

基于颜色 Petri 网的服务组合建模与验证^①

余 晓¹, 吴剑章², 王 巍²

¹(东南大学 继续教育学院, 南京 210096)

²(东南大学 计算机科学与工程学院, 南京 210096)

摘 要: Web 服务组合技术能快速、灵活的满足复杂多变的业务需求, Web 服务组合及其形式化描述和验证已经成为 Web 服务中一个非常重要的研究方向. 给出了一个基于颜色 Petri 网(CPN)的 Web 服务和 Web 服务组合的形式化定义和描述, 主要描述了 4 种组合结构: 顺序、选择、并发和循环. 给出了一个颜色 Petri 网建模 Web 服务组合的实例, 并且对模型进行了简单的正确性分析. 分析结果表明, 该建模方法满足正确性要求.

关键词: Web 服务组合; 颜色 Petri 网(CPN); 建模; 验证

Colored Petri Net-Based Service Composition Modeling and Verification

YU Xiao¹, WU Jian-Zhang², WANG Wei²

¹(School of Continuing Education, Southeast University, Nanjing 210096, China)

²(School of Computer Science & Engineer, Southeast University, Nanjing 210096, China)

Abstract: Using web service composition technology, business demand can be meet quickly and flexibly, formal description and verification of web services and their composition are very important research issues. A colored Petri net(CPN) based model is proposed to formally define and descript Web services and Web services composition, four structures of composition are mainly descript: sequence, choice, concurrent and loop, an example of web services composition modeled by Colored Petri net is proposed, the accuracy of the model is simply analyzed. The result shows that the demand of accuracy of the model is satisfied.

Key words: web services; colored petri net (CPN); model; verification

21 世纪初以来, Web 服务作为跨组织应用集成的支持技术得到学术界和企业界的广泛支持. Web 服务有一个清晰的、机器可读的、通用的标准描述, 即 WSDL (Web service description language), 其它系统以该描述的方式与 Web 服务进行交互. 面向服务的体系架构(Service Oriented Architecture, SOA)成为信息技术的新热点. SOA 最重要的特征是松耦合, 即异步性、并发性和自治性. 而 UDDI(universal description, discovery, integration)主要的作用是发布和查找服务. 然而, 这些技术没有实现复杂的 Web 服务组合, 所以上述技术对服务组合的支持是有限的^[1-4].

当前主要有基于工作流、语义网络和形式数学三种服务组合方法, 如 Pi-演算或 Petri 网等, 基于工作流

和语义网络的方法没有提供验证机制或工具支持来验证组合的有效性. 而基于 Petri 网的方法具有异步和并发的特点, 相对于 Pi-演算, Petri 网能对包含结构的复杂系统进行建模. 当前, 存在大量的用 Petri 网来建模服务组合的工作, 有服务组合的形式化描述^[5]; 服务组合的逻辑关系^[5]; 服务组合建立、执行算法^[6]等, 还包括以 Petri 网为数学理论支持和分析工具, 建模和分析服务组合的可达图、标识树、不变量等, 从而对服务组合模型进行进一步的仿真、分析和验证. 然而, Petri 网仅仅关注的是 Web 服务模型的控制和转移建构, 但它不考虑数据流转的过程.

使用颜色 Petri 网, 不仅支持用 Token 颜色对数据类型进行建模, 而且在控制描述控制流的过程中同时

^① 收稿时间:2012-01-09;收到修改稿时间:2012-02-23

也描述了数据流^[7]. 使用颜色能够丰富数据对象自身的属性, 因此颜色 Petri 网的描述能力得到很大的加强. 文献[7]采用了颜色 Petri 网对 web 服务进行建模. 但系统模型描述比较简单, 没有发挥出颜色 Petri 网强大的描述能力. 在文献[8]中, Guo 等人提出了一个用于 Web 服务组合的 CP-net 模型, 然而, 文中组合的规则和过程必须事先定义好, 并且在没有前提条件的情况下, 服务组合链并不能自动生成. 文献[9]中提出了一个基于消息导向的 Petri 网模型, Web 服务基于消息来进行组合, 但是这种方法增加了组合的复杂度. 实际应用比较困难. 鉴于采用的 Petri 网局限于描述 web 服务及其组合的控制流, 而数据流和控制流相结合才能全面、完整地描述服务组合. 本文中, 我们将使用颜色 Petri 网(CPN)^[4]对可用的 Web 服务进行组合, 用 Token 颜色表达服务的数据类型, 用变迁描述服务发生的条件, 利用 CPN 理论及建模工具来分析和验证基于上述理论构建的实例模型. 全面刻画 web 服务及其内部结构.

本文的剩余部分结构如下: 1 详细介绍了几个颜色 Petri 网建模 Web 服务组合的基本概念以及几种基本控制结构; 2 给出了一个用颜色 Petri 网建模 Web 服务组合的实例; 3 是对该实例的颜色 Petri 网模型进行简单的分析和验证; 4 对全文做了小结.

1 基于颜色Petri网的服务组合建模

使用颜色 Petri 网对 Web 服务组合进行建模, 首先要将 Web 服务用颜色 Petri 网表述出来, 下面给出颜色 Petri 网的定义及基于颜色 Petri 网的 Web 服务的定义.

1.1 基于颜色 Petri 网的 Web 服务定义

定义 1 (颜色 Petri 网). 一个简单的(带 k 种颜色的)颜色 Petri 网定义如下:

$$\Sigma = (S, T; F, C, M_0)$$

其中, (S, T; F)是一个网,

$$S \cap T = \phi, S \cup T \neq \phi, F \subseteq (S \times T) \cup (T \times S),$$

$$\text{dom}(F) \cup \text{cod}(F) = S \cup T;$$

C 是一个数据类型函数, 定义为 $C : S \rightarrow K$, 假设有 K 种颜色, 表示 Web 服务中涉及的数据类型. 各个库所内的 K 维向量表示库所内各种颜色标志个数, 一个分量表示一种颜色, 分量的数值表示对应颜色标志的个

数; M_0 是初始标识. 为了更好地描述服务的行为, Petri 网中会包含输入库所(没有输入弧)和输出库所(没有输出弧)^[10]. 在 Petri 网中, 将会定义 Web 服务组合的操作并且分析和验证组合的一些属性, 比如可达性、可用性和安全性等. 在任何时刻, 一个 Web 服务只可能处于未初始化, 准备, 运行, 挂起和结束这五个状态中的一种状态. 图 1 给出了一个简单颜色 Petri 网的例子, 图中共有三种颜色, 输入库所 i 的颜色标识为 [1, 3, 2], 即初始标识 M_0 为 [1, 3, 2], 第一种颜色的标志个数为 1, 第二和第三种颜色的标志个数分别为 3 和 2, 库所 S_1, S_2, S_3 中没有标志. 弧上的向量标识表示变迁发生所需要的条件, 即库所中的颜色种类和个数必须满足弧上的颜色种类和个数. 最后, 输出库所中的标识为 [1, 3, 2].

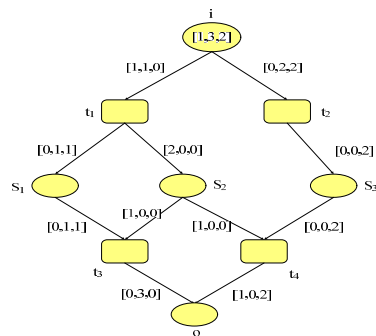


图 1 颜色 Petri 网示例

定义 2 (有色 Petri 网中的 Web 服务). 本文中, 将一个 Web 服务 S 定义为一个八元组: $S = (Id, SName, Desc, Loc, URL, CS, Oper, SN)$, 其中:

Id: Web 服务的唯一标识;

SName: Web 服务的名称;

Desc: 服务的描述;

Loc: 服务所处的位置;

URL: Web 服务的调用;

CS: 该 Web 服务的子服务, 如果 $CS = S.Id$, 则 S 是一个基本服务, 否则 S 是一个组合服务;

Oper: 服务的操作集;

SN: 加入输入库所和输出库所的颜色 Petri 网, 即 $SN = (S, T; F, C, M_0, I, o)$. 输入库所 i 是服务 S 的初始标识, 当执行流程到达输出库所时, 则认为 S 已经终止.

1.2 Web 服务组合的颜色 Petri 网描述

Web 服务组合组件由两部分组成: 原子服务和控

制结构. 控制结构有很多种, 如: 顺序、并行、选择、循环、合并、分支等, 本文只考虑几个常用的结构: 顺序、并行、选择、循环. 基于文献[5], Web 服务组合可以如下定义:

$$S ::= X \mid Seq(S_1, S_2) \mid Choice(S_1, S_2) \mid Conc(S_1, S_2) \mid Loop(S_1)$$

X 表示一个原子服务或空服务(即一个服务没有执行任何的操作);

● Seq(S₁, S₂)表示一个组合服务时顺序执行 S₁ 和 S₂后形成的, 还有一种情况是先执行 S₂后执行 S₁, 实际上, 我们可以决定是先执行 S₁, 还是后执行 S₁, 所以谁先谁后并不重要, 这里, Seq(S₁, S₂)既可以表示先执行 S₁, 又可以表示后执行 S₁, 这个由我们自己决定;

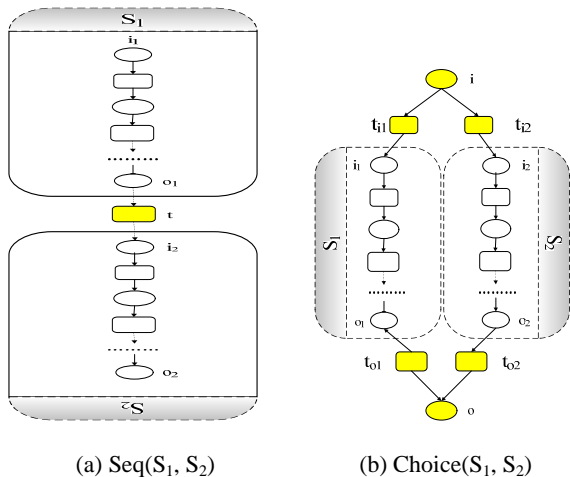
● Choice(S₁, S₂)表示一个组合服务是执行 S₁ 或者 S₂后形成的(不可兼或), 一旦执行它们中的一个, 则另外一个是不能执行的;

● Con(S₁, S₂)表示一个组合服务是由 S₁ 和 S₂ 并发执行后形成的, 在并发执行的过程中, 可能两个服务之间会有通信发生;

● Loop(S₁)表示一个组合服务是由 S₁ 循环执行 n 次后形成的.

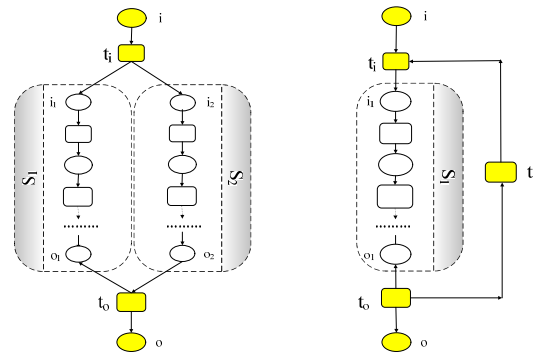
在对服务组合的建模中, 服务的状态为库所, 指定服务的操作为变迁. 在建模的过程中还必须满足几个基本要求, 即: 每个模型都存在输入库所 i 和输出库所 o; 每个变迁库所都在一条从输入库所到输出库所的路径上; 任何情况下, 服务组合都能终止; 组合过程中不存在死组合.

下面用颜色 Petri 网对上述几种控制结构进行建模.



(a) Seq(S₁, S₂)

(b) Choice(S₁, S₂)



(c) Con(S₁, S₂)

(d) loop(S₁)

图 2 四种基本架构的颜色 Petri 网模型

① Seq(S₁, S₂)

有两种情况, 先执行 S₁, 后执行 S₂, 另一种是先执行 S₂, 后执行 S₁, 为方便起见, 这里仅考虑第一种情况, 在 S₁ 执行完以后, 才能继续执行 S₁.

Seq(S₁, S₂)用颜色 Petri 网来描述, 定义如下:

Seq(S₁, S₂)=(Id, SName, Desc, Loc, URL, CS, Oper, SN), 这里,

$$CS=CS_1 \cup CS_2$$

$$SN=(S, T; F, C, M_0, I, o) S=S_1 \cup S_2,$$

$$T=T_1 \cup T_2 \cup \{t\}$$

t 是 S₁ 的输出到 S₂ 的输入的变迁, 所以为了使变迁 t 发生, 必须满足

$$F=F_1 \cup F_2 \cup \{(o_1, t), (t, i_2)\}, C(i_2) \subseteq C(o_1)$$

$$M_0=M_{01} \cup M_{02}, i=i_1, o=o_2$$

用颜色 Petri 网图形成描述 Seq(S₁, S₂), 如图 2(a) 所示.

② Choice(S₁, S₂)

Choice(S₁, S₂)表示只能选择 S₁ 和 S₂ 中的一个, 用颜色 Petri 网描述的定义如下:

Choice(S₁, S₂)=(Id, SName, Desc, Loc, URL, CS, Oper, SN), 这里,

$$CS=CS_1 \cup CS_2, SN=(S, T; F, C, M_0, i, o)$$

$$S=S_1 \cup S_2 \cup \{i, o\}$$

$$F=F_1 \cup F_2 \cup \{(i, t_{i1}), (i, t_{i2}), (t_{i1}, i_1)$$

$$(t_{i1}, i_2), (o_1, t_{o1}), (o_2, t_{o2}), (t_{o1}, o), (t_{o2}, o)\}$$

$$T=T_1 \cup T_2 \cup \{t_{i1}, t_{i2}, t_{o1}, t_{o2}\} M_0=M_{01} \cup M_{02}$$

用颜色 Petri 网图形成描述 Choice(S₁, S₂), 如图 2(b)所示.

③ Con(S₁, S₂)

Con(S₁, S₂)用颜色 Petri 网描述的定义如下:

$$\begin{aligned} \text{Con}(S_1, S_2) &= (\text{Id}, \text{SName}, \text{Desc}, \text{Loc}, \text{URL}, \text{CS}, \\ &\text{Oper}, \text{SN}), \text{ 这里,} \\ \text{CS} &= \text{CS}_1 \cup \text{CS}_2, \text{ SN} = (\text{S}, \text{T}; \text{F}, \text{C}, \text{M}_0, \text{i}, \text{o}), \\ \text{S} &= \text{S}_1 \cup \text{S}_2 \cup \{\text{i}, \text{o}\} \\ \text{F} &= \text{F}_1 \cup \text{F}_2 \cup \{(\text{i}, \text{t}_1), (\text{t}_1, \text{i}_1), (\text{t}_1, \text{i}_2), \\ &(\text{o}_1, \text{t}_0), (\text{o}_2, \text{t}_0), (\text{t}_0, \text{o})\} \\ \text{T} &= \text{T}_1 \cup \text{T}_2 \cup \{ \text{t}_1, \text{t}_0 \} \quad \text{M}_0 = \text{M}_{01} \cup \text{M}_{02} \end{aligned}$$

用颜色 Petri 网图形化描述 Con(S₁, S₂),如图 2(c)表示.

④ Loop(S₁)

Loop(S₁)用颜色 Petri 网描述的定义如下:

$$\begin{aligned} \text{Loop}(S_1) &= (\text{Id}, \text{SName}, \text{Desc}, \text{Loc}, \text{URL}, \text{CS}, \text{Oper}, \\ &\text{SN}), \text{ 这里,} \\ \text{CS} &= \text{CS}_1 \\ \text{SN} &= (\text{S}, \text{T}; \text{F}, \text{C}, \text{M}_0, \text{i}, \text{o}) \\ \text{S} &= \text{S}_1 \cup \{\text{i}, \text{o}\} \\ \text{F} &= \text{F}_1 \cup \{(\text{i}, \text{t}_1), (\text{t}_1, \text{i}_1), (\text{o}_1, \text{t}_0), (\text{t}_0, \text{o}), (\text{o}_1, \text{t}), (\text{t}, \text{i}_1)\} \\ \text{T} &= \text{T}_1 \cup \{ \text{t}_1, \text{t}_0, \text{t} \}, \quad \text{M}_0 = \text{M}_{01} \cup \text{M}_{02} \end{aligned}$$

用颜色 Petri 网图形化描述 Loop(S₁), 如图 2(d)表示.

2 基于颜色Petri网的一个Web服务组合实例

某人要从一个城市 A 到另外一个城市 B 去出差, 他需要知道他出差当天的天气情况以及某个酒店的价格, 现在一共有 4 个 web 服务可供选择, 如图 3 所示. 我们可以假设, C(i)={城市, 日期, 酒店}, C(o)={天气情况, 酒店价格}. 不难发现, 以上 4 个 web 服务中, 没有一个单独的 web 服务可以满足用户的需求, 必须通过 web 服务组合, 得到用户想要的结果. 通过上面 4 个 web 服务的输入和输出, 我们可以发现, S₁ 和 S₂ 组合在一起可以满足要求, S₃ 和 S₄ 组合在一起也可以满足要求, 在这两种组合方案中选择一种即可满足用户需求. 图 4 给出了基于颜色 Petri 网的组合建模.

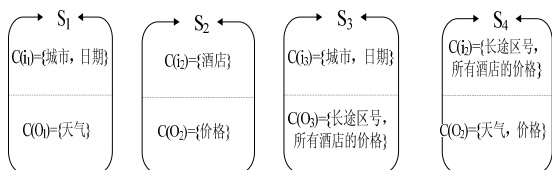


图 3 几个 web 服务

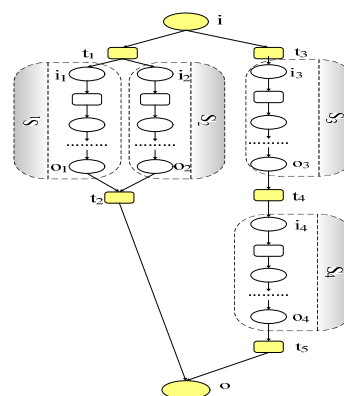


图 4 组合服务的颜色 Petri 网建模

3 颜色Petri网验证服务组合的正确性

下面对组合服务的颜色 Petri 网建模进行模型的验证, 即: 验证图 4 中的 Web 服务组合是否能够正常结束并且满足正确性要求.

Web 服务组合是否能够争持正常结束, 需要通过颜色 Petri 网对组合服务网系统的活性及有界性进行验证, 如果该颜色 Petri 网不是活性的且无界, 那么该组合方案是不合理的; 而通过分析颜色 Petri 网是否具有完全可达性、完整性和前进性来验证服务组合的正确性, 如果该颜色 Petri 网中的某些状态不能达到, 某些变迁不能被激发且存在死循环, 那么该组合服务同样是不正确的. 下面给出服务组合正确性满足的条件:

图 4 中, S₁ 和 S₂ 是并发结构, S₃ 和 S₄ 是顺序结构, 而 S₃ 和 S₁ 与 S₂ 的组合服务是选择结构.

定理 1. 服务组合是正确的, 当且仅当由它构建的颜色 Petri 网具有活性, 有界性、可达性、完整性和前进性.

证明过程可参见文献[14].

分析 Petri 网可采用的方法比较多, 使用的比较多的有: 可达树法、约简、不变量分析等. 其中可达树分析法具有直观简捷, 分析系统的可达性、有界性、活性等各种动态特征很方便, 所以这里使用可达树分析法来分析图 4 中模型的一些特性.

首先, 该服务组合过程中, 出现了城市、日期、酒店、长途区号、所有酒店的价格、天气、价格等 7 种数据类型, 分别用 7 种颜色来描述上述 7 种数据类型, M_i=(i, S₁, S₂, S₃, S₄, o), 则初始标识, 也就是 M₀(i)=([1, 1, 1, 0, 0, 0, 0], 0, 0, 0, 0, 0), 括号后面的 0 指的是七维零向量, 分别表示初始标识中, 库所 S₁, S₂, S₃, S₄ 和 o 中所有颜色

标志的个数为 0, 以此类推:

$M_1=(0,0,1,0,0,0,0), M_2=(1,1,0,0,0,0,0), M_3=(0,0,1,0,0,0,0), M_4=(0,0,1,0,0,0,0), M_5=(0,0,0,0,0,0,0)$. 则可得 web 服务组合的可达树,如图 5 所示.

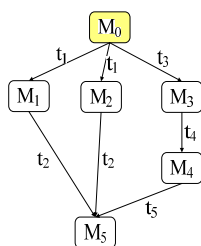


图 5 组合服务的可达树

对图 5 中的可达树进行可达性、有界性、活性、前进性和完整性分析, 结论如下:

① 从 M_0 开始, 可达的状态集合为 $R(M_0)=\{M_0, M_1, M_2, M_3, M_4, M_5\}$, 而状态集 $MS=\{M_0, M_1, M_2, M_3, M_4, M_5\}$. 不难发现, 对于任意一个标记 $M_i \in MS$, 都可以得到 $M_i \in R(M_0)$, 所以组合服务的 Petri 网中任意一个标记都是可以从 M_0 可达的, 即该组合服务任意一个状态都是从 M_0 可达的; 颜色 Petri 网中的有界性与一般 Petri 网中的有界性定义有所区别, 但本质上是一样的, 要找到一个 k (这里 $k=7$) 维非负整数向量 V , 使得对任意的 $M_i \in R(M_0)$, 都有 $M(s) \leq V$, 即可证明库所 s 是有界的, 这里取 V 中的每个分量都为 $[1, 1, 1, 1, 1, 1, 1]$, 即可证明. 所以该组合服务的 Petri 网有界的, 同时也是安全的. 即该组合服务也是安全的;

② 显然, 在图 5 的可达树中, 所有的变迁都可以从 M_0 开始, 被激发至少一次, 所以该组合服务的 Petri 网是具有活性的; 在图 5 的可达树中, 不存在无意义的循环, 所以该组合服务的 Petri 网具有前进性.

在图 5 的可达树中, 从 M_0 开始, 最终都可以到达终止标识 M_5 , 所以该组合服务的 Petri 网具有完整性. 综上所述, 图 4 中的 Web 服务组合满足正确性要求.

3 结语

本文给出了基于颜色 Petri 网的 Web 服务组合模型, 主要考虑 4 种组合模型: 顺序、选择、并发和循环, 并用颜色 Petri 网描述这几种模型. 给出了一个用颜色

Petri 网建模 Web 服务组合的实例, 并对该模型进行了正确性分析, 分析结果表明该模型满足正确性要求. 下一步的工作是进一步用颜色 Petri 网的理论知识来分析 and 验证 Web 服务组合.

参考文献

- 1 Tsalgaidou PT. An overview of standards and related technology in Web services. Distributed and Parallel Databases, 2002,12(2-3):135-162.
- 2 Dustdar SW. A survey on Web services composition. International Journal of Web and Grid Services, 2005,1(1): 1-30.
- 3 Sun HY, Wanf XD, Zhou B. Research and implementation of dynamic Web services composition. Lecture Notes in Computer Science, 2003,28(34):457-466.
- 4 Jensen K. Coloured Petri Nets-Basic Concepts, Analysis Methods and Practical Use, Vol. 1:Basic Concepts. EATCS Monographs on Theoretical Computer Science, Berlin: Springer-Verlag, 1992. 1-234.
- 5 Hamadi R, Benataliah B. A Petri net-based model for Web service composition. Australian Computer Society, Inc. 2003. Conferences in Research and Practice in Information Technology: Australia: Australian Computer Society, Inc, 2003,1-11.
- 6 郭玉彬.杜玉越.奚建清.Web 服务组合的有色网模型及运算性质.计算机学报,2006,29(7).
- 7 Jiang H, Gu JZ, Yu QS. Modeling of Web-based collaborative negotiation system using colored Petri net. Proc. of the 12th International Multi-Media Modeling Conference, Washington, D.C, USA: IEEE, 2006, 201-208.
- 8 Guo YB, Du YY, Xi JQ. A CP-net model and operation properties for Web service composition. Chinese Journal of Computers, 2006,29(7):1067-1075.
- 9 Qian ZZ, Lu SL, Xie L. Automatic composition of Petri net based Web service. Chinese Journal of Computers, 2006, 29(7):1057-1066.
- 10 Hamadi R, Benatallan B. A Petri net-based model for Web service composition. Proc. of the 14th Australasian Database Conference, Adelaide. Darlinghurst: Australian Computer Society, 2003,191-200.

(下转第 63 页)

操作. 双目定位系统是机器人的智能化的一个基本构件. 优化深度信息, 能够解决机器人定位不准确的问题, 对于机器人的准确抓取定位有很大帮助, 显著的提高机器人抓取目标物体的成功率. 尤其在复杂的多目标场景下, 机器人仍然能够较好的完成目标物体抓取操作, 具有较强的现实意义.

从第 4 部分的实验中, 可以看出, 通过优化深度信息后, 管家机器人(SIAT)^[11]抓取目标物体成功率提高了一倍之多. 所以利用这种方法, 使得机器人智能抓取目标物体具有比较高的精确度. 但是文中还没有很好的解决物体的上下高度与左右宽度问题. 如果要用到更为复杂的环境中, 可以认为机器人学习还是会准确的多. 但是训练的时间会长很多, 如何权衡利弊, 这也是我后期继续要研究的方向.

参考文献

- 1 Wen G, Xilin C. Computer Vision: Algorithm and System Principle. Beijing:Tsinghua University Press, 1999.
 - 2 Li H, Chen YL, Chang T, Wu X, Ou Y, Xu Y. Binocular vision Positioning for Robot Grasping. to appear in Proc. IEEE International Conference on Robotics and Biomimetics 2011.
 - 3 Forsyth D, Ponce J. Computer Vision: A Modern Approach. Prentice Hall,2002.
 - 4 MÜhlmann K, Maier D, Hesser J, Männer R. Calculating dense disparity maps from color stereo images. an efficient implementation, international Journal of Computer Vision, 2002,47(1):79–88.
 - 5 Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations. Proc. IEEE International Conference on Computer Vision, 1999, 666–679.
 - 6 Bouguet JY. Perona P, Camera calibration from points and lines in dual-space geometry. Proc. European Conference on Computer Vision, 1998. 2–6.
 - 7 Geiger A, Roser M, et al. Efficient Large-Scale Stereo Matching. Computer Vision--Asian Conference on Computer Vision, 2010,25–38.
 - 8 Wikipedia. OpenCV, <http://opencv.willowgarage.com/wiki/>.
 - 9 Tao TF, Koo JC. A Fast Block Matching Algorithm for Stereo Correspondence. Proc. International Conference on Computational Intelligence and Security. 2008. 38–44.
 - 10 Rosin PL. Measuring corner properties. Computer Vision and Image Understanding, 1999,73(2):291–307.
 - 11 Han L, Wu X, Liu G, Chen C, Ou Y, Xu Y. An efficient and low-cost robot grasping system household environment. Proc. World Congress on Intelligent Control and Automation (WCICA). 2011, 593–598.
 - 12 Gary B, Adrian K. Learning OpenCV. O'Reilly Mdia. Inc, 2008.
 - 13 Chuan L, Chen Y, et al. Occluded Object Grasping Based on Robot Stereo Vision, unpublished, WCICA,2011.
-
- (上接第 112 页)
- 11 Rao J, Su X. A survey of automated web service composition methods. Lecture Notes in Computer Science, 2005,3387: 43–54.
 - 12 Narayanan S, McIlraith A. Simulation, verification and automated composition of web service. Proc. of the Eleventh International World Wide Web Conference. Honolulu, USA,2002,77–88.
 - 13 Bucchiarone A, Gnesi S. A survey on services composition languages and models Proc. of International Workshop on Web Services Modeling and Testing, Palermo. Berlin: Springer-Verlag Press, 2006. 51– 63.
 - 14 吴哲辉. Petri 网导论.北京:机械工业出版社,2006.1–312.