标用户 c 的相似用户群体(邻居集) N ,它的大小为 d ,用户 c 对项目 p 的预测兴趣度的值 $Intr$ 可通过下面的公式计算得到:

$$Intr = p_i \times \delta$$

其中: p_i 为用户 i 对项目 p 的兴趣度值, δ 为用户 i 和用户 c 的相似度的值。

产生推荐就是统计最近邻居用户集中的每一个用户 i 对不同项目的兴趣度,可以用访问频率或兴趣度大小来衡量,取其中 N 个排在最前面,并且当前用户还没有浏览的项目作为Top- N 推荐集用来进行推荐。

2.3 协作过滤的实现算法

根据上面介绍的协作过滤的实现,可以用以下的算法实现。本算法采用基于中心的方法形成相似用户群体,其中 x 是相似度的阈值(其值一般是由专家制定的)^[5]。

输入: m 个用户和 n 个项目(假设第一个用户 u_1 为当前用户)。

输出:为当前用户将要推荐的项目(即为第一个用户将要推荐的项目)。

FUNCTION COLLABORATIVE(x)

Begin

For ($i=1; i \leq m; i++$)

For ($j=1; j \leq n; j++$)

$W_{ij} = \text{FUNCTION PAGEINTEREST}(\text{browtime}, x);$ //调用计算兴趣度函数计算第 i 个用户对第 j 个项目的兴趣度值放入 W_{ij}

中//

For($i=2; i \leq m; i++$)

If($\cos(u_1, u_i) > x$)

begin

$B_i.\text{xiangsi} = \cos(u_1, u_i); B_i.\text{yonghu} = i;$

end //计算当前用户 u_1 和其他用户 u_i 的相似度的值,然后存入数组 B 中,同时将用户 u_i 的信息也存入 A_i 中,这样数组 A 中的每个元素既有和用户1的相似度值,也包含用户 i 的信息//

For($i=1; i < m-1; i++$)

For($j=i+1; j \leq m; j++$)

对数组 B 按相似度值的按升序进行排序;

$J=0;$

For($i=1; i \leq n; i++$)

If ($W_{ii}=0$)

begin

$K = C_1.\text{yonghu};$

$C_j.\text{val} = W_{jk} * C_j.\text{xiangsi}; C_j.\text{xiangmu} = i;$ // $C_j.\text{val}$ 存放的是兴趣度的值, $C_j.\text{xiangmu}$ 存入是项目 i

$J=J+1;$

End//计算当前用户对还没有浏览的项目的预测兴趣度,并存入数组 C 中//

对数组 C 按从大到小进行排序;

输出 $C_1.\text{xiangmu}$ // $C_1.\text{xiangmu}$ 就是向当前用户推荐的最想要的下一个项目//

End

3 实验

协作过滤技术可以应用于WEB个性化、网上图书、自动问答系统、电子图书馆、网上购物等系统。实验中所用的数据是取自于本校电子图书馆上的数据。实验中共取了7个用户对10个不同的WEB事务页的兴趣度,其中USER1表示当前用户,而其兴趣度的值是经过公式(3)计算得到的。如表1所示。

表1 电子图书馆数据收集表

	C程序 设计	电子 技术	离散 数学	大学 语文	数据 结构	微机 原理	操作 系统	数据库 原理	网络 技术	法律 知识
User1	0.8	0.88	0.89	0.4	0.895	0	0	0	0	0.77
User2	0.5	0.12	0.13	0.92	0.11	0	0	0	0	0.99
User3	0.87	0.95	0.87	0.5	0.88	0.98	0.78	0.75	0.87	0.70
User4	0.45	0	0.14	0.95	0.12	0	0	0	0	0.89
User5	0.9	0.10	0.14	0.86	0	0	0	0	0	0.78
User6	0.98	0.67	0.77	0.89	0.89	0.65	0.89	0.78	0.98	0.88
User7	0.65	0.98	0.67	0.54	0.899	0.98	0.875	0	0	0.789

上表中值为零的表示该用户还没有浏览该WEB事务,下面就利用公式(5)计算各用户间的相似度:

$$\begin{aligned} \cos(\text{User1}, \text{User2}) &\approx 0.4817 \quad \cos(\text{User1}, \text{User3}) \approx \\ 0.5092 \quad \cos(\text{User1}, \text{User4}) &\approx 0.4778 \quad \cos(\text{User1}, \text{User5}) \\ &\approx 0.477 \quad \cos(\text{User1}, \text{User6}) \approx 0.487 \quad \cos(\text{User1}, \text{User7}) \end{aligned}$$

$\approx 0.5195 \cos(\text{User2}, \text{User3}) \approx 0.479 \cos(\text{User2}, \text{User4})$
 $\approx 0.701 \cos(\text{User2}, \text{User5}) \approx 0.6637 \cos(\text{User2}, \text{User6})$
 $\approx 0.5157 \cos(\text{User2}, \text{User7}) \approx 0.503 \cos(\text{User3}, \text{User4})$
 $\approx 0.4756 \cos(\text{User3}, \text{User5}) \approx 0.4793 \cos(\text{User3}, \text{User6})$
 $\approx 0.4833 \cos(\text{User3}, \text{User7}) \approx 0.5142 \cos(\text{User4}, \text{User5})$
 $\approx 0.6699 \cos(\text{User4}, \text{User6}) \approx 0.5192 \cos(\text{User4}, \text{User7})$
 $\approx 0.4991 \cos(\text{User5}, \text{User6}) \approx 0.5017 \cos(\text{User5}, \text{User7})$
 $\approx 0.4756 \cos(\text{User6}, \text{User7}) \approx 0.4941$ 。

根据上面的计算,在这里采用基于中心的方法为当前用户 User1 寻找相似群体。取相似度的值大于 0.5 作为当前用户的相似用户群体,因此可得到 User3, User7 同 User1 形成相似用户群体,且 User7 同用户最相似,User3 次之。再根据公式(4)计算 User1 对微机原理和操作系统这两个项目的预测兴趣的值分别为 0.509 和 0.455,所以可以给当前用户 User1 推荐微机原理。从而完成了利用协作过滤技术实现个性化推荐。

4 结束语

随着互联网的普及和信息服务需求的增长,协作过滤技术在电子图书馆、电子商务、WEB 个性化、自动问答等系统得到了广泛的应用。文中将显式和隐式计

算用户兴趣度的方法相结合,提出了一种新的计算用户兴趣度的方法,并在此基础上研究了基于兴趣度的协作过滤技术,给出了利用协作过滤技术寻找兴趣相似的用户群体的算法。

并通过实验验证了利用该方法可以为当前用户推荐他所关心的网络事务。

参考文献:

- [1] 张莹.从商务网站用户行为数据提取用户兴趣[J].潍坊学报,2005,5(4):21-23.
- [2] Aggarwal C, Yu P. Data Mining Techniques for Personalization[J]. IEEE Data Engineering Bulletin, 2000, 23(1):4-9.
- [3] 张新香. WEB 日志挖掘在电子商务中的应用研究[J]. 计算机系统应用, 2006(1): 52-55.
- [4] 周晓兰,王随平. WEB 文本挖掘中用户兴趣模型的建立和更新[J]. 湘潭师范学院学报:自然科学版, 2006, 28(3):33-36.
- [5] Resnick P, Iacovon N, Bergstrom P, et al. Gouplen: An Open Architecture for Collaborative Filtering of Netnews[C]//In: Proceedings of CSCW' 94. Chapel Hill, NC: [s. n.], 1994.
- [6] Dahlen B J, Konstan J A, Herlocker J L, et al. Turn - starting Movielens: user Benefits of starting a Collaborative Filtering system with "Dead Data"[R]. Minneapolis: university of Minnesota, 1998.

(上接第 105 页)

3 结论

给出了浮点转定点问题的一个定理和一个推论,对于文中的算法在浮点转定点运算精度问题上给出了理论上的支持。针对目前已有 DCT 算法误差较大的缺点,提出了两种新的 VLIW 结构下的定点 8×8 DCT 快速算法,按照做完水平 DCT 变换后的处理方式,新算法又可分为高精度算法和低精度算法。新高精度算法比已有的定点高精度算法^[6]分别提高 13.4% 和 21.4%,而误差方差总和分别降低了 80% 和 67.5%。高精度算法比已有的低精度算法^[6]运算速度分别提高 8.5% 和 17.2%,而误差方差总和分别降低了 98.98% 和 98.3%。另外,文中还给出来了一种估计在同时执行多个指令操作的 DSP 中估计算法实际运行时间的方法,这些方法对于在允许多个指令同时执行的 DSP 上优化算法有理论上的指导作用。

参考文献:

- [1] Elnaggar A, Alnuweiri H M. A new multidimensional recursive architecture for computing the discrete cosine transform[J]. IEEE Trans on Circuits and Systems for Video Technology, 2000, 10(1):113-119.
- [2] Wang Z S, He Z Y. A generalized fast algorithm for 2-D discrete cosine transform and its application to motion picture coding[J]. IEEE Trans on Circuits and Systems II, 1999, 46(5):617-627.
- [3] Chen Xinjian, Dai Qionghai, Li Chunwen. A fast algorithm for computing multidimensional DCT on certain small sizes[J]. IEEE Trans on Signal Processing, 2003, 51(1):213-220.
- [4] Loeffler C. Practical fast 12D DCT algorithms with 11 multiplications[J]. Acoustics, Speech, and Signal Processing, 1989, 289(12):988-991.
- [5] Chen W A, Harrison C, Fralick S C. A Fast computational Algorithm for the Discrete Cosine Transform[J]. IEEE Transactions on Communications, 1977, 25(9):1004-1011.
- [6] Volume 2 Part D CookBook Trimedia Compilation System 4.3 User Manual v1.0[S]. 2003:22-28.
- [7] 李学明,李继.用超长指令实现 DCT 的新算法[J]. 电子学报, 2003, 31(7):1074-1077.
- [8] TM3260 Architecture Databook TriMedia VLIW Core Rev. 1.02[S]. 2004.
- [9] Hou H S. A Fast Recursive Algorithm for computing the Discrete Cosine Transform[J]. IEEE Transactions on Acoustics, Speech and Signal Processing, 1987, 35(10):1455-1461.