

产品开发协同工作环境模型及其应用

龙红能^{1,2},王为民¹,戈鹏²,夏开渝¹,王政¹

LONG Hong-neng^{1,2},WANG Wei-min¹,GE Peng²,XIA Kai-yu¹,WANG Zheng¹

1.东方汽轮机厂,四川 德阳 618201

2.四川大学,成都 610065

1.Dong Fang Steam Turbin Works,Deyang,Sichuan 618201,China

2.Sichuan University,Chengdu 610065,China

E-mail:lhnsu@vip.sina.com

LONG Hong-neng,WANG Wei-min,GE Peng,et al.Collaboration model for product development and its application. Computer Engineering and Applications,2007,43(20):235-239.

Abstract: Team working and collaboration is involved in product development process.Collaboration technology is key in realizing product development process management.Through analyzing product development collaboration working character,a collaboration model for product development is put forward consisting of client view,security control,whole character and network.The key application technology is studied.The application and verification of the model is done in the developed product information integration and sharing platform and solve the problem of large-size nuclear steam turbine collaboration development and paperless application.

Key words: collaboration work;environment model;process management;edition management

摘要: 产品开发过程是多人参与协作的设计过程,协同技术是实现产品开发过程管理的一个关键技术。通过分析产品开发协同工作的特点,提出一种由用户视图层、安全控制层、全局特征层以及网络层所组成的产品开发协同工作环境模型,研究了实施的关键技术,并在所开发的产品信息集成共享平台中获得应用验证,较好解决了大型核电汽轮机协同开发与无纸化应用问题。

关键词: 协同工作;环境模型;过程管理;版本管理

文章编号:1002-8331(2007)20-0235-05 文献标识码:A 中图分类号:TP311

1 产品开发的协同工作环境模型

1.1 产品开发协同工作的特点

产品的开发过程是一个多人参与、跨越多个部门、涉及多个领域知识的复杂过程,它不仅需要设计、材料、标准化、工艺等不同领域的知识和专家的经验,更重要的是要有综合和协调这些经验知识的有效机制和知识,来耦合不同专家的设计任务。产品开发过程具有以下特点^[1]:

(1)多主体。产品开发过程中往往是多人参与一项开发任务,为了完成一个产品零部件的开发任务,有人负责产品设计,有人负责材料研究,有人负责制定工艺路线、材料定额,有人负责编写工艺卡片、制定工时定额等等。这些开发人员既相互独立又相互联系,分别拥有各自的领域知识、经验和一定的问题求解能力。

(2)动态联盟。根据任务的需要,具有不同专长的开发人员组成动态的设计联盟组织,联盟成员可以动态地加入或退出,参与开发的设计人员可以是不固定的,联盟的组织结构也是可以变动的。

(3)一致性。一个联盟要实现的总的开发目标是一致的,所在环境的上、下游信息是一致的,使用统一的产品信息共享模型、统一的知识表示方法。

(4)协同性。具有协同各种开发人员、管理人员完成共同目标的机制,其中包括通信协议、冲突检测等。

(5)互相信任。设计联盟的存在是建立在盟员间互相信任的基础之上的。

(6)流程。开发的参与者都必须遵守规定的设计流程,例如核电产品设计明细必须经过校对、审核、标准化会签、材研会签、工艺会签、审批等才能发布生效,核电产品工艺明细必须经过校对、锻热会签、生产处会签、审批等才能发布生效,结果修改必须进行修改申请、经批准后进行升级换版等等。

1.2 产品开发协同工作的内涵

(1)产品开发协同工作是一个知识共享和集成的过程。知识共享包括领域内共享和领域间共享。在设计或工艺部门内

基金项目:四川省重点科技研究项目(No.2006Z03-004)。

作者简介:龙红能,男(汉族),四川大学博士后研究人员,东方汽轮机厂计算机处副处长、高级工程师,主要研究方向为CAPP/CAM,CBR,ERP,PDM等;王为民,东方汽轮机厂总工程师,研究员级高级工程师。戈鹏,四川大学工商管理学院副教授;夏开渝,东方汽轮机厂副总工程师,计算机处处长,研究员级高级工程师;王政,东方汽轮机厂厂长助理,副总工程师,总工艺师,研究员级高级工程师。

部,开发人员必须共享设计的经验和知识,在设计与工艺部门之间,工艺人员必须与设计人员共享产品设计信息、知识。这些共享的经验和知识最终还必须进行整理、集成,形成领域知识库,并在此基础上产生新的观点和设计方案。

(2)产品开发协同是一个管理过程。强调产品开发中的管理任务,包括任务管理、过程管理、版本管理、用户权限管理等。

1.3 产品开发协同工作环境模型

根据以上分析,本文给出了产品开发协同工作环境模型^[2]如图1所示。模型包括四层:

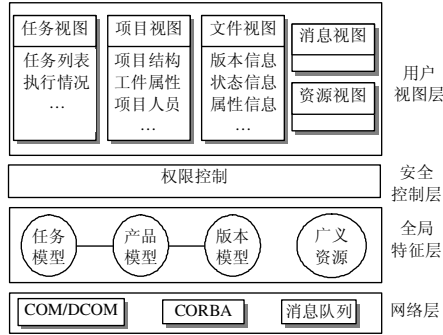


图1 产品开发协同工作环境模型

(1)用户视图层。是在用户的权限范围内可以看到的全球特征层信息的子集,反映用户登入系统后的各种视图信息,视图信息可能跨越全局特征层的多个模型,并将之以最合理的形式呈现出来。该层包括:

①任务视图,反映登录用户当前被分配的任务信息,包括任务的内容、执行情况等等。如图2所示的核电产品登录用户任务列表。



图2 核电产品登录用户任务列表

②项目视图,反映登录用户参与的项目属性、项目结构以及参与该项目的成员信息。项目视图是集成共享产品模型的子集。如图3所示的项目视图,反映了登录用户授权参与的项目信息、项目结构以及参与该项目的人员及执行状态等信息。

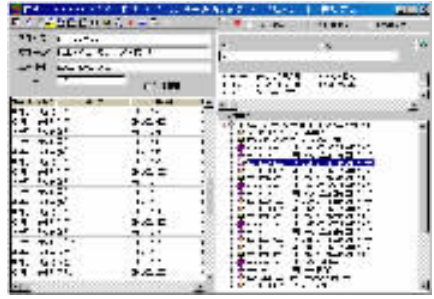


图3 核电产品登录用户项目视图

③文件视图,反映了隶属零部件的文档信息,包括文档的版本、创建的时间、修改的历史以及文件当前的所处的流程状态等等。如图4所示的零部件E804H00403G03所属的设计文档信息。

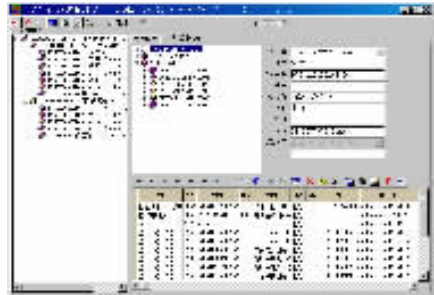


图4 核电产品文件视图

④消息视图,反映登录用户收到的各种系统消息、项目消息的列表。利用消息视图可以反映一些即时的协同设计信息。目前系统未主动提供消息,用户根据需要获取,如图5所示的发布消息列表。

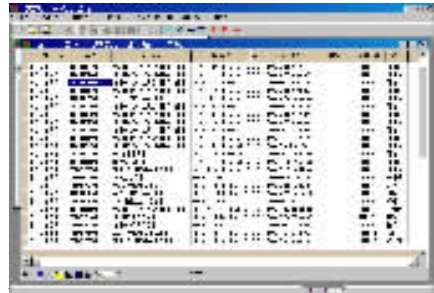


图5 核电产品登录用户发布消息列表

⑤资源视图,反映登录用户可见的各种企业资源信息,包括资源信息的分类结构、资源的详细属性信息等等,如图6所示的资源视图。

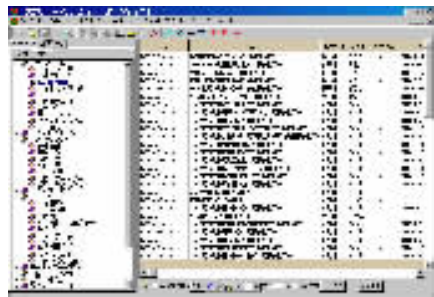


图6 资源视图

(2)安全控制层。用户所看到的各种视图信息都必须经过安全控制层的过滤,以确保用户只看到它应看到的全局特征层

的内容,只做他能够执行的操作。

(3)全局特征层。描述多种全局数据信息,包括系统的任务管理信息、产品信息、版本信息以及广义资源信息。

(4)网络层。为产品开发的协同工作环境提供网络支撑。在产品开发过程中,由于涉及到多种不同的任务类型、系统平台,其通讯方式也有较大的差别。

产品开发协同工作环境的形式化模型可以通过如下定义反映:工作空间(WS, Work Space)是用户在成功登录系统以后的协同工作环境。工作空间

$$WS = \langle RL, PV, TV, DV, RV, SM \rangle.$$

式中,RL——用户的权限实体集, $RL = \{ID, R_i, i=0 \dots I\}$, 其中ID为用户标识, R_i 是用户的权限实体;PV——项目视图, $PV = \langle F, BOM \rangle$, 其中F为产品的特征集, BOM为产品的结构;TV——

任务视图, $TV=\{T_j, j=0 \dots J\}$, 其中 T_j 为分配给当前登录用户的任务; DV ——文档视图, $DV=\{D_k, k=0 \dots K\}$, 其中 D_k 为分配给当前登录用户的任务; RV ——资源视图; SM ——即时短信。

产品开发以工作空间的形式维护用户协同工作环境的上下文信息。

2 产品开发协同工作环境模型的关键技术

2.1 过程管理

产品开发过程管理就是对产品开发过程进行监控和调度, 对多种信息进行管理。过程管理主要涉及三类对象: 任务、数据和人员。

(1) 任务: 产品开发过程是由一系列的任务构成, 任务是可以分解的, 如企业级开发任务分解为多个部门任务、部门任务分解为多个任务元。任务元的角色可以分为设计角色和签审角色, 设计角色产生数据, 签审角色认可数据。

(2) 数据: 设计过程中要处理到各式各样的信息, 包括开发过程支持数据、设计数据和设计过程数据, 设计过程数据是指反映设计过程中设计数据的发放、变更、审批、签字等的的数据。这些数据是由设计过程中的不同角色来处理的, 角色的实质就是使设计过程中的各种信息清晰化、完整化。

(3) 人员: 在设计过程中, 不同的角色要求由不同的用户来执行。此外, 要保证数据的一致性和安全性, 也需要对参与设计过程的用户进行权限管理。

三个对象之间是相辅相成的关系, 如图 7 所示。数据支持设计人员作出决策, 设计人员对数据的操作通过权限进行管理; 任务需要对数据进行操作, 同时也输出数据; 设计人员可以承担某个任务, 也可以动态地建立或撤消与任务之间的关联。

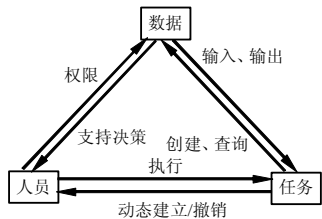


图 7 对象之间的关系

因此, 产品开发过程管理需要对任务、数据、人员进行管理, 如图 8 所示。任务管理和数据管理是主要要求。产品开发过程管理的核心是工作流程管理, 在《基于知识框架的工作流建模及其在核电产品开发中的应用》中, 作者详细论述了工作流建模方法及其应用方法。

2.2 协作任务机制与角色权限

2.2.1 任务模型

产品开发协同工作环境模型采用面向任务驱动的协同工作方法。系统的形式化模型^[3]通过以下定义反映:

(1) 过程模型 P 是一个有限状态集, $P=\langle S, I, f, s_0, F \rangle$ 。其中, S ——有限状态的集合; I ——输入事件的集合, $I=\{e_1, e_2, e_3, \dots\}$; f ——状态转移函数, 为状态和事件指派一个新状态; s_0 ——初始状态; F ——终结状态, $F \in S$ 。

(2) 状态 S 是三元组, $S=\langle Name, functionary, v \rangle$, 其中, $Name$ ——状态的名称; $functionary$ ——负责该状态的负责人, v ——状态的权值, $0 < v < 1$ 。在状态中引入权值是为了描述一个

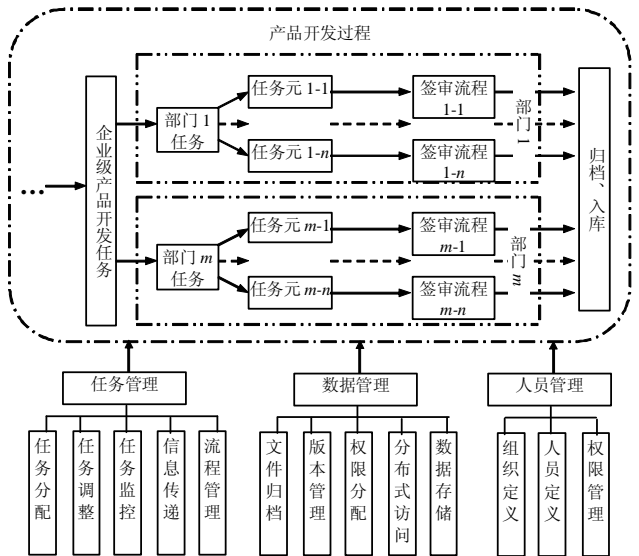


图 8 产品开发过程管理内容

任务处于某个状态时的执行程度。

(3) 一个状态的执行人 $functionary$ 是一个二元组, $functionary=\langle Name, RL \rangle$, 其中, $Name$ ——执行人的姓名; RL ——执行人所属角色的集合, 如设计明细设计、设计明细审核、工艺明细审批等等。

(4) 角色 $Role$ 是一个三元组, $Role=\langle Name, C, L \rangle$, 其中, $\langle Name$ ——角色名; C ——角色的能力集合, 如熟悉热处理工艺、熟悉中小件工艺等等; L ——一组授权的集合, 表明此角色对其它资源的访问或操作权限。

(5) 任务 T 是一个五元组, $T=\langle D, O, P, s, g \rangle$, 其中, D ——一个谓词, 表示要执行的一个动作; O ——动作 D 的作用对象; P ——任务的执行过程模型; s ——任务的当前执行状态; g ——状态进度指示。

一个复杂的任务可以分解为若干个子任务(称一个不含子任务的为原子任务), 子任务可以继承父任务的过程模型。任务的执行程度由任务的当前执行状态 s 和状态进度 g 指示。 g 是一个状态内部的指标, 由于原子任务的工作量很难准确估计, 故 g 仅仅是用当前已完成数来表示, 如已完成的工艺卡片的页数, 文件从设计之初到现在累积被打开的时间等等。父任务 T_p 的进度 g_p 可以通过下式计算:

$$g_p = (\sum s_i \cdot v) / n, i=1 \dots n$$

式中 n 为 T_p 的子任务个数, $s_i \cdot v$ 为第 i 个子任务的当前状态 s_i 权值。 g_p 可以给出复杂任务的综合进度指标。

任务的 T 动作对象 O 是一个知识元的集合, 即 $O=\{k_0, k_1, k_2, \dots\}$, 在 O 中有几个固定的知识元分别用来表示与任务 T 绑定的零部件名称、零部件编号、工艺文件名称、工艺文件编号等。

2.2.2 任务分解

任务分解是指提供结构化的任务定义方式, 遵循自顶向下逐步细化的原则, 将任务分解为一个相对独立的子任务, 通过任务分配, 分解后的任务被分别分配给不同的设计者。

由于实际中, 一份文档可能被多个任务访问, 从而造成冲突。这就涉及到任务协调机制的问题。

2.2.3 基于锁的任务协调机制

产品协同开发是一个在多用户参与的协同设计环境, 存在

着许多并发活动,这种并发活动必须加以控制,使这些并发活动不会产生相互干扰。本文通过引入锁机制^[4]来消除这样的相互干扰,使设计活动表现为一种协同的状态。锁分为以下两种类型:

(1)只读锁。除了锁的拥有者之外,其他设计人员只能对设计对象进行读操作,不能保存。

(2)排它性锁。除了锁的拥有者之外,别的用户不能对设计对象进行任何操作。

锁的分配由系统的锁控制器分配,锁被作为设计对象的一个属性保存在数据库中。

2.3 版本管理

2.3.1 版本控制特点

产品开发中设计对象的版本呈现出三个特点,即多版本、动态性和关联性^[5]。

(1)多版本。一个零部件的一个文档(设计明细、工艺明细、工艺卡片等)同时存在着个版本。主要是由于产品开发是在各种约束条件下,进行反复迭代(设计和修改),获得产品满意解或最优解的过程,因此,对设计结果的修改是一种经常性现象。每次修改产生新的设计版本。在对旧版本进行修改,而新版本未得到最终审批时,出于版本保护的需要,旧版本依旧保存,这样,新旧两个版本得以共存。有时,为了比较和回顾,版本更新多次后,保存了多个旧版本以便于了解设计全过程。这种版本可以用时间进行标记。

(2)动态性。是指版本的状态在设计过程中时刻发生着变化,每个版本存在其相应的生命周期,某一时刻处在某一特定状态。对版本的操作引起其状态的转换。当设计人员设计完一个对象后,即生成一版本。版本生成后需要提交给有关人员审批,此时的版本为待批状态,数据不能被使用。一旦得到了批准,则此版本状态即转为有效。若被否决,版本状态转为无效。有效版本在设计需求变更或其它相关条件变更时,转为无效。无效版本可能被删除,从而在系统中消失。每一版本处于某状态,并随时可能向下一状态转化。

(3)关联性。关联性是指同一项目中各个对象的版本之间存在相互关联。一个工程对象往往由一系列简单的工程对象组成,呈现复合特性。因此,各子对象的当前有效版本的复合形成整个工程对象的一个有效版本。一种简单的复合是树状复合,例如零部件组成部件,部件组成产品,相应地,各零部件的设计版本复合成部件的版本,各部件的版本复合成产品版本。则整个产品的版本组成是一棵“版本树”。

2.3.2 版本管理内容

(1)版本信息建立和维护。当一份设计完成后,必须为其建立相应的版本信息。版本信息中应对此版本进行唯一标识,并能够反映出此版本的相关信息,例如人员信息、版本生成日期等。在版本的整个生命周期中,版本信息应该能反映出版本的状态变化。版本的生成有可能需要对多个版本进行合并,在继承了多个原版本的内容后,可能增加了自己的内容,或者对继承的内容进行了修改,最后生成了新版本。版本生成后,版本信息中应该表示出这种继承关系,还应该表示出版本数据的引用情况。

(2)版本操作。设计过程中对版本进行各种操作,使版本的状态发生着变化。典型的版本的操作有:版本建立、版本提交、版本审核、版本删除、版本复制、版本数据引用和版本发布等。这些操作是保证设计人员在需要的时候取得正确数据的重要基础。

(3)版本的安全性维护。由于产品开发中设计对象的信息有多个使用者,因此需要对不同的人员设置不同的权限,以维护版本的安全性。权限的种类包括读/写、复制、删除、审核等。为了防止对版本的误操作,需定义版本的一些操作规则,如在新版本未被批准为有效时,旧版本不许被删除、修改版本只能在原版本的复制件即副本上进行等。

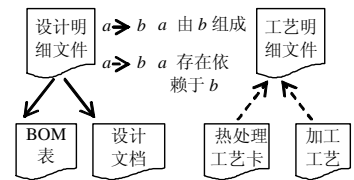
(4)版本的一致性维护。为使设计人员能够访问到所需数据,并保证数据的有效性,应该维持版本的一致性。新版本被批准为有效后需进行发布,及时地通知原版本数据使用者。新版本的生成,也需要对与其有约束关系的版本进行检查以判定约束是否仍然满足。

2.3.3 版本控制模型

根据以上分析,本文给出的产品开发过程版本控制模型的形式化定义^[6]如下:

(1)一个零部件的文档空间 DS (Document Space)是一个二元组: $DS=\langle D,R \rangle$,其中 D ——隶属于零部件的所有文件的集合; R —— D 中文件之间的关系集合。

在讨论文档以及文档之间的关系时都是在文档空间中进行的。如图9所示,产品开发文件之间常见的关系有聚合关系和存在依赖关系两种,如BOM表与设计文档为聚合关系,工艺明细与工艺卡片之间为依赖关系。在文档空间讨论文档时只考虑文档之间的关系,而不考虑文档的版本特性。



(a)聚合关系 (b)存在依赖关系
图9 产品开发文件之间的关系

(2)文档 D 是一个二元组, $D=\langle DID,RS \rangle$, DID ——是文档的唯一标识; RS ——文档中记录的集合。

不同类型的文档中“记录”的含义是不同的,本文中文档(保存在数据库中)的记录是一个元组,因此文档是元组的集合。无结构(或不知道其内部结构)的文档中以一个字节为记录单位。

(3)被确认修改的文档称为文档的一个版本,记作 V , $V=\langle C,\Delta \rangle$, C ——修改后的文档相对于修改以前没有变动的部分; Δ ——修改后的文档相对于修改以前变动的部分。

版本之间的关系如图10所示。文档在 v_1 的基础上作了修改得到 v_2 , Δ 为修订的内容。用 $\Delta(v_1,v_2)$ 表示从版本 v_1 到 v_2 针对文档所作的修改。

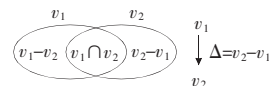


图10 版本之间的关系

(4)文档的所有的版本的集合称为文档的版本空间,记作 VS 。实际中用版本对象 VO 来描述文档的一个版本, $VO=\langle DID,VID,PVID,OID,Date\ Time,Description,Source,\Delta \rangle$,其中 DID (Document Identifier)是与该版本关联的文档的唯一标识; VID (Version Identifier)版本的唯一标识,用于在 DID 文档的众多版本中唯一标识一个版本,一个新创建的文档的版本号为0;

PVID(Previous Version Identifier)是前驱版本的标识;OID(Operator Identifier)是产生 VID 版本的操作员的编号;Description 是该版本的说明;Source 是 VID 版本的文档的内容; Δ 是 VID 版本在 PVID 版本的基础上所作的修改。本文用 $\Delta(v_1, v_2)$ 表示从 v_1 版到 v_2 版所作的修改,则 $\Delta(v_1, v_2) = \{op_1, op_2, \dots, op_n\}$, 其中 op_i 被称为版本项, $op_i \in \{\text{Insert}, \text{Update}, \text{Delete}\}, i=1 \dots n$ 。

系统以新版本保存文件的同时,也将 Δ 以修改日志的形式同时保存,可以通过 Δ 对版本进行跟踪,如图 11 所示的版本差异信息。

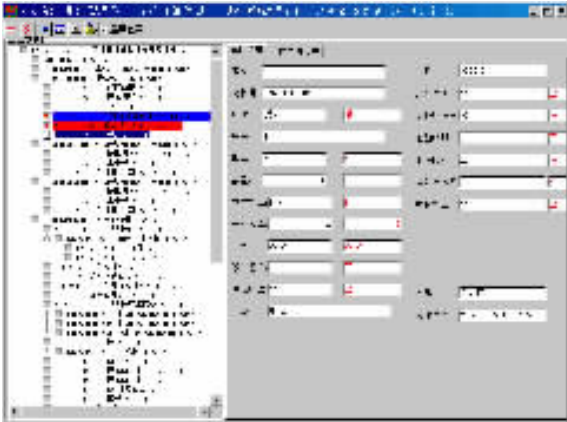


图 11 版本差异信息(Δ)显示

3 小结

上述理论已经被用于所开发的产品信息集成共享平台中,并在东方汽轮机厂大型核电汽轮机产品开发中获得实际应用。实现了核电产品开发中的开发过程支持数据的管理,设计数据(设计明细)和工艺数据(工艺明细和工艺卡片)的管理,签批过程、更改过程的有效控制和过程数据的管理,为核电产品开发的流程管理与无纸化应用,实现以纸质为中心向以电子数据为中心转移发挥了重要的作用。(收稿日期:2006年11月)

参考文献:

- [1] 魏宝刚,潘云鹤.协同设计技术的研究[J].中国机械工程,1999,10(4):454-457.
- [2] 赵剑颖,赵正德,王正茂.实时协同编辑系统共享工作空间的研究[J].计算机工程,2001,21(1):41-43.
- [3] 洪帆,邓集波.基于任务的授权控制及其实现[J].计算机工程与科学,2002,24(2):61-65.
- [4] 殷勇,蔡希尧.CoVer:一种支持协同技术的版本模型[J].西安电子科技大学学报,1998,25(2):193-197.
- [5] 曹健,张友良,赵海燕.协同设计的版本管理[J].计算机集成制造系统,1998(6):16-20.
- [6] Supanat Kitcharoensakkul,Vilas Wuwongse.Unified versioning using resource description framework [J].Annals of Software Engineering, 2001(11):259-297.

(上接 230 页)

在实时性比较中,本算法与文献[6]中仅利用空间信息进行分割相比,运行速度提高了 2.3 倍,与文献[7]中的提出的高效时空结合分割算法相比,时间上也具有优越性(见表 1)。从实验结果可以看出,本算法兼顾分割效果和实时性两方面,分割效果比较精确。与传统的方法相比,分割视频序列中运动前景具有比较明显的优势。而且由于采取了两类累积帧差交集聚类的方法,使得属于运动区域的象素能够准确的收敛到前景轮廓附近,克服了文献[5]前景轮廓模糊现象和边界效应,同时序列的利用率也提高了一倍。

表 1 运行时间比较

	本文方法/s	文献[7]方法/s
文件 I/O	5.9	5.5
时域分割	22.3	17.5
空域分割	0.0	4.4
形态学操作	6.1	11.8
总计	34.3	39.2

3 结论

本文提出了一种从可视电话、视频会议等视频序列中分割运动前景的算法。首先,对视频序列同时做邻帧差分与隔帧差分,然后采取将两类累积结果交集聚类的方法获取前景轮廓,克服了传统方法的局限性,用少量的帧数就获得了较好的分割效果,提高了序列的利用率;另外,由于该算法充分利用了视频序列的时域信息,不涉及复杂的运算,便于实现,实时性高。实

验证明,该方法快速准确,是一种较好的分割视频序列中运动前景的算法,对于数字视频检索、图像编码及压缩具有一定的应用价值。(收稿日期:2006年11月)

参考文献:

- [1] Meier T,Ngan K N. Automatic segmentation of moving object for video object plane generation[J].IEEE Transactions on Circuits and Systems for Video Technology,1998,8(5):525-538.
- [2] Wildes R P. A measure of motion salience for surveillance applications[C]/Proc of Image Processing,1998:102-121.
- [3] Collins R T. A system for video Surveillance and Monitoring, Technical Report CMU-RI-TR-00-12[R].Carnegie Mellon University, 2000
- [4] Meier T,Ngan K N. Video segmentation for content-based coding[J]. IEEE Trans on Circuits and Systems for Video Technology, 1999,9(8):1190-1203.
- [5] 许悦雷,左继章,张雄.一种累积帧差视频对象分割算法[J].光电工程,2004,31(7):69-72.
- [6] Yaakov Tsaig,Amir Averbuch. A region-based MRF model for unsupervised segmentation of moving objects in image sequences[C]/IEEE Computer Society Conference,2001,1:889-896.
- [7] Chien Shao-Yi, Ma Shyh-Yih. Efficient moving object segmentation algorithm using background registration technique[J].IEEE Transactions on Circuits and Systems For Video Technology,2002,12: 577-582.